2020 美团技术年货

CODE A BETTER LIFE

【顶会论文精选】

美团 美团

目录

论文

ISIA Food-500: A Dataset for Large-Scale Food Recognition via Stacked Global-Local Attention Network

Weiqing Min[1,2], Linhu Liu[1,2], Zhiling Wang[1,2], Zhengdong Luo[1,2], Xiaoming Wei[3], Xiaolin Wei[3], Shuqiang Jiang[1,2]

[1] Key Lab of Intelligent Information Processing, Institute of Computing Technology, CAS, Beijing, 100190, China
[2] University of Chinese Academy of Sciences, Beijing, 100049, China
[3] Meituan-Dianping Group

{minweiqing,sqjiang,luozhengdong}@ict.ac.cn;{linhu.liu,zhiling.wang}@vipl.ict.ac.cn;{weixiaoming,weixiaolin02}@meituan.com

ABSTRACT

Food recognition has received more and more attention in the multimedia community for its various real-world applications, such as diet management and self-service restaurants. A large-scale ontology of food images is urgently needed for developing advanced large-scale food recognition algorithms, as well as for providing the benchmark dataset for such algorithms. To encourage further progress in food recognition, we introduce the dataset ISIA Food-500 with 500 categories from the list in the Wikipedia and 399,726 images, a more comprehensive food dataset that surpasses existing popular benchmark datasets by category coverage and data volume. Furthermore, we propose a stacked global-local attention network, which consists of two sub-networks for food recognition. One sub-network first utilizes hybrid spatial-channel attention to extract more discriminative features, and then aggregates these multi-scale discriminative features from multiple layers into global-level representation (e.g., texture and shape information about food). The other one generates attentional regions (e.g., ingredient relevant regions) from different regions via cascaded spatial transformers, and further aggregates these multi-scale regional features from different layers into local-level representation. These two types of features are finally fused as comprehensive representation for food recognition. Extensive experiments on ISIA Food-500 and other two popular benchmark datasets demonstrate the effectiveness of our proposed method, and thus can be considered as one strong baseline. The dataset, code and models can be found at http://123.57.42.89/FoodComputing-Dataset/ISIA-Food500.html.

CCS CONCEPTS

• **Computing methodologies** → **Image representations**; **Object recognition**.

KEYWORDS

Food Recognition, Food Datasets, Benchmark, Deep Learning

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3414031

ACM Reference Format:
Weiqing Min[1,2], Linhu Liu[1,2], Zhiling Wang[1,2], Zhengdong Luo[1,2], Xiaoming Wei[3], Xiaolin Wei[3], Shuqiang Jiang[1,2]. 2020. ISIA Food-500: A Dataset for Large-Scale Food Recognition via Stacked Global-Local Attention Network. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3414031

1 INTRODUCTION

Food computing [38] is emerging as a new field to ameliorate the issues from many food-relevant fields, such as nutrition, agriculture and medicine. As one significant task in food computing, food recognition has received more attention in multimedia and beyond [15, 25, 36, 41] for its various applications, such as visual food diary [36], health-aware recommendation [42] and self-service restaurants [2].

Despite its great potential applications, recognizing food from images is still a challenging task, and the challenge derives from three-fold:

- **There is a lack of large-scale food dataset for food recognition.** Existing works mainly focus on utilizing smaller datasets for food recognition, such as ETH Food-101 [6] and Vireo Food-172 [7]. For example, Bossard *et al.* [6] released one food dataset ETH Food-101 from western cuisines with 101 food categories and 101,000 images. Chen *et al.* [7] introduced the Vireo Food-172 dataset from 172 Chinese food categories. These data-sets is lack of diversity and coverage in food categories and do not include a wide range of food images. Therefore, they are probably not sufficient to construct more complicated deep learning models for food recognition.
- **There are larger intra-class variations in the global appearance, shape and other configurations for food images.** As shown in Fig. 1, there are different shapes for the butter pecan and different textures appear in the mie goreng dish. Although numerous methods have been developed for addressing the problem of food recognition, most of these methods mainly focus on extracting features with certain type or some types while ignoring other aspects. For example, works on [4] mainly extracted color features while Niki *et al.* [32] designed a network to capture certain vertical structure for food recognition.
- **There are subtle discriminative details from food images, which are harder to capture in many cases.** Food recognition belongs to fine-grained recognition. Therefore, discriminative details are too subtle to be well-represented by existing

CNNs in many cases. As shown in Fig. 1, global features are not discriminative enough to distinguish between corn stew and leek soup. Although local regional features are probably more useful, we should carefully design one network to capture and represent such subtle difference. In order to improve the recognition performance, additional context information, such as location and ingredients [4, 41, 51, 59] is utilized. However, when these information is unavailable, these methods probably do not work.

Figure 1: Some samples from ISIA Food-500

In this work, we address data limitations by introducing a new large-scale dataset ISIA Food-500 with 399,726 images and 500 categories. In contrast with existing popular benchmark datasets, it is a more comprehensive food dataset with larger category coverage, larger data volume and higher diversity. To solve another two challenges, we propose a Stacked Global-Local Attention Network (SGLANet) to jointly learn complementary global and local visual features for food recognition. This is achieved by two sub-networks, namely Global Feature Learning Subnetwork (GloFLS) and Local-Feature Learning Subnetwork (LocFLS). GloFLS first utilizes hybrid spatial-channel attention to obtain more discriminative features for each layer, and then aggregates these features from different layers with both coarse and fine-grained levels, such as shape and texture cues about food into global-level features. LocFLS adopts cascaded Spatial Transformers (STs) to localize different attentional regions (e.g., ingredient-relevant regions), and aggregates fused regional features from different layers into local-level representation. In addition, SGLANet is trained with different types of losses in an end-to-end fashion to maximize their complementary effect in terms of discriminative power.

The contributions of our paper can be summarized as follows:

- We introduce a new large-scale and highly diverse food image dataset with 500 categories and about 400,000 images, which will be made publicly available to further the development of scalable food recognition.
- We propose a stacked global-local attention network architecture to jointly learn food-oriented global and local features via combining hybrid spatial-channel attention and multi-scale strategy for food recognition.
- We conduct extensive evaluation on our proposed dataset and other two popular food benchmark datasets to verify the effectiveness of our approach. As one strong baseline, code and models will also be released upon publication to support future research.

Table 1: Summary of available datasets for food recognition.

Dataset	#Images	#Categories	#Coverage
PFID [9]	4,545	101	Japanese
UEC Food100 [34]	14,361	100	Japanese
UEC Food256 [27]	25,088	256	Japanese
ETHZ Food-101 [6]	101,000	101	Western
UPMC Food-101 [48]	90,840	101	Western
UNIMIB2015 [12]	2,000	15	Misc.
UNIMIB2016 [13]	1,027	73	Misc.
ChineseFoodNet [10]	192,000	208	Chinese
Vireo Food-172 [7]	110,241	172	Chinese
KenyanFood13 [23]	8,174	13	Kenyan
Sushi-50 [44]	3,963	50	Japanese
FoodX-251 [26]	158,846	251	Misc.
ISIA Food-200 [41]	197,323	200	Misc.
ISIA Food-500	**399,726**	**500**	**Misc.**

2 RELATED WORK

Food-centric datasets More and more food datasets have been developed [6, 7, 26, 27, 34, 41] in recent years. Table 1 summarizes statistics of publicly available datasets for food recognition. The first benchmark is the PFID dataset [9] with only 4,545 images from 101 fast food categories. ETHZ Food-101 dataset [6] and VIREO Food-172 dataset [7] consist of more food images. However, these datasets failed in term of more comprehensive coverage of food categories, like object-centric ImageNet [14] and place-centric Places [58]. We hence introduce a new large scale food dataset ISIA Food-500 with 399,726 images and 500 food categories, and it aims at advancing multimedia food recognition and promoting the development of food-oriented multimedia intelligence.

There are some recipe-relevant multimodal datasets, such as Yummly28K [39], Yummly66K [37] and Recipe1M [45]. Recipe1M is the most known dataset, which contains about 1 million structured cooking recipes and their images for cross-modal retrieval. In contrast, the goal of our proposed ISIA Food-500 is for advancing multimedia food recognition.

Food Recognition Recently, Min *et al.* [38] gave a survey on food computing including food recognition. In the earlier years, various hand-crafted features are utilized for recognition [6, 53]. For example, Lukas *et al.* [6] utilized random forests to mine discriminative image patches as visual representation. Recent advances in deep learning have gained significant attention due to its impressive performance. As a result, existing methods resort to deep learning for food recognition [18, 25, 32]. There are also literatures, which utilize additional context information, such as ingredients and location [7, 41, 59] to improve the recognition performance. For example, Zhou *et al.* [59] exploited rich relationships among

ingredients and restaurant information through the bi-partite graph for food recognition. Different from these works, our work does not introduce additional context information, and design a two-branch network to jointly learn food-oriented global features(e.g., texture and shape) and local features (e.g.,ingredient-relevant regional features) to enable comprehensive and discriminative feature representation for food recognition.

In addition, our work is also very relevant to fine-grained image recognition [49], which aims to classify subordinate categories. Food recognition belongs to fine-grained image recognition. However, compared with other types of fine-grained objects, we should take characteristics of food images into consideration, and design the targeted network for food recognition.

3 ISIA FOOD-500

3.1 Dataset Construction

In order to obtain one high-quality food dataset with broad coverage, high diversity and density of samples, we build ISIA Food-500 from the following four steps:

(1) Constructing the Food Category List. In order to guarantee high-coverage of the categorical space, we resort to Wikipedia to construct the food concept system. Particularly, we built the food list according to "Lists of foods by ingredient" from Wikipedia[1]. The Deep-First-Search algorithm is used to traverse links of the website to find food categories more completely. After that, we obtained the original food list with 4,943 types. We then removed redundant food types and conducted the combination for synonyms. Finally, we obtained 3,309 food categories.

(2) Collecting Food Images. Using a query term from the constructed food category list, we crawled candidate images from various search engines (i.e., Google, Bing and Baidu) for broader coverage and higher diversity of food images compared with other datasets from only one data source. In order to ensure that crawled images are less noisy, we expanded search terms by adding keywords, such as "food" and "dish". In this case, images for each term are retrieved and these images are then combined from different search engines. Because some images crawled from different search engines are repeated, we conducted hash based duplication detection to remove repeated ones.

(3) Cleaning and Pre-processing Food Images. Images are cleaned up through both automatic and manual processing. For automatic data cleaning, we removed candidate images with incomplete RGB channels, and the length or width of an image less than 100 pixels. We next trained a food/non-food binary classifier to further remove non-food images. Particularly, we combined images from the training set of both ETHZ Food-101 (western dishes) and VireoFood-172 (eastern dishes) as positive samples of the training set. We then randomly selected about 400,000 non-food images from both ImageNet and Places365 as negative samples of the training set. All the test samples of both ETHZ Food-101 and VireoFood-172 and the other 100,000 non-food images randomly selected from both ImageNet and Places365 constitute the test set. We trained a deep network (VGG-16 in our work) on the constructed training set and the classification accuracy of the network achieved 99.48% on the test set. The trained model is then used to filter out non-food images from downloaded images. After automatic cleaning, we then conduct manual verification by crowd-sourcing the task to 20 Lab members.

(4) Scaling Up the Dataset. After image collection and annotation, there are still many food categories with few images. To further increase the number of the candidate dataset, we translated the name of these food categories into different languages, such as Chinese and French, and then crawled images from three search engines. We also crawled more food images from other recipe/food shared websites, such as Allrecipes.com and foodgawker.com. We finally selected 500 categories with more than 500 images per category as our resulting dataset.

3.2 Dataset Statistics and Characteristics

ISIA Food-500 consists of 399,726 images with 500 categories. The average number of images per category is about 800. Fig. 2 shows sorted distribution of the number of images from sampled classes while Fig. 3 shows some samples. Note that we represented the food category with more than two words by concatenating them using '-'. ISIA Food-500 is a more comprehensive food dataset that surpasses existing popular benchmark datasets, such as ETH Food-101 and Vireo Food-172 from the following three aspects: **(1) Larger data volume**. It has 399,726 images from 500 food categories, which has created a new milestone for the task of complex food recognition. **(2) Larger category coverage**. It consists of 500 categories, which is about $3 \sim 5$ times that of existing datasets, such as Food-101 and Vireo Food-172. **(3) Higher diversity**. Food categories from this dataset covers various countries and regions including both eastern and western cuisines. Fig. 4 provided the comparisons of distributions of food categories on food types, such as ETH Food-101 (western food), Vireo Food-172 (eastern food) and ISIA Food-200 (Misc. food). According to the GSFA standard[2], the food from our dataset and existing typical ones mainly belongs to the following 11 categories: Meat, Cereals, Vegetables, Fish, Fruits, Dairy, Bakery, Fats, Confectionary, Beverages and Eggs. We can see that for most of food types, the number of food categories from ISIA Food-500 is larger than these existing datasets. Furthermore, some food types are covered in ISIA Food-500, but missing in other ones, such as Dairy and Beverages.

4 FRAMEWORK

Fig. 5 illustrates the proposed Stacked Global-Local Attention Network (SGLANet), which can jointly learn complementary global and local features for food recognition. SGLANet mainly consists of two components, namely **Glo**bal **F**eature **L**earning **S**ub-network (GloFLS) and **Loc**al-**F**eature **L**earning **S**ub-network (LocFLS). GloFLS first adopts hybrid Spatial-Channel Attention (SCA) to obtain more discriminative features from each layer of the network, and then aggregates a set of features from these layers to capture different types of global level features, such as shape and texture cues about food. LocFLS adopts cascaded STs to localize different local regions for each layer, and then aggregates fused features with different regions from different layers into final local feature representation. Finally, SGLANet fuses both global and local features

[1] https://en.wikipedia.org/wiki/Category:Lists_of_foods

[2] http://www.fao.org/gsfaonline/index.html?lang=en

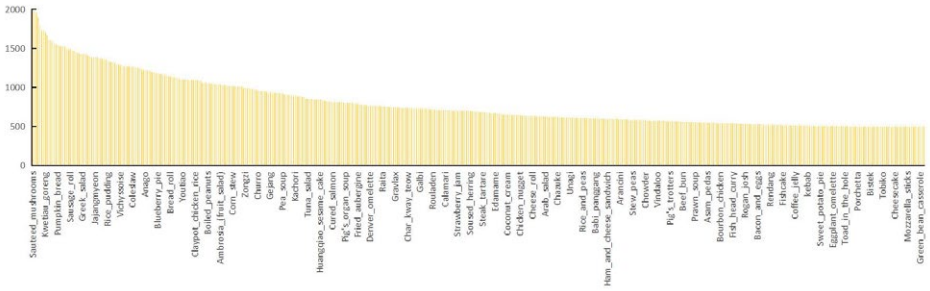


Figure 2: Sorted distribution of the number of images from sampled classes in the ISIA Food-500.

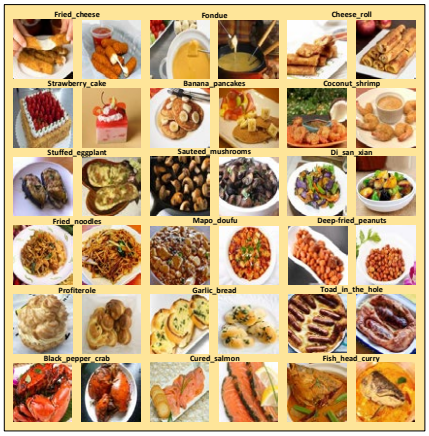


Figure 3: Image samples from the ISIA Food-500 dataset

for food recognition. In addition, SGLANet is trained with different types of losses, including global loss, local loss and joint loss in an end-to-end fashion to maximize their complementary benefit in terms of the discriminative power.

4.1 GloFLS

Given the whole input image, GloFLS first learns more discriminative features via hybrid Spatial-Channel Attention (SCA) for each layer, and then aggregates these discriminative features from different layers into global level representations via multi-layer feature fusing. Considering features extracted from different layers contain low-level, mid and high ones, GloFLS can capture various types of global level features, such as shape, texture and edge cues about food.

Spatial-Channel Attention (SCA) The combination of both spatial and channel attention can capture discriminative features comprehensively from different dimensions, and thus have been successfully applied in many tasks, such as image captioning [8] and person ReID [29]. Different from these works, we apply SCA to the food recognition task by capturing food-oriented discriminative features.

The input to a SCA module is a 3-D tensor $\mathbf{X}^l \in \mathrm{R}^{h \times w \times c}$ with width w, height h, channels c and the layer of GloFLS l, respectively. The output of this module is a saliency weight map $\mathbf{A}^l \in \mathrm{R}^{h \times w \times c}$ of the same size as $\mathbf{X}$. We calculate $\mathbf{A}^l \in \mathrm{R}$ for SCA learning [29]:

$$\mathbf{A}^l = \mathbf{S}^l \times \mathbf{C}^1 \tag{1}$$

where $\mathbf{S}^l \in \mathrm{R}^{h \times w \times 1}$ and $\mathbf{C}^l \in \mathrm{R}^{1 \times 1 \times c}$ mean spatial and channel attention maps, respectively.

The Global Averaging Pooling (GAP) is used to calculate the spatial attention as follows:

$$\mathbf{S}^l = \frac{1}{c} \sum_{i=1}^{c} \mathbf{X}^l_{1:h,1:w:i} \tag{2}$$

The channel attention from the squeeze-and-excitation block [19] is computed as follows:

$$\begin{aligned} \mathbf{C}^l_1 &= \frac{1}{h \times w} \sum_{i=1}^{h} \sum_{j=1}^{w} \mathbf{X}^l_{i,j,1:c} \\ \mathbf{C}^l &= ReLU(\mathbf{M}^{ca}_2 \times \mathrm{Relu}(\mathbf{M}^{ca}_1 \mathbf{C}^l_1)) \end{aligned} \tag{3}$$

where $\mathbf{M}^{ca}_1 \in \mathbf{R}^{\frac{c}{r} \times c}$ and $\mathbf{M}^{ca}_2 \in \mathbf{R}^{c \times \frac{c}{r}}$ represent the parameter matrix of 2 conv layers respectively, and r denotes the bottleneck reduction rate.

Multi-Layer Feature Fusing By extracting attentional features from multiple layers, we can obtain low, mid and high-level features, which include various types of global features, such as texture, shape and edge information [54]. Such global features are important cues for food recognition. Therefore, we aggregate discriminative attentional features from different layers into global level feature representation for food recognition via a concatenation layer and a fully connected layer.

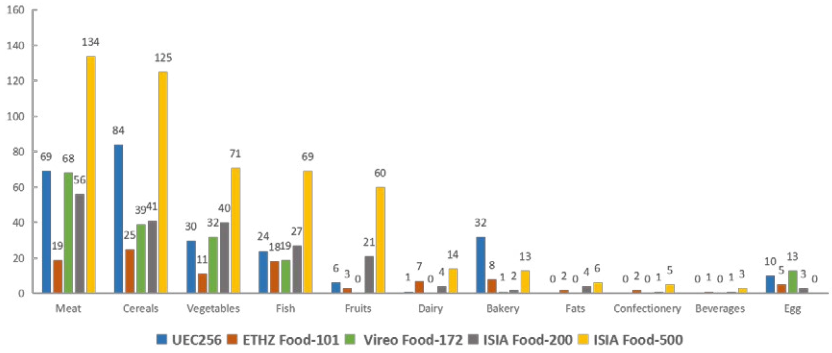


Figure 4: Comparison on distributions of categories on ISIA Food-500 and other existing typical ones.

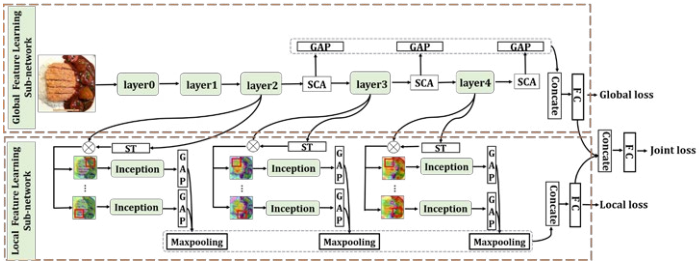


Figure 5: The proposed framework. GAP: Global Average Pooling layer. SCA: Spatial-Channel Attention. ST: Spatial Transformer. FC: Full-Connected layer.

4.2 LocFLS

LocFLS localizes discriminative regions with different positions and scales to capture local features. It uses stacked STs [22] to localize regions from different layers. For each layer, one inception block is introduced to extract regional features, and followed by a global average pooling layer and a max-pooling layer for fusing these regional features. The features from each layer are fused to final local features via a concatenation layer and a fully connected layer.

Spatial Transformer (ST) For each layer, we adopt ST to locate latent T regions, and model this regional attention by a transformation matrix as:

$$\mathbf{A}^l = \begin{bmatrix} s_h & 0 & t_x \\ 0 & s_w & t_y \end{bmatrix} \tag{4}$$

which allows for image cropping, translation, and isotropic scaling operations by varying two scale factors (s_h , s_w) and 2-D spatial position (t_x, t_y).

4.3 Learning with Multiple Losses

SGLANet is jointly optimized by three types of losses, i.e., joint loss L_{Joi}, global loss L_{Glo}, and local loss L_{Loc} respectively, leading to the final loss function:

$$L = L_{Joi} + \gamma_1 L_{Glo} + \gamma_2 L_{Loc} \tag{5}$$

where γ_1 and γ_2 are balance parameters, and the cross-entropy classification loss function is used for all three types of losses.

Such learning with different types of losses can maximize their complementary benefit in terms of the discriminative power.

5 EXPERIMENT

5.1 Experimental Setup

Our model is implemented on the Pytorch platform. The images are resized to 224×224. The models are optimized using stochastic gradient descent with a batch size of 80 and momentum of 0.9. The learning rate is set to 10^{-2} initially and divided by 10 after 30 epochs. For GloFLS, we selected SENet [19] as the backbone, and

Table 2: Evaluating individual modules in GloFLS on ISIA Food-500 (%).

Method	Top-1 acc.	Top-5 acc.
SENet-154	63.83	88.61
SENet-154+SCA	64.42	89.05
SENet-154+Multi-scale	64.60	89.08
GloFLS	**64.63**	**89.14**

Table 3: Ablation experiments on ISIA Food-500 with global & local-level features (%).

Method	Top-1 acc.	Top-5 acc.
GloFLS	64.63	**89.14**
LocFLS	64.10	88.86
SGLANet	**64.74**	89.12

the bottleneck reduction rate $r = 16$. For LocFLS, we selected simple Inception-B unit as basic building block. For each layer, $T = 4$ and the scale of ST is fixed as $s_h = s_w = 0.5$. We set $\gamma_1 = \gamma_2 = 0.5$ in Eq. 5. Top-1 accuracy (Top-1 acc.) and Top-5 accuracy (Top-5 acc.) are used as evaluation metrics.

5.2 Experiment on ISIA Food-500

ISIA Food-500 is divided into 60%, 10% and 30% images for training, validation and testing, respectively. All the experiments adopt a single centered crop (1-crop) at test time in the defaulting setting.

Ablation Study We first evaluated the effect of each individual component in GloFLS in Table 2. It shows that: (1) Any of two components in isolation brings recognition performance gain; (2) The combination of SCA and Multi-scale gives further accuracy boost, which suggests the complementary effect. We then evaluated the effect of joint global and local feature learning by comparing their individual global and local features. Table 3 shows that a performance gain is obtained in Top-1 accuracy by joining two representations, which validates the complementary effect of jointly learning global and local features from GloFLS and LocFLS.

We finally evaluate the effect of different losses as shown in Table 4. The experimental results demonstrate that we obtain the best recognition performance when different losses are utilized. The reason is that different loss functions can regulate the deep network from different aspects and work together to improve the recognition performance. Another observation is that the performance with one additional loss does not improve the performance compared with the baseline without both global and local losses. The probable reason is that the performance improvement needs joint work from two losses.

Comparisons with State-of-the-Art We evaluated SGLANet against different baseline methods on Table 5. These baselines include not only various typical deep networks, such as VGG16 and SENet, but also some recently proposed fine-grained methods, such as NTS-NET [55] and WS-DAN [20]. Note that for these fine-grained methods, we followed the same setting in their mentioned papers. We observed that the performance superiority of SGLANet over all the state-of-the-arts in both Top-1 accuracy and Top-5 accuracy. Compared with best baseline SENet-154, there is the performance improvement of about 0.9 percent in Top-1 accuracy for the test set. These results validate the advantage of joint global and local feature learning of SGLANet.

Table 4: Evaluating individual losses on ISIA Food-500 (%).

Method	Top-1 acc.	Top-5 acc.
$\lambda_1 = \lambda_2 = 0$	64.16	88.94
$\lambda_1 = 0, \lambda_2 = 0.5$	63.95	88.57
$\lambda_1 = 0.5, \lambda_2 = 0$	64.02	88.59
$\lambda_1 = 0.5, \lambda_2 = 0.5$	**64.74**	**89.12**

Table 5: Performance comparison on ISIA Food-500 (%).

Method	Top-1 acc.	Top-5 acc.
VGG-16 [47]	55.22	82.77
GoogLeNet [36]	56.03	83.42
ResNet-152 [17]	57.03	83.80
WRN-50 [46]	60.08	85.98
DenseNet-161 [21]	60.05	86.09
NAS-NET [60]	60.66	86.38
SE-ResNeXt101_32x4d [19]	61.95	87.54
NTS-NET [55]	63.66	88.48
WS-DAN [20]	60.67	86.48
DCL [11]	64.10	88.77
SENet-154 [19]	63.83	88.61
SGLANet	**64.74**	**89.12**

Visualization of GloFLS and LocFLS We visualized both SCA from GloFLS and STs from LocFLS at three different layers of SGLANet. Fig. 6 shows: (1) in GloFLS, SCA captures different global level features at different layers, such as shape information for Boiled_beef and texture information from Pumpkin_bread. Meanwhile, with increased depth of SGLANet, SCA captures more focused and discriminative features (2) in LocFLS, STs capture different local regions with less background at different layers from LocFLS, such as Crudites. This again verified complementary effect of joint global and local feature learning.

Qualitative Analysis We selected 20 classes in the test phase to further evaluate our method. Particularly, we listed the Top-1 accuracy of both 10 best and 10 worst performing classes in Fig. 7. We can observe that some categories can be easily recognized, such as Chakli and Edamame, and their Top-1 accuracy is above 97%. However, there are some categories, which are very hard to recognize, such as Curry_rice and kebab, and their Top-1 accuracy is below 10%. We further demonstrate some challenging recognized examples from the 10 worst performing classes, and Fig. 8 shows that too small inter-class variations is the main reason for bad performance. We have shown that existing methods are far from tackling large-scale recognition task with high accuracy like ImageNet, pointing to exciting future directions.

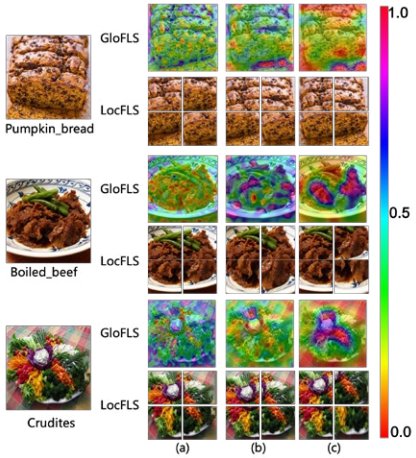
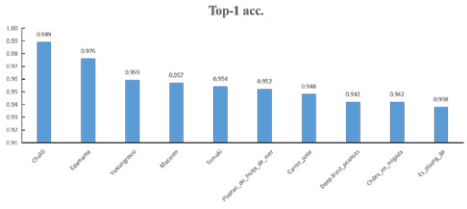


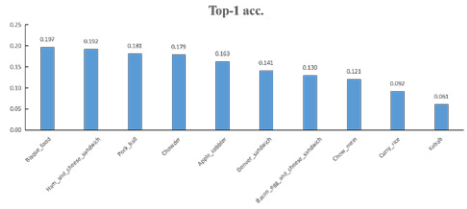
Figure 6: Visualization of SCA in GloFLS and STs in LocFLS from (a) The 2^{th} layer,(b) The 3^{th} layer and (c) The 4^{th} layer.

Table 6: Performance comparison on ETHZ Food-101 (%).

Method	Setting	Top-1 acc.	Top-5 acc.
AlexNet-CNN [6]	1-crop	56.40	-
SELC [33]	1-crop	55.89	-
ResNet-152+SVM-RBF [35]	1-crop	64.98	-
DCNN-FOOD [52]	1-crop	70.41	-
LMBM [50]	1-crop	72.11	-
Ensemble Net [43]	1-crop	72.12	91.61
GoogLeNet [3]	1-crop	78.11	-
DeepFOOD [30]	1-crop	77.40	93.70
ILSVRC [5]	1-crop	79.20	94.11
WARN [31]	1-crop	85.50	-
CNNs Fusion(I_2) [1]	1-crop	86.71	-
Inception V3 [16]	1-crop	88.28	96.88
SENet-154 [19]	1-crop	88.62	97.57
WRN [32]	10-crop	88.72	97.92
SOTA[28]	1-crop	90.00	-
DLA[57]	1-crop	90.00	-
WISeR [32]	10-crop	90.27	**98.71**
IG-CMAN [41]	1-crop	90.37	98.42
PAR-Net [44]	1-crop	89.30	-
PAR-Net [44]	10-crop	90.40	-
Inception-Resnet-v2 SE [56]	1-crop	90.40	-
MSMVFA [24]	1-crop	90.59	98.25
SGLANet	1-crop	89.69	98.01
SGLANet	10-crop	90.33	98.20
SGLANet(Pretrained)	1-crop	90.47	98.21
SGLANet(Pretrained)	10-crop	**90.92**	98.24



(a) The 10 best performing classes



(b) The 10 worst performing classes

Figure 7: Selected categories from (a)The 10 best and (b)The 10 worst performing classes.

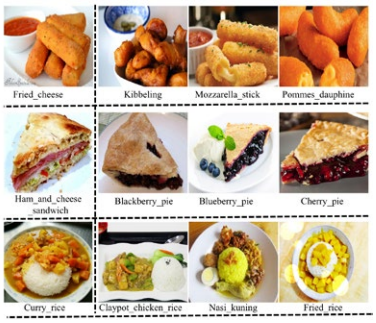


Figure 8: Some confused classes, where the first column denotes some classes from the 10 worst performing classes and for each class, 3 more confused classes are listed for each row.

5.3 Experiment on Other Benchmarks

We further conduct extensive evaluation on other two popular food benchmark datasets to verify the effectiveness of our approach, and also assessed the generalizability of our model learned using ISIA Food-500 to the two datasets. Considering some evaluations from

Table 7: Performance comparison on Vireo Food-172 (%).

Method	Setting	Top-1 acc.	Top-5 acc.
AlexNet	1-crop	64.91	85.32
VGG-16 [47]	1-crop	80.41	94.59
DenseNet-161 [21]	1-crop	86.93	97.17
MTDCNN(VGG-16) [7]	1-crop	82.06	95.88
MTDCNN(DenseNet-16) [7]	1-crop	87.21	97.29
SENet-154 [19]	1-crop	88.71	97.74
PAR-Net [44]	1-crop	89.60	-
PAR-Net [44]	10-crop	90.20	-
IG-CMAN [41]	1-crop	90.63	**98.40**
MSMVFA [24]	1-crop	90.61	98.31
SGLANet	1-crop	89.88	97.83
SGLANet	10-crop	90.30	98.03
SGLANet(Pretrained)	1-crop	90.78	98.16
SGLANet(Pretrained)	10-crop	**90.98**	98.35

existing works are conducted in the setting of 10-crop test, we show the experimental results of our method in the setting of both 1-crop and 10-crop at test time.

Experiments on ETHZ Food-101 ETHZ Food-101 contains 101,000 images from 101 food categories. There are 1,000 images including 750 training images and 250 test images for each category [6]. We evaluated SGLANet against existing methods on Food-101. Table 6 shows that our method exceeds many baseline methods except some ones, such as MSMVFA [24], IG-CMAN [41] and Inception-Resnet-v2 SE [56] under the 1-crop test setting. The reason is that MSMVFA and IG-CMAN require multiple stages training for feature extraction and introduced additional ingredient information as the supervision. Inception-Resnet-v2 SE used additional data and adopted transfer learning method. When we used the pretrained model on ISIA Food-500, namely SGLANet(Pretrained), there is the performance improvement of about 0.8 percent and 0.6 percent in Top-1 accuracy on 1-crop and 10-crop test respectively. These results also verify the generalization of models learned using ISIA Food-500.

Experiments on Vireo Food-172 Vireo Food-172 consists of 110,241 food images from 172 categories. In each food category, 60%, 10%, 30% of images are randomly selected for training, validation and testing, respectively [7]. Table 7 shows experimental results on Vireo Food-172. We can see that the performance from SGLANet is better than many baselines, except that few ones, such as IG-CMAN. This is because that these methods, such as IG-CMAN introduced additional ingredient information for food recognition. In addition, these methods generally need multi-stage feature learning. When we fine-tuned SGLANet pre-trained on ISIA Food-500, there is the performance improvement of about 0.9 percent and 0.7 percent in Top-1 accuracy on 1-crop and 10-crop test respectively, and achieved the best performance under the 1-crop setting. These results again demonstrate the generalization of our model learned using ISIA Food-500.

6 CONCLUSIONS

In this paper, we present a new large-scale dataset ISIA Food-500 with larger data volume, larger category coverage, and higher diversity compared with existing typical datasets. We then propose a stacked global-local attention network to jointly exploit complementary global and local features via the designed two subnetworks for food recognition. Extensive evaluation on ISIA Food-500 and another two benchmark datasets have verified its effectiveness, and thus can be considered as one strong baseline.

Future work includes: (1) We are expanding ISIA Food-500 dataset, and aim to complete the construction of about 1.5 million food images spread over about 2,000 food categories. We expect it will serve as a new challenge to train high-capacity models for large-scale food recognition in the multimedia community. (2) We plan to collect rich attribute information, e.g., ingredients, cooking instructions and flavor information [40] to support multimodal food recognition.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under Grant 61972378, 61532018, U1936203, U19B2040. This research was also supported by Meituan-Dianping Group.

REFERENCES

[1] Eduardo Aguilar, Marc Bolaños, and Petia Radeva. 2017. Food recognition using fusion of classifiers based on CNNs. In *International Conference on Image Analysis and Processing*. 213–224.
[2] Eduardo Aguilar, Beatriz Remeseiro, Marc Bolaños, and Petia Radeva. 2018. Grab, Pay and Eat:Semantic Food Detection for Smart Restaurants. In *IEEE Transactions on Multimedia*, Vol. 20. 3266–3275.
[3] Shuang Ao and Charles X. Ling. 2015. Adapting new categories for food recognition with deep representation. In *IEEE International Conference on Data Mining Workshop*. 1196–1203.
[4] Vinay Bettadapura, Edison Thomaz, Aman Parnami, Gregory D Abowd, and Irfan Essa. 2015. Leveraging context to support automated food recognition in restaurants. In *IEEE Winter Conference on Applications of Computer Vision*. 580–587.
[5] Marc Bolanos and Petia Radeva. 2017. Simultaneous food localization and recognition. In *International Conference on Pattern Recognition*. 3140–3145.
[6] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101–mining discriminative components with random forests. In *European Conference on Computer Vision*. 446–461.
[7] Jingjing Chen and Chong-Wah Ngo. 2016. Deep-based ingredient recognition for cooking recipe retrieval. In *Proceedings of the ACM on Multimedia Conference*. 32–41.
[8] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T. Chua. 2017. SCA-CNN: Spatial and Channel-Wise Attention in Convolutional Networks for Image Captioning. In *IEEE Conference on Computer Vision and Pattern Recognition*. 6298–6306.
[9] Mei Chen, Kapil Dhingra, Wen Wu, Lei Yang, Rahul Sukthankar, and Jie Yang. 2009. PFID: Pittsburgh fast-food image dataset. In *IEEE International Conference on Image Processing*. 289–292.
[10] Xin Chen, Hua Zhou, and Liang Diao. 2017. ChineseFoodNet: A large-scale image dataset for Chinese food recognition. In *CoRR*, Vol. abs/1705.02743.
[11] Yue Chen, Yalong Bai, Wei Zhang, and Tao Mei. 2019. Destruction and Construction Learning for Fine-Grained Image Recognition. *In: IEEE International Conference on Computer Vision and Pattern Recognition* (2019), 5157–5166.
[12] Gianluigi Ciocca, Paolo Napoletano, and Raimondo Schettini. 2015. Food Recognition and Leftover Estimation for Daily Diet Monitoring. In *International Conference on Image Analysis and Processing*. 334–341.
[13] Gianluigi Ciocca, Paolo Napoletano, and Raimondo Schettini. 2016. Food recognition: a new dataset, experiments, and results. In *IEEE Journal of Biomedical and Health Informatics*, Vol. 21. 588–598.
[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 2009. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*. 248–255.
[15] Lixi Deng, Jingjing Chen, Qianru Sun, Xiangnan He, Sheng Tang, Zhaoyan Ming, Yongdong Zhang, and Tat-Seng Chua. 2019. Mixed-dish Recognition with

Contextual Relation Networks. In *ACM Multimedia*. ACM, 112–120.
[16] Hamid Hassannejad, Guido Matrella, Paolo Ciampolini, Ilaria De Munari, Monica Mordonini, and Stefano Cagnoni. 2016. Food image recognition using very deep convolutional networks. In *International Workshop on Multimedia Assisted Dietary Management*. 41–49.
[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.
[18] S. Horiguchi, S. Amano, M. Ogawa, and K. Aizawa. 2018. Personalized Classifier for Food Image Recognition. *IEEE Transactions on Multimedia* 20, 10 (2018), 2836–2848.
[19] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-Excitation Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*. 7132–7141.
[20] Tao Hu and Honggang Qi. 2019. See Better Before Looking Closer: Weakly Supervised Data Augmentation Network for Fine-Grained Visual Classification. *CoRR* abs/1901.09891.
[21] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. 2017. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2261–2269.
[22] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. 2015. Spatial Transformer Networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems*. 2017–2025.
[23] Mona Jalal, Kaihong Wang, Sankara Jefferson, Yi Zheng, Elaine O. Nsoesie, and Margrit Betke. 2019. Scraping Social Media Photos Posted in Kenya and Elsewhere to Detect and Analyze Food Types. In *Proceedings of the 5th International Workshop on Multimedia Assisted Dietary Management*. 50–59.
[24] Shuqiang Jiang, Weiqing Min, Linhu Liu, and Zhengdong Luo. 2019. Multi-Scale Multi-View Deep Feature Aggregation for Food Recognition. *IEEE Transactions on Image Processing* 29, 1, 265–276.
[25] Hokuto Kagaya, Kiyoharu Aizawa, and Makoto Ogawa. 2014. Food detection and recognition using convolutional neural network. In *Proceedings of the ACM International Conference on Multimedia*. 1085–1088.
[26] Parneet Kaur, Karan Sikka, Weijun Wang, Serge J. Belongie, and Ajay Divakaran. 2019. FoodX-251: A Dataset for Fine-grained Food Classification. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*.
[27] Yoshiyuki Kawano and Keiji Yanai. 2014. Automatic expansion of a food image dataset leveraging existing categories with domain adaptation. In *European Conference on Computer Vision*. 3–17.
[28] Simon Kornblith, Jonathon Shlens, and Quoc Le. 2019. Do Better ImageNet Models Transfer Better?. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2661–2671.
[29] W. Li, X. Zhu, and S. Gong. 2018. Harmonious Attention Network for Person Re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2285–2294.
[30] Chang Liu, Yu Cao, Yan Luo, Guanling Chen, Vinod Vokkarane, and Yunsheng Ma. 2016. Deepfood: Deep learning-based food image recognition for computer-aided dietary assessment. In *International Conference on Smart Homes and Health Telematics*. 37–48.
[31] Pau RodrÍguez López, Diego Velazquez Dorta, Guillem Cucurull Preixens, Josep M. Gonfaus, and Jordi Gonzàlez Sabaté. 2020. Pay attention to the activations: a modular attention mechanism for fine-grained image recognition. In *IEEE Transactions on Multimedia*, Vol. 22. 502–514.
[32] Niki Martinel, Gian Luca Foresti, and Christian Micheloni. 2018. Wide-slice residual networks for food recognition. In *IEEE Winter Conference on Applications of Computer Vision*. 567–576.
[33] Niki Martinel, Claudio Piciarelli, and Christian Micheloni. 2016. A supervised extreme learning committee for food recognition. In *Computer Vision and Image Understanding*, Vol. 148. Elsevier, 67–86.
[34] Yuji Matsuda and Keiji Yanai. 2012. Multiple-food recognition considering co-occurrence employing manifold ranking. In *International Conference on Pattern Recognition*. 2017–2020.
[35] Patrick McAllister, Huiru Zheng, Raymond Bond, and Anne Moorhead. 2018. Combining deep residual neural network features with supervised machine learning algorithms to classify diverse food image datasets. In *Computers in Biology and Medicine*, Vol. 95. Elsevier, 217–233.
[36] Austin Meyers, Nick Johnston, Vivek Rathod, Anoop Korattikara, Alex Gorban, Nathan Silberman, Sergio Guadarrama, George Papandreou, Jonathan Huang, and Kevin P Murphy. 2015. Im2Calories: towards an automated mobile vision food diary. In *Proceedings of the IEEE International Conference on Computer Vision*. 1233–1241.
[37] Weiqing Min, Bing-Kun Bao, Shuhuan Mei, Yaohui Zhu, Yong Rui, and Shuqiang Jiang. 2018. You are what you eat: Exploring rich recipe information for cross-region food analysis. *IEEE Transactions on Multimedia* 20, 4 (2018), 950–964.
[38] Weiqing Min, Shuqiang Jiang, Linhu Liu, Yong Rui, and Ramesh Jain. 2019. A Survey on Food Computing. In *ACM Computing Surveys*, Vol. 52. 1–36.
[39] Weiqing Min, Shuqiang Jiang, Jitao Sang, Huayang Wang, Xinda Liu, and Luis Herranz. 2017. Being a Super Cook: Joint Food Attributes and Multi-Modal Content Modeling for Recipe Retrieval and Exploration. *IEEE Transactions on Multimedia* 19, 5 (2017), 1100–1113.
[40] Weiqing Min, Shuqiang Jiang, Shuhui Wang, Jitao Sang, and Shuhuan Mei. 2017. A Delicious Recipe Analysis Framework for Exploring Multi-Modal Recipes with Various Attributes. In *ACM Multimedia*. ACM, 402–410.
[41] Weiqing Min, Linhu Liu, Zhengdong Luo, and Shuqiang Jiang. 2019. Ingredient-Guided Cascaded Multi-Attention Network for Food Recognition. In *Proceedings of the ACM International Conference on Multimedia*. 1331–1339.
[42] Nitish Nag, Vaibhav Pandey, and Ramesh Jain. 2017. Health multimedia: Lifestyle recommendations based on diverse observations. In *Proceedings of the ACM on International Conference on Multimedia Retrieval*. 99–106.
[43] Paritosh Pandey, Akella Deepthi, Bappaditya Mandal, and N. B. Puhan. 2017. FoodNet: Recognizing foods using ensemble of deep networks. In *IEEE Signal Processing Letters*, Vol. 24. 1758–1762.
[44] Jianing Qiu, Frank P.-W. Lo, Yingnan Sun, Siyao Wang, and Benny Lo. 2019. Mining Discriminative Food Regions for Accurate Food Recognition. In *British Machine Vision Conference (Accepted)*.
[45] Amaia Salvador, Nicholas Hynes, Yusuf Aytar, Javier Marin, Ferda Ofli, Ingmar Weber, and Antonio Torralba. 2017. Learning cross-modal embeddings for cooking recipes and food images. In *IEEE Conference on Computer Vision and Pattern Recognition*. 3020–3028.
[46] Zagoruyko Sergey and Komodakis Nikos. 2016. Wide Residual Networks. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 87.1–87.12.
[47] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1–9.
[48] Xin Wang, Devinder Kumar, Nicolas Thome, Matthieu Cord, and Frederic Precioso. 2015. Recipe recognition with large multimodal food dataset. In *IEEE International Conference on Multimedia and Expo Workshops*. 1–6.
[49] Xiu-Shen Wei, Jianxin Wu, and Quan Cui. 2019. Deep Learning for Fine-Grained Image Analysis: A Survey. *CoRR* abs/1907.03069 (2019).
[50] Hui Wu, Michele Merler, Rosario Uceda-Sosa, and John R Smith. 2016. Learning to make better mistakes: Semantics-aware visual food recognition. In *ACM Multimedia Conference*. 172–176.
[51] Ruihan Xu, Luis Herranz, Shuqiang Jiang, Shuang Wang, Xinhang Song, and Ramesh Jain. 2015. Geolocalized modeling for dish recognition. In *IEEE Transactions on Multimedia*, Vol. 17. 1187–1199.
[52] Keiji Yanai and Yoshiyuki Kawano. 2015. Food image recognition using deep convolutional network with pre-training and fine-tuning. In *IEEE International Conference on Multimedia and Expo Workshops*. 1–6.
[53] Shulin Yang, Mei Chen, Dean Pomerleau, and Rahul Sukthankar. 2010. Food recognition using statistics of pairwise local features. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2249–2256.
[54] S. Yang and D. Ramanan. 2015. Multi-scale Recognition with DAG-CNNs. In *IEEE International Conference on Computer Vision*. 1215–1223.
[55] Ze Yang, Tiange Luo, Dong Wang, Zhiqiang Hu, Jun Gao, and Liwei Wang. 2018. Learning to Navigate for Fine-Grained Classification. In *European Conference on Computer Vision*. 438–454.
[56] Cui Yin, Song Yang, Sun Chen, Howard Andrew, and Belongie Serge. 2018. Large Scale Fine-Grained Categorization and Domain-Specific Transfer Learning. In *IEEE Conference on Computer Vision and Pattern Recognition*. 4109–4118.
[57] F. Yu, D. Wang, E. Shelhamer, and T. Darrell. 2018. Deep Layer Aggregation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2403–2412.
[58] Bolei Zhou, Àgata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2018. Places: A 10 Million Image Database for Scene Recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 40. 1452–1464.
[59] Feng Zhou and Yuanqing Lin. 2016. Fine-Grained Image Classification by Exploring Bipartite-Graph Labels. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1124–1133.
[60] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc Le. 2018. Learning Transferable Architectures for Scalable Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 8697–8710.

Query Twice: Dual Mixture Attention Meta Learning for Video Summarization

Junyan Wang[1], Yang Bai[2], Yang Long[3],
Bingzhang Hu[2], Zhenhua Chai[1], Yu Guan[2], Xiaolin Wei[1]
[1]Vision Intelligence Center, Meituan-Dianping Group, Beijing, China
[2]OpenLab, Newcastle University, Newcastle upon Tyne, UK
[3]Department of Computer Science, Durham University, Durham, UK
{wangjunyan04, chaizhenhua, weixiaolin02}@meituan.com,
{y.bai13, bingzhang.hu, yu.guan}@newcastle.ac.uk, yang.long@ieee.org

ABSTRACT

Video summarization aims to select representative frames to retain high-level information, which is usually solved by predicting the segment-wise importance score via a softmax function. However, softmax function suffers in retaining high-rank representations for complex visual or sequential information, which is known as the *Softmax Bottleneck* problem. In this paper, we propose a novel framework named Dual Mixture Attention (DMASum) model with Meta Learning for video summarization that tackles the softmax bottleneck problem, where the Mixture of Attention layer (MoA) effectively increases the model capacity by employing twice self-query attention that can capture the second-order changes in addition to the initial query-key attention, and a novel Single Frame Meta Learning rule is then introduced to achieve more generalization to small datasets with limited training sources. Furthermore, the DMASum significantly exploits both visual and sequential attention that connects local key-frame and global attention in an accumulative way. We adopt the new evaluation protocol on two public datasets, SumMe, and TVSum. Both qualitative and quantitative experiments manifest significant improvements over the state-of-the-art methods.

CCS CONCEPTS

• **Computing methodologies** → **Video summarization**.

KEYWORDS

video summarization, attention network, meta learning

ACM Reference Format:
Junyan Wang, Yang Bai, Yang Long, Bingzhang Hu, Zhenhua Chai, Yu Guan and Xiaolin Wei. 2020. Query Twice: Dual Mixture Attention Meta Learning for Video Summarization. In *Proceedings of the 28th ACM International Conference on Multimedia (MMâĂŹ20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3414064

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3414064

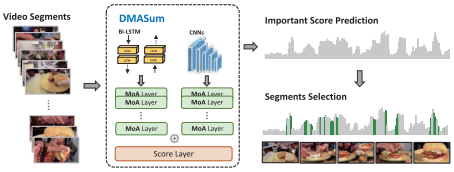


Figure 1: An illustration of the video summarization task using our proposed DMASum. Each gray bar represents the predicted important score of a segment and green bars denote the key-segments in the summarized video. Highlights of DMASum include Visual-sequential Dual Channels, Stacked MoA modules.

1 INTRODUCTION

With the tremendous growth of video materials uploaded to various online video platforms like YouTube, automatic video summarization has received increasing attention in recent years. The summarized video can be used in many scenarios such as fast indexing and human-computer interaction in a light and convenient fashion. The main objective of video summarization is to shorten a whole video into summarized frames while preserving crucial plots. One of the mainstream directions focuses on key-frames summarization [8] is illustrated in Fig. 1 A video is first divided into 15-second segments, and the problem is modeled as an importance score prediction task to select the most informative segments.

The nature of video summarization task encourages a line of research [13, 21, 32, 35] focusing on unsupervised learning methods. Besides, [39] applied deep reinforcement learning with a diversity-representativeness reward function for the generated summary; Currently, the most popular benchmarks are SumMe [8] and TVSum [25]. Otani *et al.* [23] proposed to evaluate the methods by using the rank-order correlation between predicted and human-annotated importance scores. These key evaluation matrices measure agreements between generated summaries and reference summaries. Therefore, supervised methods [7, 32, 36, 38] are still very important for investigating essential technical questions because they can directly compare against human-annotated scores as ground truth. One of the mainstream directions focuses on key-frames summarization [8] is illustrated in Fig. 1.

The challenges for supervised key-frames summarization are two-fold. First, the importance scores are very subjective and highly related to human perception. Second, the annotations are expensive to be obtained; thus, the model should be able to cope with limited labeled data while retaining high generalization. These are not only unsolved questions for video summarization but also essential for many other research domains. To this end, this paper proposes a new framework, namely the Dual Mixture Attention model (DMASum) that aims to achieve **1)** human-like attention by adopting cutting-edge self-attention architecture and takes both visual and sequential information into a unified process; and **2)** high-level semantic understanding of the whole content by incorporating a novel meta learning module to maximally exploit the training data and improve the model generalization.

The proposed framework manifested promising results in our early experiments. However, the early implementation reflected two major technical challenges. The first is known as the *Softmax Bottleneck* problem associated with the self-attention architecture. Both theoretical and empirical evidences in this paper show that traditional softmax function does not have the sufficient capability to retain high-rank representation for long and complex videos. To this end, we propose a *Query Twice* module by adding self-query attention to query-key attention. The Mixture of Attention layer can then compare the two attentions to capture the second-order changes and increase the model capability. The second problem is that the most common meta learning strategy does not naturally fit the video summarization task. We propose a Single-video Meta Learning rule to refrain the learner tasks so as to purify the meta learner updating processes. To summarize our contributions:

- To our best knowledge, this is the first paper that successfully introduces self-attention architecture and meta learning to jointly process dual representations of visual and sequential information for video summarization.
- We provide in-depth theoretical and empirical analyses of the Softmax Bottleneck problem when applying attention model to video summarization task. And a novel self-query module with Mixture-of-Attention is provided as the solution to overcome the problem effectively.
- We explore the meta learning strategy, and a Single-Video Meta Learning rule is particularly designed for video summarization tasks.
- Quantitatively and qualitatively experiments on two datasets: SumMe [8] and TVSum [25] demonstrate our superior performance over the state-of-the-art methods. More impressively, our model achieves human annotator level performance under new protocols of Kendall's τ correlation coefficients and Spearman's ρ correlation coefficients. The groundbreaking results suggested that our DMASum has effectively modeled human-like attention.

2 RELATED WORK

Video summarization. Video, as a media containing complex spatio-temporal relationship of visual contents, has a wide range of applications [4, 5, 28, 29, 33, 41]. However, because of its huge volume, video summarization is to compress such huge volume data into its light version while preserving its information. Early works have presented various solutions to this problem, including storyboards [7, 9, 20, 20] and objects [16, 19, 30]. LSTM-based deep learning approaches are proposed for both supervised and unsupervised video summarization in recent years. Zhang *et al.* [36] proposed a bidirectional LSTM model to predict the importance score of each frame directly, and this model is also extended with determinantal point process [15]. Mahasseni *et al.* [21] specified a generative adversarial framework that consists of the summarizer and discriminator for unsupervised video summarization. The summarizer is an auto-encoder LSTM network for reconstructing the input video, and the discriminator is another LSTM network for distinguishing between the original video and its reconstruction. Meanwhile, based on the observation of Otani *et al.* [23], they propose another evaluation approach as well as a visualization of correlation between the estimated scoring and human annotations.
Attention-based Models. The attention mechanism was born to help memorize long source sentences in neural machine translation [2]. Rather than building a single context vector out of the translation encoder, the attention method is to create shortcuts between the context vector and the entire input sentence, then customize the weights of these shortcut connections for each element. The Transformer [27], without a doubt, is one of the most impressive works in the machine translation task. The model is mainly built on self-attention layers, also known as intra-attention, and the self-attention network is relating different positions of the same input sequence. Many recent works have applied self-attention to a wide range of video-related applications, such as video question answer [18] and video captioning [40]. Particularly for the video summarization task, Ji *et al.* [12] proposed an attention-based encoder-decoder network for selecting the key shots. He *et al.* [11] proposed an unsupervised video summarization method with attentive conditional Generative Adversarial Networks.
Meta Learning. Meta learning, also known as learning to learn, aims to design a model that can be learned rapidly with fewer training examples. Meta learning usually used in few-shot learning [6, 22] and transfer learning [31]. Finn *et al.* [6] propose a Model Agnostic Meta Learning (MAML) which is compatible with any model trained with gradient descent and applicable to a variety of different learning problems, including classification, regression, and reinforcement learning. Like MAML, the work of Nichol *et al.* [22] proposed a strategy which repeatedly sampling and training a single task, then moving the initialization towards the trained weights on that task. Recently, meta learning methods have been applied in a few video analysis tasks. Especially in video summarization, Li *et al.* [17] proposed a meta learning method that explores the video summarization mechanism among summarizing processes on different videos.

3 THE PROPOSED APPROACH

Video summarization is modeled as a sequence labeling (or sequence to sequence mapping) problem. Given a sequence of video frames, the task is to assign each frame an importance score based on which key-frames can be selected. Existing sequence labelling approaches include deep sequential models such as LSTM [36, 37], attention model [12]. However, the key difficulty is to learn the frame dependencies within the video and capturing the internal

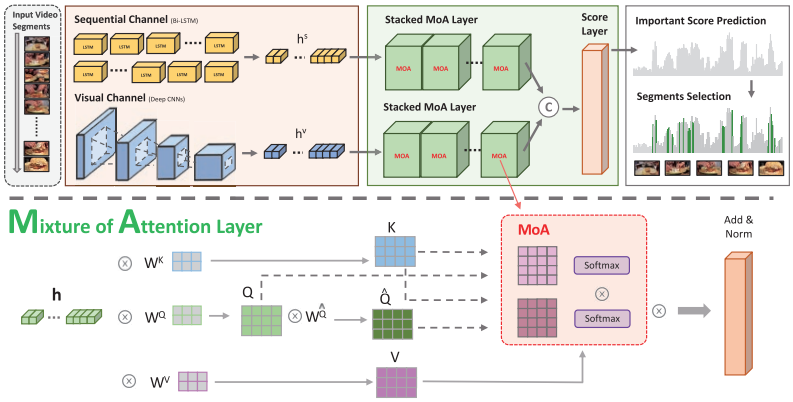


Figure 2: The overall architecture of our DMASum is shown as the top figure, which consists of a sequential channel and a visual channel and stacked MoA layers. The bottom part shows the structure of the Mixture of Attention layer.

contextual information of the video. Considering video is a highly context-dependent source that shares many similar properties in sentences. As the outstanding performance of the Transformer [27], we introduce the self-attention structure that has been widely used in natural language processing (NLP) as our architecture basis. Both visual and sequential representations are considered in order to model complex human-like attention and better match the subjective annotations. Also, the motivation of meta learning aims to improve the model generalization when training sources are insufficient due to expensive human annotations. An overview of the proposed video summarization architecture and the details of the Mixture of Attention layer that are illustrated in Figure 2.

3.1 Architecture Design

Dual-representation Learning: For the video summarization task, we introduce both visual and sequential channels as the input. The visual channel (deep CNNs) extracts visual features $\boldsymbol{H}^v = \{\boldsymbol{h}_t^v\}_{t=1}^T$ from each video frame image. Based on the extracted visual features, the sequential features $\boldsymbol{H}^s = \{\boldsymbol{h}_t^s\}_{t=1}^T$ is obtained by the sequential channel (bidirectional LSTM network) and consists of the dual-channel feature $\boldsymbol{H} \in \{\boldsymbol{H}^v, \boldsymbol{H}^s\}$. The dual representation is critical to model complex human-like attention and can link frame-wise attention to the overall story line.

The Attention Module: Taking a feature sequence $\boldsymbol{H} = \{\boldsymbol{h}_t\}_{t=1}^T \in \mathbb{R}^{D\times T}$ extracted from the video as input, the attention network can re-express each $\boldsymbol{h}_t^*$ within input $\boldsymbol{H}$ by utilizing weighted combination of the entire neighborhood from $\boldsymbol{h}_1$ to $\boldsymbol{h}_T$, where D is the feature dimension and T is number of frames within a video. In concreteness, the attention network first linearly transforms $\boldsymbol{H}$ into $\boldsymbol{Q} = \boldsymbol{W}^Q\boldsymbol{H}^*$, $\boldsymbol{K} = \boldsymbol{W}^K\boldsymbol{H}^*$ and $\boldsymbol{V} = \boldsymbol{W}^V\boldsymbol{H}^*$, where $\boldsymbol{Q} = \{\boldsymbol{Q}_t\}_{t=1}^T \in \mathbb{R}^{D_a\times T}$, $\boldsymbol{K} = \{\boldsymbol{K}_t\}_{t=1}^T \in \mathbb{R}^{D_a\times T}$ and $\boldsymbol{V} = \{\boldsymbol{V}_t\}_{t=1}^T \in \mathbb{R}^{D_a\times T}$ are known as Queries, Keys and Values vectors, respectively and D_a represents the attention feature size, and $\boldsymbol{W}^Q, \boldsymbol{W}^K, \boldsymbol{W}^V \in \mathbb{R}^{D_a\times D}$ are the corresponding learnable parameters. $\boldsymbol{K}$ is employed to learn the distribution of attention matrix on condition of the query matrix $\boldsymbol{Q}$, and $\boldsymbol{V}$ is used to exploit information representation. Thus the scaled dot-product attention $\boldsymbol{A}$ is defined as:

$$\mathcal{F}_{Scale}(\boldsymbol{K}, \boldsymbol{Q}) = \frac{\boldsymbol{K}^T\boldsymbol{Q}}{\sqrt{D_a}}, \tag{1}$$

$$\boldsymbol{A} = \mathcal{F}_{Softmax}(\boldsymbol{K}, \boldsymbol{Q}) = \frac{\exp(\mathcal{F}_{Scale}(\boldsymbol{K}, \boldsymbol{Q}))}{\sum_{t=1}^T \exp\mathcal{F}_{Scale}(\boldsymbol{K}, \boldsymbol{Q})}, \tag{2}$$

where $\boldsymbol{A} \in \mathbb{R}^{T\times T}$ and we consider $\boldsymbol{A}$ as the distribution of attention matrix on condition of the query matrix $\boldsymbol{Q}$. In Eq.1, due to the large degree of high dimensional $\boldsymbol{K}^T\boldsymbol{Q}$, scaling factor $\frac{1}{\sqrt{D_a}}$ is used to prevent the potential small gradient suffered by softmax. The output of attention network is:

$$\boldsymbol{Z} = \boldsymbol{V}\boldsymbol{A}. \tag{3}$$

After applying the attention module to both channels, We concatenate their outputs and feed into a score layer, which consists of multiple fully-connected layers ended with a sigmoid function. The score layer predicts the importance score $\hat{s}$ is sampled as:

$$\hat{\boldsymbol{S}} = \mathcal{F}_{Score}(\mathcal{F}_{Concat}(\boldsymbol{Z}^v, \boldsymbol{Z}^s)), \tag{4}$$

where $\mathcal{F}_{Score}$ denotes the score layer and $\mathcal{F}_{Concat}$ in this paper means concatenation operation on different channels.

Overall Objective Function. We intend to treat the outputs as the importance scores of the whole video frames in this work. Thus, we simply employ the mean square loss $\mathcal{L}$ between the ground truth importance scores and the predicted importance scores.

$$\mathcal{L} = \frac{1}{T}\sum_{i=1}^T (s_i - \hat{s_i})^2, \tag{5}$$

3.2 The Softmax Bottleneck

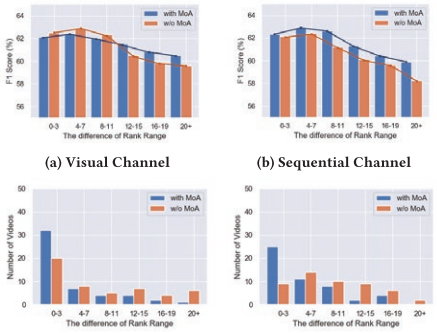
Almost all existing attention models follow the original pipeline from NLP tasks using the softmax function Eq. (2) to compute the attention. However, this section identifies the key limitation of softmax function for video summarization. It can be considered that the attention distribution is a finite set of pairs of a context and its conditional distribution $\mathcal{V} = \{(c_1, P^*(X|c_1)), \ldots, (c_1, P^*(X|c_T))\}$, where $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ denotes T compatible keys in the video $\mathcal{V}$ and $C = \{c_1, c_2, \ldots, c_N\}$ denotes the contexts. It is assumed $P^* > 0$ and $\boldsymbol{A}^*$ represents the true attention distribution. Thus the true attention distribution in (2) can be re-formulated as:

$$\boldsymbol{A}^* = \begin{bmatrix} logP^*(x_1|c_1) & logP^*(x_2|c_1) & \cdots & logP^*(x_T|c_1) \\ logP^*(x_1|c_2) & logP^*(x_2|c_2) & \cdots & logP^*(x_T|c_2) \\ \vdots & \vdots & \ddots & \vdots \\ logP^*(x_1|c_T) & logP^*(x_2|c_T) & \cdots & logP^*(x_T|c_T). \end{bmatrix} \quad (6)$$

The objective of attention model is to learn the conditional attention distribution $P_\theta(\mathcal{X}|C)$ parameterized by θ to match the true attention distribution $P^*(\mathcal{X}|C)$. It can be seen that the attention distribution problem is now turned into a **matrix factorization problem**. Since $\boldsymbol{A}$ is a matrix with size $N \times N$, the rank of learned attention distribution $\boldsymbol{A}$ is upper bounded by the embedding size d. If $d < rank(\boldsymbol{A}^*) - 1$, for any model parameter θ, there exists a context c in $\mathcal{V}$ such that $P_\theta(\mathcal{X}|C) \neq P^*(\mathcal{X}|C)$. This is so called ***Softmax Bottleneck*** which reflects the circumstance when softmax function does not have the capacity to express the true attention distribution when d is smaller than $rank(\boldsymbol{A}^*) - 1$. In the contexts of video summarization, the log probability matrix $\boldsymbol{A}$ becomes a high-rank matrix when the visual contents are complex and the changes between frames are severe. For example, cooking may contain multiple repetitive actions than eating. While humans can intuitively assign equal importance to both of the actions, the former one actually results in a much higher rank in the representation matrix. The softmax function may compromise features from rich content to maintain consistency.

In Figure 3 we empirically verify such a Softmax Bottleneck problem can degrade the performance severely. We choose the TVSum dataset and calculate the difference $\mathcal{D} = T - rank(\boldsymbol{A})$, where T denotes the video length. This is because video lengths are not consistent so we only consider the difference between the actual rank and the full rank T. Lower difference values indicate the attention layer, after softmax, can retain high rank with minimum redundancy. On the other hand, Higher difference values mean the attention matrix of the whole video is low-rank. It can be due to the input video is not complex, e.g. no movement and the background is monotonous. But for most of the cases, the low-rank attention matrix is often resulted by key information missing due to long videos with high complexities. The statistics are collected from attention matrices of both visual and sequential channels. Our key observations are summarised as follows.

(1) From Figure 3 (a) and (b), higher rank representations tend to achieve higher F1 score. But due to the softmax capacity, significant performance drops can be seen in visual (after range 8-11) and sequential channels (after range 4-7), which confirms the existence of bottleneck. In other words, the softmax function cannot retain high-rank information for long complex videos.
(2) From the distribution of video numbers in Figure 3 (c) and (d), many video representations fall out of high-rank range (0-7) after softmax. According to the last observation, these videos are prone to getting lower performances.
(3) The softmax bottleneck problem is more severe on sequential attention, which indicates the changes between frames are the key missing information that results in the lower rank.



(a) Visual Channel (b) Sequential Channel

(c) Visual Channel (d) Sequential Channel

Figure 3: Averaged F1-score (%) and Number of videos with respect to the rank difference $\mathcal{D}$ in TVSum dataset.Blue and Orange bars compare our MoA against traditional softmax.

Motivated by the above insights and inspired by the work of Yang *et al.* [34], we come up with a **Mixture of Attention layer** (MoA) to alleviate the softmax bottleneck issue. We propose the Associated Query $\hat{\boldsymbol{Q}} = tanh(\boldsymbol{W}^{\hat{Q}}\boldsymbol{Q})$, where $\boldsymbol{W}^{\hat{Q}}$ is the Associated Query parameter. The idea is to capture the second-order changes between queries so that both complex and simple contents can be represented in a more smoothed attention representation. The conditional attention distribution is defined as:

$$P(x|c) = \sum_{t=1}^{T} \frac{\exp(\mathcal{F}_{Scale}(\boldsymbol{K}_{c,t}, \boldsymbol{Q}_{c,t}))}{\sum_{t=1}^{T} \exp \mathcal{F}_{Scale}(\boldsymbol{K}_{c,t}, \boldsymbol{Q}_{c,t})} \hat{\boldsymbol{A}}_{c,t} \,, \quad s.t. \sum_{t=1}^{T} \hat{\boldsymbol{A}}_{c,t} = 1 \,, \quad (7)$$

$$where\ \hat{\boldsymbol{A}} = \mathcal{F}_{Softmax}(\boldsymbol{K}, \hat{\boldsymbol{Q}}) \,, \quad (8)$$

In Eq.8, $\hat{\boldsymbol{A}} \in T \times T$ is the associated attention distribution. Thus, MoA formulates the conditional attention distribution as:

$$\boldsymbol{A}_{moa} = \boldsymbol{A}\hat{\boldsymbol{A}}^T \,, \quad (9)$$

where $\boldsymbol{A}_{moa} \in \mathbb{R}^{T \times T}$. In Eq.1, due to the large degree of high dimensional $\boldsymbol{K}^T\boldsymbol{Q}$, scaling factor $\frac{1}{\sqrt{D_a}}$ is used to prevent the potential

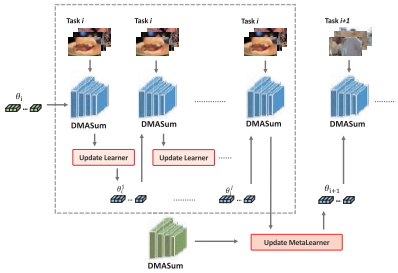


Figure 4: Overview of the i^{th} iteration for update θ_i to θ_{i+1}. There are two stages in this update process. The middle part shows the stage about how the Learner updates θ_i to θ_i^m by iterating m times. The outside parts shows the stage about how the Meta Learner updates θ_i to θ_{i+1}.

small gradient suffered by softmax. As $\boldsymbol{A}_{moa}$ is a non-linear function of the attention distribution, $\boldsymbol{A}_{moa}$ can be arbitrarily higher rank than standard self-attention structure $\boldsymbol{A}$. Thus the output of the mixture of attention network $\boldsymbol{Z} = \boldsymbol{V}\boldsymbol{A}_{moa}$ now can break the bottleneck problem. In Figure 3, after applying the MoA, we can see a large proportion of videos fall into the 0-3 high rank range compared that of traditional softmax. Also, videos especially with lower ranks ($\mathcal{D} > 11$) can be predicted with higher F1 scores. The performance of the sequential channel is boosted, which indicates that all of the previous softmax representations missed high rank information. The smoothed performance drop and increased number of high rank videos serve as strong evidence to manifest the Softmax Bottleneck has been resolved by proposed MoA.

Besides, the DMASum utilizes stacked mixture of attention networks, and in each layer we employ residual dropout connection [10] for allowing gradients to flow through a network directly and layer normalizaion [1] for normalizing the inputs across the features. Overall, the n^{th} layer output can be defined as:

$$Z_n = \mathcal{F}_{Normalize}(\mathcal{F}_{Attention}(Z_{n-1}) \oplus Z_{n-1}), \tag{10}$$

where $\mathcal{F}_{Normalize}$ denotes as layer normalization, $\mathcal{F}_{attention}$ represents the attention layer and $\oplus$ represents the residual connection.

3.3 Single-Video Meta Learning

The key motivation to introduce Meta Learning is to improve the model generalization when the dataset of video summarization is small. Different from gradient descent, the *MetaLearner* is updated by weighted parameters of *Learner* in subtasks, which can be formularized as:

$$Learner^* = MetaLearner(Learner(\tau_i)) \tag{11}$$

where τ_i denotes i^{th} video, *Learner* and *MetaLearner* means the DMASum model in meta learning. We first employ the MAML [6] due to its flexibility and superior performance but did not achieve expected results. Our observation is that in the video summarization context each video has its own latent mechanism that is not shared by different videos. Therefore, we propose a *Single-Video Meta Learning* rule to refrain the learner by only one video at each task. The process is shown as Figure 4.

There are two stages of each epoch in this meta learning strategy. Firstly, to train the task τ_i, the *Learner* updates the parameter θ_i by traditional gradient descent. And, the *Learner* trains the task in a set number m recurrently to explore its latent summarizing context. The equation of updating parameter θ is:

$$\theta_i^j = \theta_i^{j-1} - \alpha\nabla\mathcal{L}_i^j(\mathcal{F}_{\theta_i^{j-1}}), \quad where j = 1 \dots m \tag{12}$$

where α denotes learning rate and ∇ denoted as the gradient, and $\mathcal{F}_\theta$ is the loss function on i^{th} task. After j^{th} iteration, the *MetaLearner* updates the parameter θ_{i+1} by using the parameter θ_i^m of the *Learner* by:

$$\theta_i = \theta_{i-1} - \beta\nabla\mathcal{L}_i(\mathcal{F}_{\theta_i^m}), \tag{13}$$

where β is the learning rate of the *Learner*. θ_i updated state of *Learner* after the j^{th} iteration in *MetaLearner*. Overall, our meta learning is summarized in Algorithm 1. Note that in the last step of the algorithm, we treat θ_i^m - θ_i as a gradient and plug it into Adam instead of simply updating θ_i in the direction θ_i^m - θ_i.

Algorithm 1: Meta learning in DMASum

```
/* θ : Parameter of Learner;                          */
/* α : Learning rate in Learner;                      */
/* β : Learning rate in MetaLearner;                  */
/* n : The number of videos;                          */
/* m : Recurrent training Learner number;             */
/* F : the DMASum model;                              */
Initialize: θ
for k = 1 to epoch number do
    for i = 1 to n do
        Sample video i as task τ_i
        for j = 1 to m do
            θ_i^j = θ_i^{j-1} − α∇L_i^j(F_{θ_i^{j-1}})
        Update θ_{i+1} ← θ_i + β(θ_i^m - θ_i)
```

4 EXPERIMENTS

4.1 Experiment Setup

Datasets. We evaluate our model on two datasets: SumMe [8] and TVSum [25]. SumMe consists of 25 videos covering a variety of events, such as sports and cooking. The duration of each video varies from 1 to 6.5 minutes. TVSum contains 50 videos downloaded from Youtube, which are selected from 10 categories. The video length varies from 1 to 10 minutes. Both datasets include ego-centric and third-person camera views, and the annotations were labeled by 25 human annotators. We also exploit two auxiliary datasets to augment the training data, where Open Video Project[1] (OVP) contains 50 videos and Youtube [3] contains 39 videos.

[1]Open video project: https://open-video.org

Evaluation Metrics. We follow the commonly used protocol from [36] and converted the importance scores to shot-based summaries for both datasets, and the user annotations are changed from frame-level scores to key-shots scores using the kernel temporal segmentation (KTS) [24] method, which can temporally segment a video into disjoint intervals. We then compute the harmonic mean F-score as the evaluation metric. In addition, according to the recent evaluation protocol [23], we apply Kendall's τ [14] and Spearman's ρ [42] correlation coefficients for comparing the ordinal association between generated summaries and the ground truth (i.e. the relationship between rankings). Also, they provided correlation curves to visualize the predicted importance score ranking with respect to the reference annotations, i.e., when the predicted importance scores are perfectly concordant with averaged human-annotated scores, the curve lies on the upper bound of the light-blue area. Otherwise, the curve coincides with the lower bound of the area when the ranking of the scores is in reverse order of the reference.
Evaluation Settings. Following [36], we conducted the experiments under three settings. (1) Canonical (C): we used the standard 5-fold cross-validation (5FCV) for SumMe and TVSum datasets. (2) Augmented (A): we used OVP and YouTube datasets to augment the training data in each fold under the 5FCV setting. (3) Transfer (T): we set a target testing dataset, e.g., SumMe or TVSum, and used the other three as the training data.
Implementation details. To be consistent with existing methods, the 1024 dimensional visual features extracted from the *pool5* layer of the GoogLeNet [26] are used for training. To extract the temporal features, we design a Bi-LSTM model in the proposed network, as a two-layer LSTM with 512 hidden units per layer. For each attention layer, we set the attention dimension as 1024. We stack four attention layers for visual feature attention pipeline, and two layers for the sequential feature attention pipeline. The score layer consists of two fully-connected layers with 1024 hidden units. For Single-video Meta Learning, we set the learning rate of *Learner* as 3×10^{-5} and the learning rate of *MetaLearner* as 6×10^{-5}. Moreover, the recurrent training Learner number is set as 3 and 5 in SumMe and TVSum datasets respectively. During the test, we follow the strategy of prior work [21, 36, 39] to generate the summary. In addition, we employ the ADAM optimizer to train our network and the hyperparameters are optimized via cross-validation.

4.2 Quantitative Evaluation

We first compare our method with state-of-the-art supervised approaches in three evaluation settings. Then, we re-implement the VS-LSTM, SUM-GAN, and DR-DSN models, and quote results for other methods from [11–13, 17, 23, 32, 35]. An in-depth ablation study is then provided to better understand of our DMASum.
Comparison with State-of-the-art Methods. Our main comparison with state-of-the-art methods is summarized in Table 1. The compared methods can be mainly categorized into LSTM, GAN, Attention, and meta learning models. M-AVS [12] and ACGAN [11] are based-on attention models and MetaL-TDVS [17] is based on meta learning. It can be seen that DMASum outperforms other approaches on both datasets consistently. The F1-score results can reflect that our attention mechanism with meta learning can better predict importance scores.

Table 1: F1-score (%) of DMASum with state-of-the-art approaches on both SumMe and TVSum dataset.

Method	SumMe	TVSum
DPP-LSTM [36]	38.6	54.7
SASUM [32]	45.3	58.2
SUM-GAN [21]	41.7	54.3
Cycle-SUM [35]	41.9	57.6
DR-DSN [39]	42.1	58.1
MetaL-TDVS [17]	44.1	58.2
ACGAN [11]	46.0	58.5
CSNet [13]	51.3	58.8
M-AVS [12]	44.4	61.0
DMASum	**54.3**	**61.4**

Table 2: Rank-order correlation coefficients computed between predicted importance scores by different models and human-annotated scores on both SumMe and TVSum datasets using Kendall's τ and Spearman's ρ correlation coefficients.

Method	SumMe		TVSum	
	τ	ρ	τ	ρ
Random	0.000	0.000	0.000	0.000
DPP-LSTM [36]	-	-	0.042	0.055
SUM-GAN [21]	0.049	0.066	0.024	0.031
DR-DSN [39]	0.028	-0.027	0.020	0.026
Human	**0.227**	**0.239**	0.178	0.205
DMASum	0.063	0.089	**0.203**	**0.267**

We also evaluate our DMASum by using the most recent rank-order statistics [23]. The new evaluation matrix can also consider the frame dependencies and annotator consistency so as to reflect the true importance better. Because, F1 score can partially reflect the consistency between prediction and importance scores due to large variations in segment length (i.e. two-peak, KTS, and randomized KTS). The correlation coefficients (Kendall's τ and Spearman's ρ) can be used to measure the similarity between the implicit ranking provided by the frame-level importance score of the generated frame annotation and the human annotation. From Table 2, We can see the correlation coefficients given by DMASum are significantly higher than other state-of-the-art models. More importantly, the performance on the TVSum dataset (0.233 and 0.267) is even better than human annotators (0.205 and 0.267). We believe it is because the dual-channel attention mechanism itself is simulating human behavior and memorizing visual and sequential sources of information and the meta learning method could learn the latent mechanism of summarizing a video story. Also, different human annotators might pay different attention to the given video. Our model can summarize the information from multiple human annotators so that the learned attention-based model is moderated and can achieve better consistency. Figure 5 demonstrates two examples of the correlation coefficients. Curves above the random importance scores in the black dash are positive with better consistency. Our

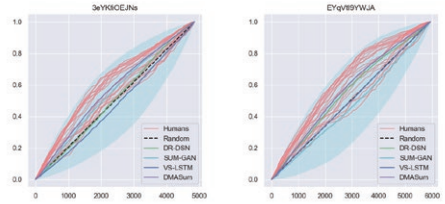


Figure 5: Example correlation curves produced for two videos from the TVSum dataset (3eYKfiOEJNs and EYqVtl9YWJA are video ids). The red lines represent correlation curves for 25 human annotators and the black dashed line is the expectation for a random importance score. The magenta curve shows the corresponding result.

Table 3: F1-score (%) of ablation study on SumMe and TVSum datasets. There are five ablation models: DMASum$_{wom}$ (without meta learning strategy), DMASum$_{softmax}$ (with standard softmax function in self-attention network), DMASum$_{v}$ (without sequential channel), DMASum$_{s}$ (without visual channel), DMASum$_{b}$ (with multiple videos in a batch), and DMASum$_{maml}$ (with MAML)

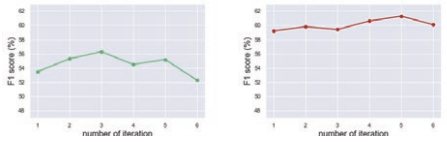
Method	SumMe	TVSum
DMASum$_{wom}$	51.6	60.6
DMASum$_{softmax}$	50.6	60.1
DMASum$_{v}$	53.2	60.5
DMASum$_{s}$	53.3	61.0
DMASum$_{b}$	51.3	60.0
DMASum$_{maml}$	49.3	59.2
DMASum	**54.3**	**61.4**

model achieves averaged performance among all human annotators and outperforms the other compared methods.

4.3 Ablation study.

The success of our DMASum ascribes to both the framework design and technical improvement in each module. To analyze the effect of each component in DMASum, we conduct six ablation study models including DMASum without single video meta learning (DMASum$_{wom}$), DMASum with standard softmax function in self-attention network (DMASum$_{softmax}$), DMASum without sequential channel (DMASum$_{v}$), DMASum without visual channel (DMASum$_{s}$), DMASum$_{b}$ is developed with the batch version of Reptile, and DMASum is designed with MAML (DMASum$_{maml}$). Results are summarised in Table 3, from which we can understand the following questions.

The Basis of Self-attention Architecture provided the initial performance boost. By removing our meta learning module, we can make a straight comparison with state-of-the-art DPP-LSTM [36] and M-AVS [12] which are using no attention and normal attention modules. Averaged improvement is around 5% to 10%. Note that M-AVS is slightly better than our method on the TVSum dataset due to their extra autoencoder architecture.



(a) SumMe Dataset **(b) TVSum Dataset**

Figure 6: Different recurrent training Learner number with respect to the F1-score (%) in DMASum on both SumMe and TVSum datasets.

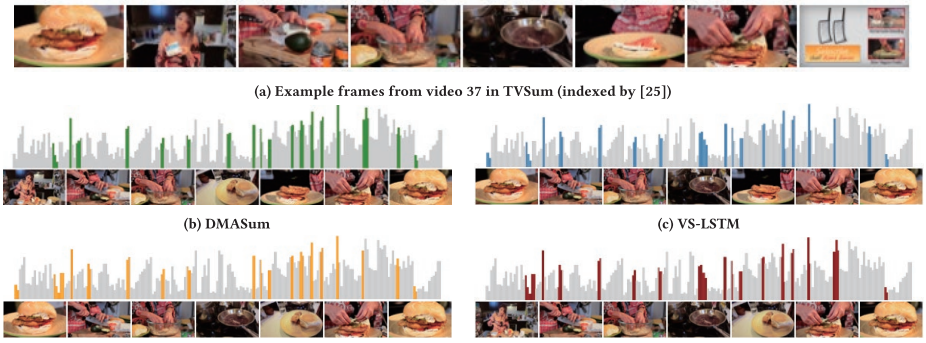
The Softmax Bottleneck problem results in severe performance gaps. By replace the MoA back to traditional softmax function, the performance drops 3.7% and 1.3% respectively on the two datasets. A more detailed analysis has been discussed in Section 3, from where we can see the problem is more critical when video contents are long and complex, involving rich sequential information.

Visual vs Sequential Representation. By comparing the performance of DMASum$_{v}$ or DMASum$_{s}$, we can observe that: 1) In TVSum dataset, the DMASum$_{v}$ gained a slightly better performance than DMASum$_{s}$. 2) The performance in SumMe dataset benefits more from the sequential channel. The self-attention network can effectively connect visual features from frames and the sequential information for the whole story line and thus our combined DMASum achieves better results.

The Necessity of Meta Learning. Removing the meta learning can heavily affect the performance by 2.7% on SumMe dataset. The key reason is that SumMe is a relatively small dataset. This observation serves as strong evidence to validate the motivation and necessity of our meta learning module.

MAML, Batch, and Single Video Meta Learning. The Single Video rule is the key finding that distinguish it from meta learning in other applications, e.g. few-shot learning. This is due to a video itself is rich and complex. By increasing each meta learning task from one video to three in a batch, the performance of DMASum$_{b}$ drops 3% and 1.4% with the clearly slowed training process. In addition, we can see that the performance of our proposed meta learning strategy is better than the batch version of the Reptile strategy, and the batch version of the Reptile strategy is time-consuming during the training process. The efficiency of Single-Video rule is also validated by comparing it to DMASum$_{maml}$.

Number of Recurrent Learning. In a controlled experiment, we observe that when the recurrent training Learner number is 3 for SumMe Dataset and 5 for the TVSum dataset, the F-score reaches the highest shown from Figure 6. Which means, the Learner might not learn the summarizing mechanism when the number is too low, and when the number is too high, the Learner might overfit the current video. In this paper, the number of recurrent training is automatically chosen by using the standard 5-fold cross validation.



(a) Example frames from video 37 in TVSum (indexed by [25])

(b) DMASum

(c) VS-LSTM

(d) SUM-GAN

(e) DR-DSN

Figure 7: Quantitative results of different approaches for video 16 in TVSum. In (b) to (e), the light-gray bars represent the ground truth importance scores, and the colored bars correspond to the selected frames by different methods.

Table 4: F-score (%) of approaches in canonical, augmented and transfer settings on SumMe and TVSum datasets.

Method	SumMe			TVSum		
	C	A	T	C	A	T
DPP-LSTM [36]	38.6	42.9	40.7	54.7	59.6	58.7
SUM-GAN [21]	41.7	43.6	-	54.3	**61.2**	-
DR-DSN [39]	42.1	43.9	42.6	58.1	59.8	58.9
CSNet [13]	51.3	52.1	45.1	58.8	59.0	59.2
DMASum	**54.3**	**54.1**	**52.2**	**61.4**	**61.2**	**60.5**

Comparison under Different Settings. Another approach to examining the model generalization is to investigate its performance under different task settings. Table 4 shows the experimental results of the comparison between the DMASum and cited results of state-of-the-art approaches in canonical, augmented and transfer settings. Note that even though the performance of our model in augmented and transfer settings are partially better than the best results. We observe that the given importance scores in Youtube and OVP datasets are either 0 or 1. However, the DMASum is learning by the importance scores within the range of zero to one from SumMe and TVSum datasets. Such discrepancy of importance score format in both Youtube and OVP datasets would cause the meta learning strategy to be ineffective or even counterproductive because our model is not tailored to handle the discrepancy in labels. Thus in the future, we can improve our framework to adapt to this situation. But on the positive side, our DMASum is still capable in both augmented and transfer settings and achieves comparable results to that of state-of-the-art models despite the above difficulties.

4.4 Qualitative Evaluation

To better illustrate the important frames selection of different approaches, we provide qualitative results for an exemplary video in Figure 7, which tells a story of how to cook a burger. Overall, we can observe that all summaries generated by the different models can cover the intervals with high importance scores. Moreover, according to the figure, the summaries produced by both our DMASum and SUM-GAN contain more peaks, which proves that our proposed model can effectively capture key-frames from the original video. Also, the summary of our model is more sparse and much closer to the entire storyline, i.e., the different cooking stages, which means our meta learning strategy can learn the latent mechanism of summarizing a video.

5 CONCLUSION

We have presented the first work to introduce self-attention meta learning architecture to estimate the visual and sequential attentions jointly for video summarization. The self-attention formula was derived into a matrix factorization problem and key technical Softmax Bottleneck has been identified with both theoretical and empirical evidences. Our work also confirmed the importance of high-rank representation for video summarization tasks. A novel MoA module was proposed to replace the softmax, which can compare twice by query-key and self-query attentions. The Single-Video Meta Learning rule was designed and particularly tailored for video summarization tasks and significantly improved off-the-shelf Meta Learning, e.g. MAML. On two public datasets, our DMASum outperforms other methods in terms of both F1-score and achieved human-level performance using rank-order correlation coefficients. Future work could focus on further improve the generalisation for cross-dataset settings using an integrated framework.

ACKNOWLEDGMENTS

Bingzhang Hu and Yu Guan are supported by Engineering and Physical Sciences Research Council (EPSRC) Project CRITiCaL: Combatting cRiminals In The CLoud (EP/M020576/1). Yang Long is supported by Medical Research Council (MRC) Fellowship (MR/S003916/1).

REFERENCES

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[3] Sandra Eliza Fontes De Avila, Ana Paula Brandão Lopes, Antonio da Luz Jr, and Arnaldo de Albuquerque Araújo. 2011. VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters* 32, 1 (2011), 56–68.

[4] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. 2019. Temporal cycle-consistency learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1801–1810.

[5] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. Slowfast networks for video recognition. In *Proceedings of the IEEE international conference on computer vision*. 6202–6211.

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1126–1135.

[7] Boqing Gong, Wei-Lun Chao, Kristen Grauman, and Fei Sha. 2014. Diverse sequential subset selection for supervised video summarization. In *Advances in Neural Information Processing Systems*. 2069–2077.

[8] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. 2014. Creating summaries from user videos. In *European conference on computer vision*. Springer, 505–520.

[9] Michael Gygli, Helmut Grabner, and Luc Van Gool. 2015. Video summarization by learning submodular mixtures of objectives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3090–3098.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[11] Xufeng He, Yang Hua, Tao Song, Zongpu Zhang, Zhengui Xue, Ruhui Ma, Neil Robertson, and Haibing Guan. 2019. Unsupervised Video Summarization with Attentive Conditional Generative Adversarial Networks. In *Proceedings of the 27th ACM International Conference on Multimedia*. 2296–2304.

[12] Zhong Ji, Kailin Xiong, Yanwei Pang, and Xuelong Li. 2019. Video summarization with attention-based encoder-decoder networks. *IEEE Transactions on Circuits and Systems for Video Technology* (2019).

[13] Yunjae Jung, Donghyeon Cho, Dahun Kim, Sanghyun Woo, and In So Kweon. 2019. Discriminative Feature Learning for Unsupervised Video Summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 8537–8544.

[14] Maurice G Kendall. 1945. The treatment of ties in ranking problems. *Biometrika* 33, 3 (1945), 239–251.

[15] Alex Kulesza, Ben Taskar, et al. 2012. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning* 5, 2–3 (2012), 123–286.

[16] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman. 2012. Discovering important people and objects for egocentric video summarization. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1346–1353.

[17] Xuelong Li, Hongli Li, and Yongsheng Dong. 2019. Meta Learning for Task-Driven Video Summarization. *IEEE Transactions on Industrial Electronics* (2019).

[18] Xiangpeng Li, Jingkuan Song, Lianli Gao, Xianglong Liu, Wenbing Huang, Xiangnan He, and Chuang Gan. 2019. Beyond RNNs: Positional Self-Attention with Co-Attention for Video Question Answering. In *The 33rd AAAI Conference on Artificial Intelligence*, Vol. 8.

[19] David Liu, Gang Hua, and Tsuhan Chen. 2010. A hierarchical visual model for video object summarization. *IEEE transactions on pattern analysis and machine intelligence* 32, 12 (2010), 2178–2190.

[20] Zheng Lu and Kristen Grauman. 2013. Story-driven summarization for egocentric video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2714–2721.

[21] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. 2017. Unsupervised video summarization with adversarial lstm networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 202–211.

[22] Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999* (2018).

[23] Mayu Otani, Yuta Nakashima, Esa Rahtu, and Janne Heikkila. 2019. Rethinking the Evaluation of Video Summaries. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7596–7604.

[24] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. 2014. Category-specific video summarization. In *European conference on computer vision*. Springer, 540–555.

[25] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. 2015. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5179–5187.

[26] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[28] Junyan Wang, Bingzhang Hu, Yang Long, and Yu Guan. 2019. Order Matters: Shuffling Sequence Generation for Video Prediction. In *Proc. BMVA British Mach. Vis. Conf.* 275.1–275.14.

[29] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. 2019. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1328–1338.

[30] Xi Wang, Yu-Gang Jiang, Zhenhua Chai, Zichen Gu, Xinyu Du, and Dong Wang. 2014. Real-Time Summarization of User-Generated Videos Based on Semantic Recognition. In *Proceedings of the 22nd ACM International Conference on Multimedia* (Orlando, Florida, USA) *(MM âĂŹ14)*. Association for Computing Machinery, New York, NY, USA, 849âĂŞ852. https://doi.org/10.1145/2647868.2655013

[31] Yu-Xiong Wang and Martial Hebert. 2016. Learning to learn: Model regression networks for easy small sample learning. In *European Conference on Computer Vision*. Springer, 616–634.

[32] Huawei Wei, Bingbing Ni, Yichao Yan, Huanyu Yu, Xiaokang Yang, and Chen Yao. 2018. Video summarization via semantic attended networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[33] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. 2019. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 284–293.

[34] Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W Cohen. 2017. Breaking the softmax bottleneck: A high-rank RNN language model. *arXiv preprint arXiv:1711.03953* (2017).

[35] Li Yuan, Francis EH Tay, Ping Li, Li Zhou, and Jiashi Feng. 2019. Cycle-SUM: Cycle-consistent Adversarial LSTM Networks for Unsupervised Video Summarization. *arXiv preprint arXiv:1904.08265* (2019).

[36] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. 2016. Video summarization with long short-term memory. In *European conference on computer vision*. Springer, 766–782.

[37] Ke Zhang, Kristen Grauman, and Fei Sha. 2018. Retrospective encoders for video summarization. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 383–399.

[38] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. 2018. Hsa-rnn: Hierarchical structure-adaptive rnn for video summarization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7405–7414.

[39] Kaiyang Zhou, Yu Qiao, and Tao Xiang. 2018. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[40] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. 2018. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8739–8748.

[41] Yi Zhu, Yang Long, Yu Guan, Shawn Newsam, and Ling Shao. 2018. Towards universal representation for unseen action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9436–9445.

[42] Daniel Zwillinger and Stephen Kokoska. 1999. *CRC standard probability and statistics tables and formulae*. Crc Press.

An Accurate Segmentation-Based Scene Text Detector with Context Attention and Repulsive Text Border

Xi Liu, Gaojing Zhou, Rui Zhang, Xiaolin Wei
Meituan-Dianping Group, Beijing, China
{liuxi12, zhougaojing, zhangrui36, weixiaolin03}@meituan.com

Abstract

Scene text detection is one of the most challenging problems in computer vision and has attracted great interest. In general, scene text detection methods are divided into two categories: detection-based and segmentation-based methods. Recently, the segmentation-based methods are more and more popular due to their superior performances and the advantages of detecting arbitrary-shape texts. However, there still exist the following problems: (a) the misclassification of the unexpected texts, (b) the split of long text lines, (c) the failure of separating very close text instances. In this paper, we propose an accurate segmentation-based detector, which is equipped with context attention and repulsive text border. The context attention incorporates global channel attention, non-local self-attention and spatial attention to better exploit the global and local context, which can greatly increase the discriminative ability for pixels. Due to the enhancement of pixel-level features, false positives and the misdetections of long texts are reduced. Besides, for the purpose of solving very close text instance, a repulsive pixel link, which focuses on the relationships between pixels at the border, is proposed. Experiments on several standard benchmarks, including MSRA-TD500, ICDAR2015, ICDAR2017-MLT and CTW1500, validate the superiority of the proposed method.

1. Introduction

Scene text detection, which refers to precisely localizing all the instances of texts in a scene image, has been widely studied. It is a critical step in many text-related real-world applications, such as photo translation [1], autonomous driving, image retrieval [14] and augmented reality. It is quite challenging due to the large variations of color, size, aspect ratio, font, orientation, lighting conditions and background in scene texts [54].

With the development of deep learning, great progress has been made in the computer vision tasks such as object detection and segmentation [9, 10, 13, 21, 25, 42, 44, 45]. Scene text detection, which can be seen as a type of object detection applied to text, has also witnessed great success [11, 22, 23, 26, 27, 28, 30, 32, 33, 43, 59, 61]. In general, scene text detection methods can be divided into two categories: detection-based and segmentation-based methods. The detection-based methods adapt the general object detection framework to detect the text or text parts by directly regressing rectangles or quadrangles with certain orientations. However, these frameworks cannot detect the text instances with arbitrary shapes and often fail to detect small texts. The segmentation-based methods use pixel-wise segmentation to segment text areas and extract text instances by post-processing the segmented areas. They have gained more interest due to their advantages of detecting arbitrary-shape texts and the superior performances compared with detection-based methods. However, there still exist several problems. The first one is the misclassification of the unexpected texts or text-like patterns. The second one is the split of the long text line into several text instances. The third one is the failure of separating very close text instances. Some examples are shown in Fig. 1 (a)(b)(c).

Figure 1. Different types of problems in scene text detection and the results of our method. Note that the error detections are marked with Red boxes. (a) is the misclassification of the unexpected texts, (b) is the split of long text lines, (c) is the failure of separating very close text instances. (d), (e), (f) are the results of our method, which successfully solves the problems.

To address these problems, in this paper, we propose an accurate segmentation-based text detector. Two modules:

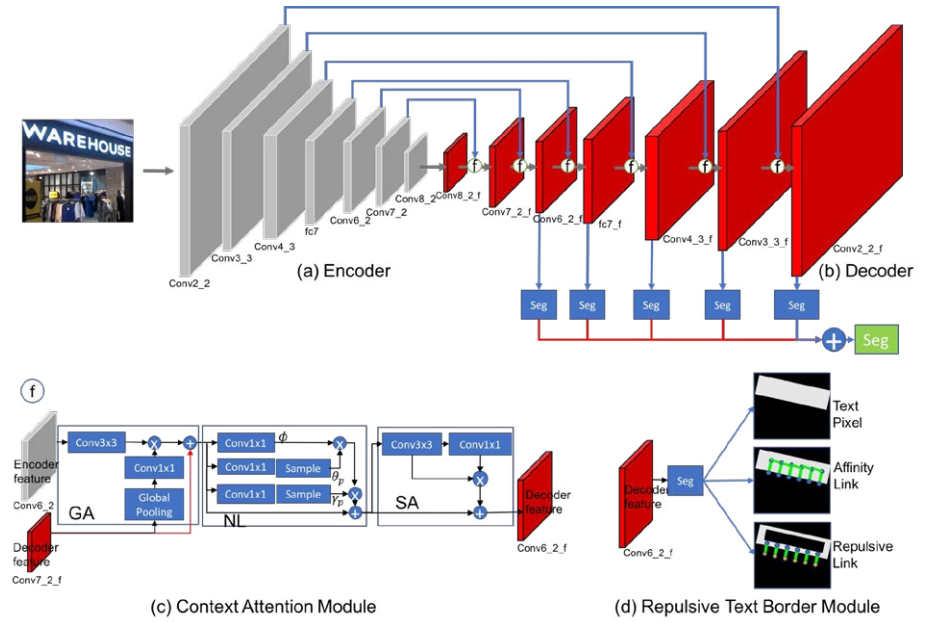


Figure 2. Architecture of the proposed method. The network consists of (a) Encoder, (b) Decoder, (c) Context Attention Module, GA is global attention, NL is non-local self-attention, and SA is spatial attention, (d) Repulsive Text Border Module. The red arrow line means upsample.

context attention module and repulsive text border module are specifically introduced. First, context plays a critical role in segmentation since it is very helpful for reducing local ambiguities for pixel classification. We design an effective attention mechanism to better exploit the context information by sequentially applying global attention, non-local self-attention and spatial attention. The global attention uses global average-pooled high-level decoder features to compute channel-wise attention to the low-level encoder features, which increases the discriminative ability of low-level features. Non-local attention mechanism is proved to be effective for capturing long range dependencies. For the long text lines detection, long range contextual information is necessary to avoid the split of long text line into several text instances. We use a simple yet effective non-local module introduced in the work of [60] as our non-local attention module. It embeds a pyramid sampling module into non-local blocks to largely reduce the computation. The spatial attention utilizes the local inter-spatial relationship of features and focuses on 'where' is text, which further solve the false positives. It applies a convolution layer with one channel to generate a spatial - attention map and enhances the input features by broadcasting the attention map. As shown in Fig. 1(d)(e), our method can successfully solve the false positive and the split of long text. Second, text border is key to separating very close text lines. In PixelLink [3], it learns two kinds of pixel-wise predictions: text/non-text prediction and link prediction. The pixel link is important for separating text instance since texts are detected by linking pixels within the same text instance. However, the pixel link generally pays attention to the link between neighbor pixels that belong to the same text instance. Note that the link between pixels located at the text border requires more attention. Therefore, we introduce an extra repulsive pixel link that explicitly represents the relationship between two pixels at the text border. Predicted positive pixels are then joined together by predicted positive pixel links and negative repulsive links. Fig. 1(f) is the result of our method which shows that the very close text instances can be separated.

To validate the effectiveness of our proposed scene text detector, we conduct extensive experiments on four standard benchmarks and achieve an F-measure of 86.1%

on MSRA-TD500, 87.5% on ICDAR2015, 75.3% on ICDAR2017-MLT, 82.0% on CTW1500. The experimental results show that our method outperforms most of the state-of-art methods. The contributions of this paper can be summarized as follows:

(1) We propose an effective attention mechanism to better exploit the context information, which can effectively reduce the false positives and avoid the split of long text line into several text instances.

(2) To further solve the very close text instance, we propose to learn an extra repulsive pixel link that explicitly represents the relationship between pixels located at text border.

(3) The proposed method achieves state-of-the-art performance on several benchmark datasets of scene text including long straight, horizontal, multi-oriented and curved text.

2. Related Work

Scene text detection has been extensively studied in the last decades. State-of-the-art text detection algorithms are deep neural network based methods. Most of the deep learning based text detection methods can roughly be divided into two branches: detection-based and segmentation-based approaches.

Detection-based methods treat text as a specific object and take advantage of the development in general object detection. Zhong et al. [57] proposed a text detection framework based on Faster-RCNN. They designed an inception-RPN which used multi-scale convolution filters to produce text region proposals. Ma et al. [34] added rotation to both anchors and RoIPooling in Faster R-CNN, to deal with the orientation of scene text. Gupta et al. [6] borrowed the YOLO [41] framework and employed a fully-convolutional regression network to perform text detection and bounding box regression at all locations and multiple scales of an image. TextBoxes [22] modified anchors and kernels of SSD to detect large aspect-ratio scene text. TextBoxes++ [20] extended TextBoxes by regressing quadrilaterals instead of horizontal bounding boxes to handle arbitrary-oriented text. Shi et al. [43] employed SSD framework and learned the locally detectable text elements, namely segments and links. RRD [23] also relied on SSD framework and introduced rotation-sensitive feature for detection branch and rotation-invariant feature for classification branch to learn better regression of long oriented text. These methods always need complex anchor setting and fail to detect texts with arbitrary shapes.

Segmentation-based methods are mostly inspired by fully convolutional networks (FCN) [31]. Zhang et al. [56] first presented a framework which used FCN to produce a coarse saliency map for text. Yao et al. [53] casted the detection task as a segmentation problem by predicting three kinds of score maps: text/non-text, character classes, and character linking orientations. PixelLink [3] performed pixel-wise text/non-text and link prediction, then added some post-processing on the linked positive pixels to obtain the final text boxes. PSENet [18] used FCN to predict text instances with multiple scales, then designed a progressive scale expansion algorithm to reconstruct the whole text instance. More recently, several works such as Mask Text Spotter [32] and SPCNet [50] borrowed the state-of-art instance segmentation approach Mask R-CNN to detect text instances and achieved impressive performance. The biggest advantage of these methods is the ability to extract arbitrary-shape texts. However, their performances are greatly affected by the segmentation results.

Compared with previous works, our method incorporates context attention and repulsive text border to improve text detection performance. Relying on the context information, the misclassification of the unexpected texts or text-like patterns and the split of long text lines are greatly reduced, which are common issues for most of segmentation-based methods. Moreover, the proposed repulsive pixel link that explicitly represents the relationship between two pixels at the text border are verified to be effective for separating the very close text instances.

3. Approach

In this section, we describe our proposed method in detail. Firstly, we present the general framework of our method. Secondly, we elaborate the context attention and repulsive text border modules. Finally, the training and inferring details are presented.

3.1. Overall Architecture

The network architecture of our approach is illustrated in Fig. 2. It is based on a fully convolutional network with encoder-decoder structure. In the encoder part, VGG-16 is used as backbone and the last two layers fc6 and fc7 are converted from fully-connected layers into convolutional layers. Besides, three extra layers are added after fc7 layer in the same manner as SSD [25]. In the decoder part, the output feature maps are generated by fusing low-level decoder features with high-level encoder features. The fusing process is implemented by introducing a context attention module. As shown in Fig. 2(c), the context attention module uses global attention, non-local self-attention and spatial attention to effectively model the local and global context, which will be detailed in Section 3.2. For each output feature map of the decoder (conv2_2_f, conv3_3_f, conv4_3_f, fc7_f, conv6_2_f), three sibling 1x1 convolution and softmax layers are attached to generate three score maps for text pixel, affinity pixel link and repulsive pixel link (see Fig. 2(d)). Since every pixel has 8 neighbors, the output score maps have 2, 16 and 16 channels, respectively. The details of learning pixel links are presented in Section 3.3. Finally, the score maps of each output feature map are resized and added together to obtain

three segmentation masks: text pixel mask, affinity link and repulsive link masks. Based on the segmentation results, we join the positive pixels with positive pixel links and negative repulsive links together, and obtain the detection results by extracting the bounding boxes of the connected components.

3.2. Context Attention

Context plays a critical role in segmentation since it is helpful for reducing local ambiguities for pixel classification. In our context attention, there are three sub-modules: global attention, non-local self-attention and spatial attention. Given the low-level encoder feature map $F_{low} \in \mathbb{R}^{C\times H\times W}$ and the high-level decoder feature map $F_{high} \in \mathbb{R}^{C'\times H'\times W'}$ as input, the context attention module sequentially goes through 1D channel attention, non-local self-attention and 2D spatial attention to generate the output feature map $F_{CA} \in \mathbb{R}^{C'\times H\times W}$, as illustrated in Fig.2(c). The overall process can be summarized as:

$$F_{GA} = GA(F_{low}, F_{high}), \quad (1)$$

$$F_{NL} = NL(F_{GA}), \quad (2)$$

$$F_{CA} = F_{SA} = SA(F_{NL}), \quad (3)$$

where $GA(\cdot)$ is global attention, $NL(\cdot)$ is non-local self-attention, and $SA(\cdot)$ is spatial attention.

Global Attention Module. High-level features always contain rich text category information, which can be a good guidance for low-level features to select text localization details.

We perform global average pooling on the high-level decoder features $F_{high} \in \mathbb{R}^{C'\times H'\times W'}$ and a 1×1 convolution over the pooled features to generate the global attention map. The low-level encoder features $F_{low} \in \mathbb{R}^{C\times H\times W}$ are then multiplied by the attention map. Note that the channel number of the attention map and the low-level features may be different. A 3×3 convolution is added to the low-level features. Finally, the high-level features are upsampled and added with the weighted low-level features to get the output features $F_{GA} \in \mathbb{R}^{C'\times H\times W}$. In short, the output feature is computed as:

$$\begin{aligned} F_{GA} &= GA\left(F_{low}, F_{high}\right) \\ &= GAttMap \odot Conv_{3\times3}(F_{low}) + UP\left(F_{high}\right), \end{aligned} \quad (4)$$

$$GAttMap = Conv_{1\times1}\left(AvgPool\left(F_{high}\right)\right), \quad (5)$$

where $\odot$ represents element-wise multiplication, $UP(\cdot)$ is upsample operation.

Non-local Self-Attention Module. Non-local attention is potent to capture the long range dependencies that are crucial for pixel classification. Especially for the long text lines, long range contextual information is necessary to avoid the split of long text line into several text instances.

Considering the large computation of non-local operation, we use a simple yet effective non-local module introduced in the work of [60]. Given the output feature $F_{GA} \in \mathbb{R}^{C'\times H\times W}$ of the global attention module as input, three 1x1 convolutions are first used to transform the input to different embeddings: $\phi \in \mathbb{R}^{\hat{C}\times H\times W}$, $\theta \in \mathbb{R}^{\hat{C}\times H\times W}$ and $\gamma \in \mathbb{R}^{\hat{C}\times H\times W}$. Spatial pyramid pooling is then applied after θ and γ to get sampled θ_p and γ_p.

The ϕ, θ_p and γ_p are flattened to $\phi \in \mathbb{R}^{\hat{C}\times N}$, $\theta_p \in \mathbb{R}^{\hat{C}\times S}$, $\gamma_p \in \mathbb{R}^{\hat{C}\times S}$. A normalized similarity matrix is calculated as:

$$\overrightarrow{V_p} = f(\phi^T \times \theta_p), \quad (6)$$

where the normalizing function f can take the form from softmax, rescaling, and none. The attention output is acquired by

$$O_p = \overrightarrow{V_p} \times {\gamma_p}^T, \quad (7)$$

and the final output $F_{NL} \in \mathbb{R}^{C'\times H\times W}$ is given by

$$F_{NL} = Reshape(W_o(O_p^T) + F_{GA}), \quad (8)$$

where W_o is a 1x1 convolution operation to recover the channel dimension from $\hat{C}$ to C'.

Spatial Attention Module. The spatial attention utilizes the local inter-spatial relationship of features and focuses on 'where' is text, which further solve the false positives.

Given the output feature F_{NL} of the non-local self-attention module as input, we perform a 3×3 convolution and then a 1×1 convolution with one channel to generate a text saliency map. A sigmoid function is further applied to obtain the spatial attention map $SAttMap \in \mathbb{R}^{H\times W}$. The attention output O_s is calculated as:

$$O_s = Broadcast(SAttMap) \odot Conv_{3\times3}(F_{NL}), \quad (9)$$

$$SAttMap = Sigmoid(Conv_{1\times1}(Conv_{3\times3}(F_{NL}))), \quad (10)$$

where $SAttMap$ is broadcast to the same C' channel as F_{NL}, $\odot$ represents element-wise multiplication. The output $F_{SA} \in \mathbb{R}^{C'\times H\times W}$ of the spatial attention, also the final output $F_{CA} \in \mathbb{R}^{C'\times H\times W}$ of the context attention, is given by

$$F_{CA} = F_{SA} = F_{NL} + O_s. \quad (11)$$

3.3. Repulsive Text Border

Text border is critical for scene text detection since the border is actually the splitting mark for different text instances. Especially for the very close text instances and the curved texts, which often appear in scene text, more accurate text border is required. Inspired by the work of PixelLink [3], which learns 8-neighbor links for a pixel and uses the links to determine the text border, we also use 8-neighbor link to learn the text border. We introduce two kinds of 8-neighbor links: affinity and repulsive pixel links for each pixel.

As shown in Fig. 3(a), for a given pixel and one of its neighbors, if they lie within the same text instance, the affinity pixel link between them is labeled as positive, and otherwise negative. We only focus on the positive pixels and the loss for affinity pixel links is calculated by:

$$L_{alink} = \frac{L_{alink\_pos}}{sum(alink\_pos)} + \frac{L_{alink\_neg}}{sum(alink\_neg)}, \quad (12)$$

where L_{alink_pos} and L_{alink_neg} are the cross-entropy losses

on the positive and negative affinity links, respectively; $sum(alink\_pos)$ and $sum(alink\_neg)$ are the number of the positive and negative affinity links, respectively.

The affinity pixel links generally pay attention to the link between neighbor pixels that belong to the same text instance. However, the links between pixels located at the text border require more attention. As illustrated in Fig. 3(b), we shrink the annotated text box G with the offset D to G_d and consider the gap between G and G_d as the text border (gray area in Fig. 3(b)). The offset D is computed from the perimeter L and area A of the box G:

$$D = \frac{A(1-r^2)}{L}, \tag{13}$$

where r is the shrink ratio, set to 0.4 empirically. We only focus on the positive pixels in the text border and ignore the other positive pixels. For a pixel in the text border and one of its neighbors, if they lie within different text instances or the neighbor pixel is non-text, the repulsive pixel link between them is labeled as positive, and otherwise negative. Similarly, we also use class-balanced cross-entropy loss as the loss for repulsive pixel links:

$$L_{rlink} = \frac{L_{rlink\_pos}}{sum(W_{rlink\_pos})} + \frac{L_{rlink\_neg}}{sum(W_{rlink\_pos})}, \tag{14}$$

where L_{rlink_pos} and L_{rlink_neg} are the cross-entropy losses on the positive and negative repulsive links, respectively; $sum(W_{rlink_pos})$and $sum(W_{rlink\_neg})$ are the sum of the weighted positive and negative repulsive links, respectively. For the positive repulsive links in which the two neighbor pixels lie in two text instances, they are assigned larger weight (2.0) while for other repulsive links, their weight is set to 1.0.

3.4. Training and Inference

The objective function of learning pixels and links is defined as follows:

$$L_{seg} = \lambda L_{pixel} + L_{alink} + L_{rlink}, \tag{15}$$

where L_{pixel} is the loss on pixel classification task, L_{alink} and L_{rlink} are the link losses. λ is the weight of pixel loss and set to 2.0.

Considering the extreme imbalance of text and non-text pixels, we use online hard example mining (OHEM) to select negative pixels and adopt the weighted cross-entropy loss to supervise pixel classification:

$$L_{pixel} = \frac{1}{(1+r)S} W L_{pixel\_CE}, \tag{16}$$

where L_{pixel_CE} is the cross-entroy loss on text/non-text prediction, r is the negative-positive ratio and is set to 3. S is the total number of the positive pixels. W is pixel weight matrix. For the negative pixels, their weights are set to 1.0, and for each positive pixel i, its weight is calculated as:

$$w_i = \frac{S}{N \cdot S_i}, \tag{17}$$

where N is the number of text instances, S_i is the number of pixels of the text instance that the positive pixel lies in.

Given predictions on pixels, affinity links and repulsive links, three different thresholds are applied to them. The pixel above the pixel threshold is regarded as positive. The link between two neighbor pixels is regarded as positive if the affinity link score is above the affinity link threshold and the repulsive link score is below the repulsive link threshold. Positive pixels are then grouped together using positive links, resulting in a collection of text instances.

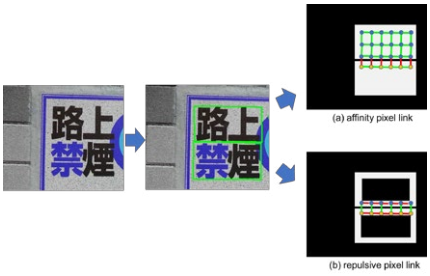


Figure 3. An illustration of affinity and repulsive pixel link. Green lines in (a) and (b) denote positive affinity and repulsive pixel links, respectively; red lines in (a) and (b) denote negative affinity and repulsive pixel links, respectively.

4. Experiments

We evaluate our method on four public datasets: MSRA-TD500 [52], ICDAR2015[15], ICDAR2017-MLT [37] and CTW1500 [29], and compare it with several state-of-art methods.

4.1. Datasets

SynthText [6] is a synthetically generated dataset containing 800 thousand images and about 8 million word instances. It is created by blending natural images with texts of random sizes and fonts. We only use the dataset for pre-training our network.

MSRA-TD500 [52] includes 300 training images and 200 test images collected from natural scenes. It is a dataset with multilingual, arbitrary-oriented and long text lines.

ICDAR2015 [15] is the most commonly used benchmark for detecting scene text in arbitrary directions. It contains 1000 training images and 500 testing images. The images are collected by Google Glass without taking care of positioning, image quality, and viewpoint. Therefore, text in these images is of various scales, orientations, contrast, blurring, and viewpoint, making it challenging for detection. Annotations are provided as word quadrilaterals.

ICDAR2017-MLT [37] is a large-scale multilingual text dataset, which includes 7200 training images, 1800 validation images and 9000 test images. The dataset

consists of scene text images which come from 9 languages. Image annotations are labeled as word-level quadrangles.

CTW1500 [29] is a recent challenging dataset for curve text detection. It has 1000 training images and 500 testing images with over 10 thousand text annotations. Text instances are annotated by 14 vertices of polygons.

4.2. Implementation Details

We pre-train our network on SynthText and then finetune it on the real datasets. The models are optimized by SGD with momentum = 0.9. For training, images are resized to 512*512 after random cropping. Batch size is set to 12 owing to the GPU memory limitation and the learning rate is fixed to 1e-4 and set to 1e-5 for the last several epochs. VGG16 is used as the backbone of our network. Thresholds on pixel and links are crucial for detecting performance. We find the thresholds for each dataset via a grid search with 0.05 step on a hold-out validation set. The whole algorithm is implemented in Tensorflow 1.8.0 and pure Python.

4.3. Ablation Study

To verify the effectiveness of our design, we conduct all experiments of ablation studies on the ICDAR2015 dataset (an oriented text dataset) and CTW1500 dataset (a curved text dataset). The scale of test image for ICDAR2015 and CTW1500 is 1280x768.

Baseline. We implement the method with no context attention and only affinity pixel link as our baseline method.

Context Attention. We implement the model with context attention and only affinity pixel link. Considering that there are three modules in context attention, we implement three models: GA, GA+NL, GA+NL+SA. From Tab. 1, the GA achieves 2.2% improvement on ICDAR2015 and 1.5% improvement on CTW1500 than baseline; the GA+NL achieves 0.5% improvement on ICDAR2015 and 1.4% improvement on CTW1500 than GA; the GA+NL+SA achieves 0.9% improvement on ICDAR2015 and 1.1% improvement on CTW1500 than GA+NL. The results demonstrate that the attention modules used in context attention are all useful. Overall, the model with context attention makes 3.6% improvement on ICDAR2015 and 4.0% improvement on CTW1500.

The effectiveness of repulsive link. To investigate the effectiveness of repulsive link, we implement the model (GA+NL+SA+RL) with context attention and the affinity and repulsive link. From Tab. 1, the model with repulsive link achieves 0.4% improvement on ICDAR2015 and 0.7% improvement on CTW1500, in comparison to the model without repulsive link (GA+NL+SA).

4.4. Results on Scene Text Benchmarks

Long straight text detection. We evaluate the performance of our method on MSRA-TD500, which contains multi-lingual, arbitrary-oriented and long text lines. Images are resized to 768x768 for testing. Thresholds of text pixel, affinity pixel link and repulsive pixel link are set to (0.9, 0.85, 0.8). As shown in Tab. 2, our method achieves F-measure of 86.1%, which is better than all the other methods. The results also demonstrate the advantages of our method for dealing with long text lines. Some of the detection results are visualized in Fig. 4(a).

Method	ICDAR2015			CTW1500		
	P	R	F	P	R	F
Baseline	85.1	82.0	83.5	81.1	73.9	77.3
GA	87.5	83.9	85.7	82.8	75.1	78.8
GA+NL	88.0	84.5	86.2	83.9	76.8	80.2
GA+NL+SA	89.7	84.6	87.1	85.3	77.7	81.3
GA+NL+SA+RL	90.0	85.1	87.5	85.8	78.6	82.0

Table 1. Ablation experiments of validating the effectiveness of different modules on ICDAR2015 and CTW1500 dataset. “GA” means global attention, “GA+NL” means global attention + non-local self-attention, “GA+NL+SA” means context attention, “GA+NL+SA+RL” means context attention + repulsive link.

Oriented text detection. We evaluate our method on the ICDAR 2015 to test its ability of detecting oriented text. Thresholds of text pixel, affinity pixel link and repulsive link are set to (0.85,0.85,0.8). We use a single scale of 1280x768 for test images and achieve 90.0, 85.1 and 87.5 in precision, recall and F-measure, respectively. As shown in Tab. 3, except for the end-to-end method FOTS which combines text detection and recognition, our method outperforms the state-of-art methods. Also note that the very high precision (90.0%) is obtained, which verifies that our method can suppress false positives effectively. Some of the detection results are visualized in Fig. 4(b).

Multilingual text detection. To verify the generalization ability of our method on multilingual scene text detection, we evaluate our method on ICDAR2017-MLT. We use a single scale of 1536x1536 for test images. The 7200 training images are used for training and the 1800 validation images are used for selecting the models and thresholds. Thresholds of text pixel, affinity link and repulsive link are set to (0.9,0.45,0.8). We achieve an F-measure of 75.3%, which is comparable to the best reported result in literature. Some of the detection results are visualized in Fig. 4(c).

Curved text detection. We evaluate the ability of our model to detect curved text on CTW1500 dataset. Our method can be flexibly applied to curved text without special modifications. The only modification lies in the interface of reading text polygons with 14 vertices. We use a single scale of 1280x768 for test images. Thresholds of text pixel, affinity link and repulsive link are set to

(0.75,0.8,0.8). As shown in Tab. 5, our method achieves the state-of-the-art results and outperforms some existing methods such as TextSnake [30] and LOMO [55]. Some of the detection results are visualized in Fig. 4(d).

Method	Precision	Recall	F-measure
RRPN [34]	82.0	68.0	74.0
SegLink [43]	86.0	70.0	77.0
PixelLink [3]	83.0	73.2	77.8
Lyu et al. [33]	87.6	76.2	81.5
MCN [27]	88.0	79.0	83.0
PAN [48]	84.4	83.8	84.1
OURS	88.8	83.5	86.1

Table 2. Quantitative results of different methods on MSRA-TD500 (**long straight text**) dataset. Our method achieves the best performance over all the other methods, showing the advantages of dealing with long text lines.

Method	Precision	Recall	F-measure
SegLink[43]	73.1	76.8	75.0
RRPN[34]	84.0	77.0	80.0
EAST[59]	83.3	78.3	80.7
TextBoxes++ [20]	87.2	76.7	81.7
TextSnake [30]	84.9	80.4	82.6
PixelLink [3]	85.5	82.0	83.7
PSENet-1s [18]	86.9	84.5	85.7
Mask Textspotter [32]	91.6	81.0	86.0
LOMO [55]	91.3	83.5	87.2
SPCNet [50]	88.7	85.8	87.2
FOTS [26]	-	-	88.0
OURS	90.0	85.1	87.5

Table 3. Quantitative results of different methods on ICDAR 2015 (**oriented text**) dataset. Except for the end-to-end method FOTS, our method outperforms all the other methods.

Method	Precision	Recall	F-measure
E2E-MLT [38]	64.6	53.8	58.7
He et al. [12]	76.7	57.9	66.0
Lyu et al. [33]	83.8	56.6	66.8
FOTS [26]	81.0	57.5	67.3
Border [51]	77.7	62.1	69.0
AF-RPN [58]	75.0	66.0	70.0
PSENet-1s [18]	77.0	68.4	72.5
LOMO MS [55]	80.2	67.2	73.1
SPCNet [50]	80.6	68.6	74.1
OURS	83.7	68.4	75.3

Table 4. Quantitative results of different methods on ICDAR2017-MLT (**multilingual text**) dataset. MS means multi-scale testing.

Method	Precision	Recall	F-measure
SegLink [43]	42.3	40.0	40.8
EAST [59]	78.7	49.1	60.4
CTD [29]	74.3	65.2	69.5
CTD+TLOC [29]	77.4	69.8	73.4
TextSnake [30]	67.9	85.3	75.6
LOMO MS [55]	85.7	76.5	80.8
PSENet-1s [18]	84.8	79.7	82.2
OURS	85.8	78.6	82.0

Table 5. Quantitative results of different methods on CTW1500 (**curved text**) dataset.

5. Conclusion and Future Work

In this paper, we propose an accurate segmentation-based scene text detector with context attention and repulsive text border. We design an effective attention mechanism to better exploit the context information by sequentially applying global attention, non-local self-attention and spatial attention. The context is helpful for reducing local ambiguities for pixel classification, which can greatly reduce false positives and the misdetections of

(a) (b) (c) (d)

Figure 4. Examples of detection results. From left to right: (a) MSRA-TD500, long straight text, (b) ICDAR2015, oriented text, (c) ICDAR2017-MLT, multilingual text, (d) CTW1500, curved text.

long text lines. To further solve the very close text instance, we propose to learn an extra repulsive pixel link that explicitly represents the relationship between pixels located at text border. The robustness and effectiveness of our approach are verified on several public benchmarks including long, curved, oriented and multilingual text cases. In the future, we would like to further focus on the text border and develop a two-stream segmentation network to simultaneously learn text pixels and text boundaries.

References

[1] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven. Photoocr: Reading text in uncontrolled conditions. In ICCV, 2013.
[2] J. L. Cao, Y. W. Pang, and X. L. Li. Triply Supervised Decoder Networks for Joint Detection and Segmentation. In CVPR, 2019.
[3] D. Deng, H. Liu, X. Li, and D. Cai. Pixellink: Detecting scene text via instance segmentation. In AAAI, 2018.
[4] M. En, Rong Li, J. Li, B. Liu. Feature Pyramid Based Scene Text Detector. In ICDAR, 2017.
[5] R. Girshick. Fast R-CNN. In ICCV, 2015.
[6] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In CVPR, 2016.
[7] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. In ICCV, 2017.
[8] T. He, W. Huang, Y. Qiao and J. Yao. Accurate text localization in natural image with cascaded convolutional text network. arXiv, 2016.
[9] K. He, X. Zhang, S. Ren, J. Sun. Deep residual learning for image recognition. In CVPR, 2016.
[10] K. He, X. Zhang, S. Ren, J. Sun. Identity mappings in deep residual networks. In ECCV, 2016.
[11] W. He, X. Zhang, F. Yin, and C. Liu. Deep direct regression for multi-oriented scene text detection. In ICCV, 2017.
[12] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu. Multi-oriented and multi-lingual scene text detection with direct regression. IEEE Transactions on Image Processing, 27(11):5406–5419, 2018.
[13] G. Huang, Z. Liu, K.Q. Weinberger, L. van der Maaten. Densely connected convolutional networks. In CVPR, 2017.
[14] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. International Journal of Computer Vision, 2016, 116(1): 1-20.
[15] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. K. Ghosh, A. D. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny. ICDAR 2015 competition on robust reading. In ICDAR, 2015.
[16] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. Almazan, and L. de las Heras. ICDAR 2013 robust reading competition. In ICDAR, 2013.
[17] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, J. Sun. DetNet: A backbone network for object detection. arXiv:1804.06215, (2018).
[18] X. Li,W. H. Wang, W. B. Hou, R. Z. Liu, T. Lu, and J. Yang. Shape robust text detection with progressive scale expansion network. In CVPR, 2019.
[19] H. Li, P. Xiong, J. An, and L. Wang. Pyramid attention network for semantic segmentation. In BMVC, 2018.
[20] M. Liao, B. Shi, and X. Bai. Textboxes++: A single-shot oriented scene text detector. IEEE Transactions on Image Processing, vol. 27, no. 8, 2018.

[21] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie. Feature Pyramid Networks for Object Detection. arXiv preprint. arXiv: 1612.03144, 2017.
[22] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu. Textboxes: A fast text detector with a single deep neural network. In AAAI, 2017.
[23] M. H. Liao, Z. Zhu, B. G. Shi, G. S. Xia, X. Bai. Rotation-sensitive Regression for Oriented Scene Text Detection. In CVPR, 2018.
[24] C. Lin, J. Lu, G. Wang, and J. Zhou. Graininess-aware deep feature learning for pedestrian detection. In ECCV, 2018.
[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In ECCV, 2016.
[26] X.Liu, D.Liang, S.Yan, D.Chen, Y.Qiao, and J.Yan. Fots: Fast oriented text spotting with a unified network. In CVPR, 2018.
[27] Z. C. Liu, G. S. Lin, S. Yang, J. S. Feng, W. S. L, W. L. Goh. Learning Markov Clustering Networks for Scene Text Detection. In CVPR, 2018.
[28] Y. Liu and L. Jin. Deep matching prior network: Toward tighter multi-oriented text detection. In CVPR, 2017.
[29] Y. L. Liu, L. W. Jin, S. T. Zhang, and S. Zhang. Detecting curve text in the wild: New dataset and new solution. arXiv preprint arXiv:1712.02170, 2017.
[30] S. B. Long, J. Q. Ruan, W. J. Zhang, X. He, W. H. Wu, C. Yao. TextSnake: A Flexible Representation for Detecting Text of Arbitrary Shapes. In ECCV, 2018.
[31] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015.
[32] P. Y. Lyu, M. H. Liao, C. Yao, W. H. Wu, X. Bai. Mask TextSpotter: An End-to-End Trainable Neural Network for Spotting Text with Arbitrary Shapes. In ECCV, 2018.
[33] P. Y. Lyu, C. Yao, W. H. Wu, X. Bai. Multi-oriented scene text detection via corner localization and region segmentation. In CVPR, 2018.
[34] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue. Arbitrary-oriented scene text detection via rotation proposals. IEEE Transactions on Multimedia, 20(11):3111–3122, 2018.
[35] J. Mao, T. Xiao, Y. Jiang, and Z. Cao. What can help pedestrian detection? In CVPR, 2017.
[36] S. Mohanty, T. Dutta, and H. P. Gupta. Robust Scene Text Detection with Deep Feature Pyramid Network and CNN based NMS Model. In ICPR, 2018.
[37] N. Nayef, F. Yin, I. Bizid, H. Choi, Y. Feng, D. Karatzas, Z. Luo, U. Pal, C. Rigaud, J. Chazalon, et al. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In ICDAR, 2017.
[38] Y. Patel, M. Busta, and J. Matas. E2e-mlt-an unconstrained end-to-end method for multi-language scene text. arXiv preprint arXiv:1801.09919, 2018.
[39] V.-Q. Pham, S. Ito, and T. Kozakaya. Biseg: Simultaneous instance segmentation and semantic segmentation with fully convolutional networks. In BMVC, 2017.
[40] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollar. Learning to refine object segments. In ECCV, 2016.
[41] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In CVPR, 2016.
[42] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In NIPS, 2015.
[43] B. Shi, X. Bai, and S. Belongie. Detecting Oriented Text in Natural Images by Linking Segments. In CVPR, 2017.
[44] K. Simonyan, K., Zisserman, A. Vedaldi. Very deep convolutional networks for large-scale image recognition. arXiv, 2014.
[45] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S.Reed, D. Anguelov, D. Erhan, et al. Going deeper with convolutions. In CVPR, 2015.
[46] J. Tang, Z. B. Yang, Y. P. Wang, Q. Zheng, Y. C. Xu, X. Bai. Detecting Dense and Arbitrary-shaped Scene Text by Instance-aware Component Grouping. Pattern Recognition, 2019.
[47] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In ECCV, 2016.
[48] W. h. Wang, E. Xie, X. G. Song, Y. H. Zang, W. J. Wang, T. Lu, G. Yu, and C. H. Shen. Efficient and Accurate Arbitrary-Shaped Text Detection with Pixel Aggregation Network. In ICCV, 2019.
[49] S. Woo, J. Park, J. Lee. CBAM: Convolutional Block Attention Module. In ECCV, 2018.
[50] E. Xie, Y. H. Zang, S. Shao, G. Yu, C. Yao, and G. Y. Li. Scene text detection with supervised pyramid context network. In AAAI, 2019.
[51] C. Xue, S. Lu, and F. Zhan. Accurate scene text detection through border semantics awareness and bootstrapping. In ECCV, 2018.
[52] C. Yao, X. Bai, W. Y. Liu, Y. Ma, and Z. W. Tu. Detecting texts of arbitrary orientations in natural images. In CVPR, 2012.
[53] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou and Z. Cao. Scene text detection via holistic, multi-channel prediction. arXiv preprint arXiv:1606.09002, 2016.
[54] Q. Ye and D. Doermann. Text detection and recognition in imagery: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 7, pp. 1480–1500, 2015.
[55] C. Zhang, B. Liang, Z. Huang, M. En, and et al. Look More Than Once: An Accurate Detector for Text of Arbitrary Shapes. In CVPR, 2019.
[56] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu and X. Bai. Multi-oriented text detection with fully convolutional networks. In CVPR, 2016.
[57] Z. Zhong, S. Huang. Deeptext: A new approach for text proposal generation and text detection in natural images. In ICASSP, 2017.
[58] Z. Zhong, L. Sun, and Q. Huo. An anchor-free region proposal network for faster r-cnn based text detection approaches. arXiv preprint arXiv:1804.09003, 2018.
[59] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang. East: An efficient and accurate scene text detector. In CVPR, 2017.
[60] Z. Zhu, M. D. Xu, S. Bai, T. T. Huang, and X. Bai. Asymmetric non-local neural networks for semantic segmentation. In ICCV, 2019.
[61] Y. Wu and P. Natarajan. Self-organized text detection with minimal post-processing via border learning. In ICCV, 2017.

CenterMask: Single Shot Instance Segmentation With Point Representation

Yuqing Wang Zhaoliang Xu Hao Shen Baoshan Cheng Lirong Yang
Meituan Dianping Group
{wangyuqing06,xuzhaoliang,shenhao04,chengbaoshan02,yanglirong}@meituan.com

Abstract

In this paper, we propose a single-shot instance segmentation method, which is simple, fast and accurate. There are two main challenges for one-stage instance segmentation: object instances differentiation and pixel-wise feature alignment. Accordingly, we decompose the instance segmentation into two parallel subtasks: Local Shape prediction that separates instances even in overlapping conditions, and Global Saliency generation that segments the whole image in a pixel-to-pixel manner. The outputs of the two branches are assembled to form the final instance masks. To realize that, the local shape information is adopted from the representation of object center points. Totally trained from scratch and without any bells and whistles, the proposed CenterMask achieves 34.5 mask AP with a speed of 12.3 fps, using a single-model with single-scale training/testing on the challenging COCO dataset. The accuracy is higher than all other one-stage instance segmentation methods except the 5 times slower TensorMask, which shows the effectiveness of CenterMask. Besides, our method can be easily embedded to other one-stage object detectors such as FCOS and performs well, showing the generation of CenterMask.

1. Introduction

Instance segmentation [11] is a fundamental and challenging computer vision task, which requires to locate, classify, and segment each instance in the image. Therefore, it has both the characters of object detection and semantic segmentation. State-of-the-art instance segmentation methods [12, 21, 14] are mostly built on the advances of two-stage object detectors [9, 8, 26]. Despite the popular trend of one-stage object detection [13, 25, 22, 17, 27, 30], only a few works[1, 2, 28, 7] are focusing on one-stage instance segmentation. *In this work, we aim to design a simple one-stage and anchor-box free instance segmentation model.*

Instance segmentation is much harder than object detection because the shapes of instances are more flexible than the two-dimensional bounding boxes. There are two main challenges for one-stage instance segmentation: (1) how to differentiate object instances, especially when they are in the same category. Some methods [3, 1] extract the global features of the image firstly then post-process them to separate different instances, but these methods struggle when objects overlap. (2) how to preserve pixel-wise location information. State-of-the-art methods represent masks as structured 4D tensors [2] or contour of fixed points [28], but still facing the pixel misalignment problem, which makes the masks coarse at the boundary. TensorMask [2] designs complex pixel align operations to fix this problem, which makes the network even slower than the two-stage counterparts.

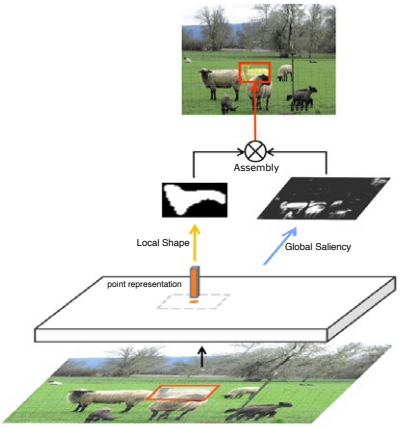


Figure 1: **Illustration of CenterMask.** The Local Shape branch separates objects locally and the Global Saliency Map realizes pixel-wise segmentation of the whole image. Then the coarse but instance-aware local shape and the precise but instance-unaware global saliency map are assembled to form the final instance mask.

To address these issues, we propose to break up the mask

Figure 2: **Results of CenterMask on COCO test set images**. These results are based on Hourglass-104 backbone, achieving a mask AP of 34.5 and running at 12.3 fps. Our method differentiates objects well in overlapping conditions with precise masks.

representation into two parallel components: (1) a Local Shape representation that predicts a coarse mask for each local area, which can separate different instances automatically. (2) a Global Saliency Map that segment the whole image, which can provide saliency details, and realize pixel-wise alignment. To realize that, the local shape information is extracted from the point representation at object centers. Modeling object as its center point is motivated by the one-stage CenterNet [30] detector, thus we call our method CenterMask.

The illustration of the proposed CenterMask is shown in Figure 1. Given an input image, the object center point locations are predicted following a keypoint estimation pipeline. Then the feature representation at the center point is extracted to form the local shape, which is represented by a coarse mask that separates the object from close ones. In the meantime, the fully convolutional backbone produces a global saliency map of the whole image, which separates the foreground from the background at pixel level. Finally, the coarse but instance-aware local shapes and the precise but instance-unaware global saliency map are assembled to form the final instance masks.

To demonstrate the robustness of CenterMask and analyze the effects of its core factors, extensive ablation experiments are conducted and the performance of multiple basic instantiations are compared. Visualization shows that the CenterMask with only Local Shape branch can separate objects well, and the model with only Global Seliency branch performs good enough in objects-non-overlapping situations. In complex and objects-overlapping situations, combination of these two branches differentiates instances and realizes pixel-wise segmentation simultaneously. Results of CenterMask on COCO [20] test set images are shown in Figure 2.

In summary, the main contributions of this paper are as follows:

- An anchor-box free and one-stage instance segmentation method is proposed, which is simple, fast and accurate. Totally trained from scratch and without any bells and whistles, the proposed CenterMask achieves 34.5 mask AP with a speed of 12.3 fps on the challenging COCO, showing the good speed-accuracy trade-off. Besides, the method can be easily embedded to other one-stage object detectors such as FCOS[27] and performs well, showing the generation of CenterMask.

- The Local Shape representation of object masks is proposed to differentiate instances in the anchor-box free condition. Using the representation of object center points, the Local Shape branch predicts coarse masks and separate objects effectively even in the overlapping situations.

- The Global Saliency Map is proposed to realize pixel-wise feature alignment naturally. Different from previous feature align operations for instance segmentation, this module is simpler, faster, and more precise. The Global Saliency generation acts similar to semantic segmentation [23], and hope this work can motivate one-stage panoptic segmentation [16] in the future.

2. Related Work

Two-stage Instance Segmentation: Two-stage instance segmentation method often follows the *detect-then-segment* paradigm, which first performs bounding box detection and then classifies the pixels in the bounding box area to obtain the final mask. Mask R-CNN [12] extends the successful Faster R-CNN [26] detector by adding a mask segmentation branch on each Region of Interest area. To preserve the exact spatial locations, it introduces the RoIAlign module to fix the pixel misalignment problem. PANet [21] aims to improve the information propagation of Mask R-CNN by introducing bottom-up path augmentation, adaptive feature pooling, and fully-connected fusion. Mask Scoring R-CNN [14] proposes a mask scoring module instead of the classi-

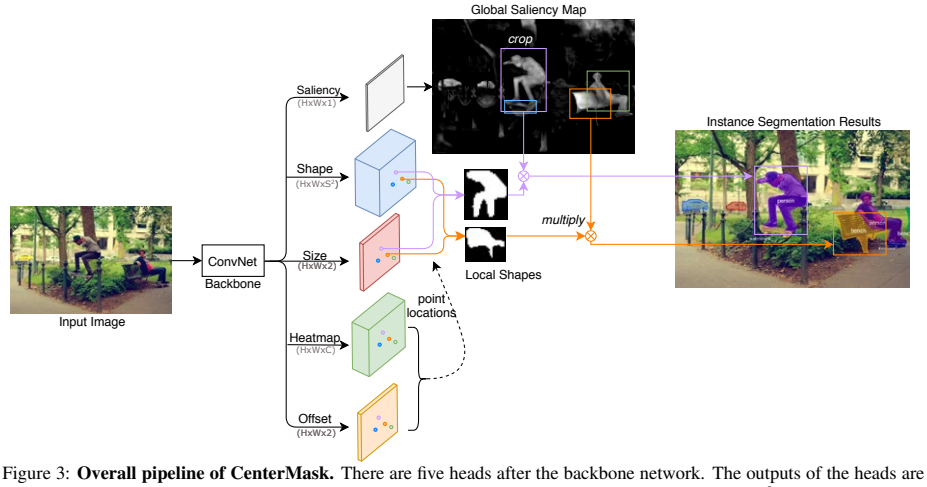


Figure 3: **Overall pipeline of CenterMask.** There are five heads after the backbone network. The outputs of the heads are with the same height (H) and width (W) but different channels. C is the number of categories, and S^2 is the size of shape vector. The Heatmap and Offset heads predict the center point locations. The Shape and Size heads predict the Local Shapes at the corresponding locations. The Saliency head predicts a Global Saliency Map. The Local Shape and cropped Saliency Map are multiplied to form the final mask for each instance. For visualization convenience, the whole segmentation pipeline for only two instances is shown in the figure, and the Global Saliency Map is visualized in the class-agnostic form.

fication score to evaluate the mask, which can improve the quality of the segmented mask.

Although two-stage instance segmentation methods achieve state-of-the-art performance, these models are often complicated and slow. Advances of one-stage object detection motivate us to develop faster and simpler one-stage instance segmentation methods.

One-stage Instance Segmentation: State-of-the-art one-stage instance segmentation methods can be roughly divided into two categories: *global-area-based* and *local-area-based* approaches. *Global-area-based* methods first generate intermediate and shared feature maps based on the whole image, then assemble the extracted features to form the final masks for each instance.

InstanceFCN [3] utilizes FCN [23] to generate multiple instance-sensitive score maps which contain the relative positions to objects instances, then applies an assembling module to output object instances. YOLACT [1] generates multiple prototype masks of the global image, then utilizes per-instance mask coefficients to produce the instance level mask. *Global-area-based* methods can maintain the pixel-to-pixel alignment which makes masks precise, but performs worse when objects overlap. In contrast to these methods, *local-area-based* methods output instance masks on each local region directly. PolarMask [28] represents mask in its contour form and utilizes rays from the center to describe the contour, but the polygon surrounded by the contour can not depict the mask precisely and can not describe objects that have holes in the center. TensorMask [2] utilizes structured 4D tensors to represent masks over a spatial domain, it also introduces aligned representation and tensor bipyramid to recover spatial details, but these align operations make the network even slower than the two-stage Mask R-CNN [12].

Different from the above approaches, CenterMask contains both a Global Saliency generation branch and a Local Shape prediction branch, and integrates them to preserve pixel alignment and separate objects simultaneously.

3. CenterMask

The goal of this paper is to build a one-stage instance segmentation method. One-stage means that there is no pre-defined Region-of-Interests (RoIs) for mask prediction, which requires to locate, classify, and segment objects simultaneously. To realize that, we break the instance segmentation into two simple and parallel sub-tasks, and assemble the results of them to form the final masks. The first branch predicts coarse shape from the center point representation of each object, which can constrain the local area for each object and differentiate instances naturally.

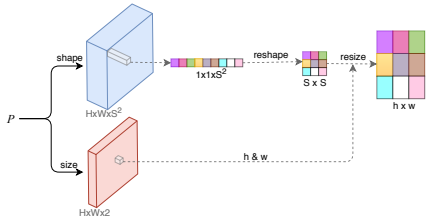


Figure 4: **Architecture of the shape head and size head for Local Shape prediction.** P represents the feature maps extracted by the backbone network. H and W represents the height and width of the head outputs. The channel size of the shape head is SxS, and the channels of the size head is 2, with h and w being the predicted height and width for the object at the point.

The second branch predicts a saliency map of the whole image, which realizes precise segmentation and preserves exact spatial locations. In the end, the mask for each instance is constructed by multiplying the outputs of the two branches.

3.1. Local Shape Prediction

To differentiate instances at different locations, we choose to model the masks from their center points. The center point is defined as the center of the surrounding bounding box for each object. A natural thought is to represent it by the extracted image feature at the center point location, but a fixed-size image feature can not represent masks in various sizes. To address this issue, we decompose the object mask into two components: the mask size and the mask shape. The size for each mask can be represented by the object height and width, and the shape can be described by a 2D binary array of fixed size.

The above two components can be predicted in parallel using fixed-size representations of the center points. The architecture of the two heads is shown in Figure 4. P represents the image features extracted by the backbone network. Let $F_{shape} \in \mathbb{R}^{H \times W \times S^2}$ be the output of the Local Shape head, with H and W represent the height and width of the whole map, and S^2 represents the number of output channels for this head. The output of the Size head $F_{size} \in \mathbb{R}^{H \times W \times 2}$ is in the same height and width, with a channel size of two.

For a center point (x, y) at the feature map, the shape feature at this location is extracted by $F_{shape}(x, y)$. The shape vector is in the size of $1 \times 1 \times S^2$, and then be reshaped to a 2D shape array of size $S \times S$. The size prediction of the center point is $F_{size}(x, y)$, with the height and width being h and w. The above 2D shape array is then resized to the size of $h \times w$ to form the final local shape prediction.

For convenience, the Local Shape Prediction branch is used to refer to the combination of the shape and size heads. This branch produces masks from local point representation, and predicts a local area for each object, which makes it suitable for instance differentiation.

3.2. Global Saliency Generation

Although the Local Shape branch generates a mask for each instance, it is not enough for precise segmentation. As the fixed-size shape vector can only predict a coarse mask, resizing and warping it to the object size losses spatial details, which is a common problem for instance segmentation. Instead of relying on complex pixel calibration mechanism [12, 2], we design a simpler and faster approach.

Motivated by semantic segmentation [23] which makes pixel-wise predictions on the whole image, we propose to predict a Global Saliency Map to realize pixel level feature alignment. The Map aims to represent the salience of each pixel in the whole image, i.e., whether the pixel belonging to an object area or not.

Utilizing the fully convolutional backbone, the Global Saliency branch performs the segmentation on the whole image in parallel with the existing Local Shape branch. Different from semantic segmentation methods which utilize *softmax* function to realize pixel-wise competition among object classes, our approach uses *sigmoid* function to perform binary classification. The Global Saliency Map can be class-agnostic or class-specific. In the class-agnostic setting, only one binary map is produced to indicate whether the pixels belonging to the foreground or not. For the class-specific setting, the head produces a binary mask for each object category.

An example of Global Saliency Map is shown in the top of Figure 3, using the class-agnostic setting for visualization convenience. As can be seen in the figure, the map highlights the pixels that have saliency, and achieves pixel-wise alignment with the input image.

3.3. Mask Assembly

In the end, the Local Shapes and Global Saliency Map are assembled together to form the final instance masks. The Local Shape predicts the coarse area for each instance, and the cropped Saliency Map realizes precise segmentation in the coarse area. Let $L_k \in \mathbb{R}^{h \times w}$ represent the Local Shape for one object, and $G_k \in \mathbb{R}^{h \times w}$ be the corresponding cropped Saliency Map. They are in the same size of the predicted height and width.

To construct the final mask, we firstly transform their values to the range of (0,1) using the $sigmoid$ function, then calculate the Hadamard product of the two matrices:

$$M_k = \sigma(L_k) \odot \sigma(G_k) \tag{1}$$

There is no separate loss for the Local Shape and Global

Saliency branch, instead, all supervision comes from the loss function of the assembled mask. Let T_k denote the corresponding ground truth mask, the loss function of the final masks is :

$$L_{mask} = \frac{1}{N}\sum_{k=1}^{N} Bce(M_k, T_k) \tag{2}$$

where Bce represents the pixel-wise binary cross entropy, and N is the number of objects.

3.4. Overall pipeline of CenterMask

The overall architecture of CenterMask is shown in Figure 3. The Heatmap head is utilized to predict the positions and categories for center points, following a typical keypoint estimation[24] pipeline. Each channel of the output is a heatmap for the corresponding category. Obtaining the center points requires to search the peaks for each heatmap, which are defined as the local maximums within a window. The Offset head is utilized to recover the discretization error caused by the output stride.

Given the predicted center points, the Local Shapes for these points are calculated by the outputs of the Shape head and the Size head at the corresponding locations, following the approach in Section 3.1. The Saliency head produces the Global Saliency Map. In the class-agnostic setting, the output channel number is 1, the Saliency map for each instance is obtained by cropping it with the predicted location and size. In the class-specific setting, the channel of the corresponding predicted category is cropped. The final masks are constructed by assembling the Local Shapes and the Saliency Map.

Loss function: The overall loss function is composed of four losses: the center point loss, the offset loss, the size loss, and the mask loss. The center point loss is defined in the same way as the Hourglass network [24], let $\hat{Y}_{ijc}$ be the score at the location (i,j) for class c in the predicted heatmaps, and Y be the "ground-truth" heatmap. The loss function is a pixel-wise logistic regression modified by the focal loss [19]:

$$L_p = \frac{-1}{N}\sum_{ijc}\begin{cases}(1-\hat{Y}_{ijc})^{\alpha}\log(\hat{Y}_{ijc}) & \text{if } Y_{ijc}=1\\ (1-Y_{ijc})^{\beta}(\hat{Y}_{ijc})^{\alpha}\log(1-\hat{Y}_{ijc}) & \text{otherwise}\end{cases} \tag{3}$$

where N is the number of center points in the image, α and β are the hyper-parameters of the focal loss; The offset loss and size loss follow the same setting of CenterNet [30], which utilize L1 loss to penalize the distance. Let $\hat{O}$ represent the predicted offset, p represent the ground truth center point, and R represents the output stride, then the low-resolution equivalent of p is $\tilde{p} = \lfloor \frac{p}{R} \rfloor$, therefore the offset loss is:

$$L_{off} = \frac{1}{N}\sum_{p}\left|\hat{O}_{\tilde{p}} - (\frac{p}{R} - \tilde{p})\right| \tag{4}$$

Let the true object size be $S_k = (h, w)$, the predicted size be $\hat{S}_k = (\hat{h}, \hat{w})$, then the size loss is:

$$L_{size} = \frac{1}{N}\sum_{k=1}^{N}\left|\hat{S}_k - S_k\right| \tag{5}$$

The overall training objective is the combination of the four losses:

$$L_{seg} = \lambda_p L_p + \lambda_{off} L_{off} + \lambda_{size} L_{size} + \lambda_{mask} L_{mask} \tag{6}$$

where the mask loss is defined in Equation 2, λ_p, λ_{off}, λ_{size} and λ_{mask} are the coefficients of the four losses respectively.

3.5. Implementation Details

Train: Two backbone networks are involved to evaluate the performance of CenterMask: Hourglass-104 [24] and DLA-34 [29]. S equals 32 for the shape vector. λ_p, λ_{off} and λ_{size}, λ_{mask} are set to 1,1,0.1,1 for the loss function. The input resolution is fixed with 512×512. All models are trained from scratch, using Adam [15] to optimize the overall objects. The models are trained for 130 epochs, with an initial learning rate of 2.5e-4 and dropped $10\times$ at 100 and 120 epoch. As our approach directly makes use of the same hyper-parameters of CenterNet [30], we argue that the performance of CenterMask can be improved further if the hyper-parameters are optimized for it correspondingly.

Inference: During testing, no data augmentation and no NMS is utilized, only returning the top-100 scoring points with the corresponding masks. The binary threshold for the mask is 0.4.

4. Experiments

The performance of the proposed CenterMask is evaluated on the MS COCO instance segmentation benchmark [20]. The model is trained on the 115k `trainval35k` images and tested on the 5k `minival` images. Final results are evaluated on 20k `test-dev`.

4.1. Ablation Study

A number of ablation experiments are performed to analyze CenterMask. Results are shown in Table 1.

Shape size Selection: Firstly, the sensitivity of our approach to the size of the Local Shape representation is analyzed in Table 1a. Larger shape size brings more gains, but the difference is not large, indicating that the Local Shape representation is robust to the feature size. When S equals 32, the performance saturates, therefore we use the number as the default Shape size.

Backbone Architecture: Results of CenterMask with different backbones are shown in Table 1b. The large Hourglass brings about 1.4 gains compared with the smaller

S	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
24	32.0	52.8	33.8	14.0	36.3	48.5
32	**32.5**	**53.6**	33.9	**14.3**	36.3	48.7
48	**32.5**	53.4	**34.1**	13.8	**36.6**	**49.0**

(a) **Size of Shape**: Larger shape size brings more gains. Performance saturates when S equals 32. Results are based on DLA-34.

Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	FPS
DLA-34	32.5	53.6	33.9	14.3	36.3	48.7	**25.2**
Hourglass-104	**33.9**	**55.6**	**35.5**	**16.1**	**37.8**	**49.2**	12.3

(b) **Backbone Architecture**: FPS represents frame-per-second. The Hourglass-104 backbone brings 1.4 gains compared with DLA-34, but its speed is more than 2 times slower.

Shape	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
w/o	21.7	44.7	18.3	9.8	24.0	31.8
w	**31.5**	**53.7**	**32.4**	**15.1**	**35.5**	**45.5**

(c) **Local Shape branch**: Comparison of CenterMask with or without Local Shape branch.

Saliency	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
w/o	26.5	51.8	24.5	12.7	29.8	38.2
w	**31.5**	**53.7**	**32.4**	**15.1**	**35.5**	**45.5**

(d) **Global Saliency branch**: Comparison of CenterMask with or without Global Saliency branch.

	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Class-Agnostic	31.5	53.7	32.4	15.1	35.5	45.5
Class-Specific	**33.9**	**55.6**	**35.5**	**16.1**	**37.8**	**49.2**

(e) **Class-Agnostic *vs*. Class-Specific**: Comparison of the class-agnostic and class-specific setting of Global Saliency branch.

	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
w/o	33.9	55.6	35.5	16.1	37.8	49.2
w	**34.4**	**55.8**	**36.2**	**16.1**	**38.3**	**50.2**

(f) **Direct Saliency supervision**: Comparison of CenterMask with or without direct Saliency supervision.

Table 1: Ablation experiments of CenterMask. All models are trained on `trainval35k` and tested on `minival`, using the Hourglass-104 backbone unless otherwise noted.

(a) **Results of CenterMask in Shape-only setting.** The Local Shape branch seperates instances with coarse masks.

(b) **Results of CenterMask in Saliency-only setting.** The Global Saliency branch performs well when there are no overlap between objects.

(c) **Comparison of CenterMask results in challenging conditions**. Images form left to right are generated by: Shape-only, Saliency-only and the combination of the two branches.

Figure 5: **Images generated by CenterMask in different settings.** The Saliency branch is in class-agnostic setting for this experiment.

Method	Backbone	Resolution	FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
two-stage									
MNC [4]	ResNet-101-C4	-	2.78	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [18]	ResNet-101-C5-dilated	multi-scale	4.17	29.2	49.5	-	7.1	31.3	50.0
Mask R-CNN [12]	ResNeXt-101-FPN	800×1333	8.3	37.1	60.0	39.4	16.9	39.9	53.5
one-stage									
ExtremeNet [31]	Hourglass-104	512×512	3.1	18.9	44.5	13.7	10.4	20.4	28.3
TensorMask [2]	ResNet-101-FPN	800×1333	2.63	37.3	59.5	39.5	17.5	39.3	51.6
YOLACT [1]	ResNet-101-FPN	700×700	23.6	31.2	50.6	32.8	12.1	33.3	47.1
YOLACT-550 [1]	ResNet-101-FPN	550×550	33.5	29.8	48.5	31.2	9.9	31.3	47.7
PolarMask [28]	ResNeXt-101-FPN	768×1280	10.9	32.9	55.4	33.8	15.5	35.1	46.3
CenterMask	DLA-34	512×512	25.2	33.1	53.8	34.9	13.4	35.7	48.8
CenterMask	Hourglass-104	512×512	12.3	34.5	56.1	36.3	16.3	37.4	48.4

Table 2: **Instance segmentation mask AP on COCO `test-dev`.** Resolution represents the image size of training. We show single scale testing for most models. Frame-per-second (FPS) were measured on the same machine whenever possible. A dash indicates the data is not available.

DLA-34 [29]. The model with DLA-34 [29] backbone realizes 32.5 mAP with 25.2 FPS, achieving a good speed-accuracy trade-off.

Local Shape branch: The comparison of CenterMask with or without Local Shape branch is shown in Table 1c, with Saliency branch in class-agnostic setting. The Shape branch brings about 10 gains. Moreover, CenterMask with only the Shape branch achieves 26.5 AP (as shown in the first row of Table 1d), images generated by this model are shown in Figure 5a. Each image contains multiple objects with dense overlaps, the Shape branch can separate them well with coarse masks. The above results illustrate the effectiveness of the proposed Local Shape branch.

Global Saliency branch: The comparison of CenterMask with or without Global Saliency branch is shown in Table 1d, introduction of the Saliency branch improves 5 points, compared with model with only Local Shape branch.

We also conduct visualization to CenterMask with only Saliency branch. As shown in Figure 5b, there is no overlap between objects in these images. The Saliency branch performs good enough for this kind of situation by predicting precise mask for each instance, indicating the effectiveness of this branch for pixel-wise alignment.

Moreover, the two settings of the Global Saliency branch are compared in Table 1e. The class-specific setting achieves 2.4 points higher than the class-agnostic counterpart, showing that the class-specific setting can help separate instances from different categories better.

For the class-specific version of Global Saliency branch, a binary cross-entropy loss is added to supervise the branch directly besides the mask loss Eq. (2). The comparison of CenterMask with or without the new loss is shown in Table 1f, direct supervision brings 0.5 points.

Combination of Local Shape and Global Saliency: Although the Saliency branch performs well in non-overlapping situations, it can not handle more complex images. We conduct the comparison of Shape-only, Saliency-only and the Combination of both in challenging conditions of instance segmentation. As shown in Figure 5c, objects overlap exists in these images. In the first column, the Shape branch separates different instances well, but the predicted masks are coarse. In the second column, the Saliency branch realizes precise segmentation but fails in the overlapping situations, which results in obvious artifacts on the overlapping area. CenterMask with both branches inherits their merits and avoid their weakness. As shown in the last column, overlapped objects are separated well and segmented precisely simultaneously, illustrating the effectiveness of our proposed model.

4.2. Comparison with state-of-the-art

In this section, we compare CenterMask with the state-of-the-art instance segmentation methods on the COCO[20] `test-dev` set.

As a one-stage instance segmentation method, our model follows a simple setting to perform the comparison: totally trained from scratch without pre-trained weights[6] for the backbone, using a single model with single-scale training and testing, and inference without any NMS.

As shown in Table 2, two models achieve higher AP than our method: the two-stage Mask R-CNN and the one-stage TensorMask, but their speed is 4 fps and 5 times slower than our largest model respectively. We think the gaps arise from the complicated and time-consuming feature align operations. Compared with the most accurate model of YOLACT [1], CenterMask with DLA-34 backbone achieves a higher AP with a faster speed. Compared with PolarMask [28], CenterMask with hourglass-104 backbone is 1.6 point higher with a faster speed.

Figure 6 shows the visualization of the results generated

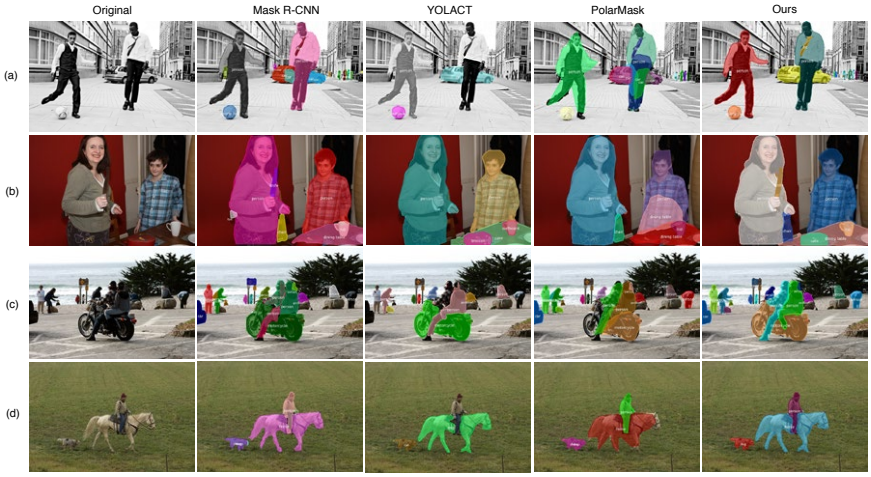


Figure 6: **Visualization comparison of three different instance segmentation methods.** From left to right are the results of : Original image, Mask R-CNN, YOLACT, PolarMask, and our method on COCO `minival` images.

by the state-of-the-art models, only comparing the ones that have released code. Mask R-CNN [12] detects objects well, but there are still artifacts in the masks, such as the heads of the two people in (a), we suppose it is caused by feature pooling. The YOLACT [1] segments instance precisely, but misses object in (d) and fails in some overlapping situations, such as the two legs in (c). The PolarMask can separate different instances, but its mask is not precise due to the polygon mask representation. Our CenterMask can separate overlapping objects well and segment masks precisely.

4.3. CenterMask on FCOS Detector

Besides CenterNet[30], the proposed Local Shape and Global Saliency branches can be embedded into other off-the-shelf detection models easily. FCOS[27], which is one of the state-of-the-art one stage object detectors, is utilized to perform the experiment. The performance of CenterMask built on FCOS with different backbones are shown in Table 3, with the training followings the same setting of Mask R-CNN[12]. With the same backbone of ResNeXt-101-FPN, CenterMask-FCOS achieves 3.8 points higher than PolarMask[28] in Table 2, and the best model achieves 38.5 mAP on COCO test-dev, showing the generalization of CenterMask.

To show the superiority of CenterMask on precise segmentation, we evaluate the model on the higher-quality LVIS annotations. The results are shown in Table 4. Based on the same backbone, the CenterMask-FCOS achieves better performance than Mask R-CNN.

Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet-101-FPN	36.1	58.7	38.0	16.5	38.4	51.2
ResNeXt-101-FPN	36.7	59.3	38.8	17.4	38.7	51.4
ResNet-101-FPN-DCN	37.6	60.4	39.8	17.3	39.8	53.4
ResNeXt-101-FPN-DCN	38.5	61.5	41.0	18.7	40.5	54.8

Table 3: **Performance of CenterMask-FCOS on COCO test-dev.** DCN represents deformable convolution[5].

Model	Backbone	AP
Mask R-CNN[12]	ResNet-101-FPN	36.0
CenterMask-FCOS	ResNet-101-FPN	40.0

Table 4: **Performance of CenterMask-FCOS on LVIS[10].** The AP of Mask R-CNN comes from the original LVIS paper.

5. Conclusion

In this paper, we propose a single shot and anchor-box free instance segmentation method, which is simple, fast and accurate. The mask prediction is decoupled into two critical modules: the Local Shape branch to separate different instances effectively and the Global Saliency branch to realize precise segmentation pixel-wisely. Extensive ablation experiments and visualization images show the effectiveness of the proposed CenterMask. We hope our work can help ease more instance-level recognition tasks.

Acknowledgements This research is supported by Beijing Science and Technology Project (No. Z181100008918018).

References

[1] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *ICCV*, 2019.

[2] Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollár. Tensormask: A foundation for dense object segmentation. *ICCV*, 2019.

[3] Jifeng Dai, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun. Instance-sensitive fully convolutional networks. In *ECCV*, pages 534–549. Springer, 2016.

[4] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, pages 3150–3158, 2016.

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009.

[7] Cheng-Yang Fu, Mykhailo Shvets, and Alexander C. Berg. RetinaMask: Learning to predict masks improves state-of-the-art single-shot detection for free. In *arXiv preprint arXiv:1901.03353*, 2019.

[8] Ross Girshick. Fast r-cnn. In *CVPR*, pages 1440–1448, 2015.

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[10] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5356–5364, 2019.

[11] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *ECCV*, pages 297–312. Springer, 2014.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[13] Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. Densebox: Unifying landmark localization with end to end object detection. *arXiv preprint arXiv:1509.04874*, 2015.

[14] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *CVPR*, pages 6409–6418, 2019.

[15] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 12 2014.

[16] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollar. Panoptic segmentation. In *CVPR*, June 2019.

[17] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, pages 734–750, 2018.

[18] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, pages 2359–2367, 2017.

[19] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014.

[21] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, pages 8759–8768, 2018.

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016.

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[24] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, pages 483–499. Springer, 2016.

[25] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016.

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[27] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *ICCV*, 2019.

[28] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. *arXiv preprint arXiv:1909.13226*, 2019.

[29] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *CVPR*, pages 2403–2412, 2018.

[30] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019.

[31] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krähenbühl. Bottom-up object detection by grouping extreme and center points. In *CVPR*, 2019.

Reference-guided Face Component Editing

Qiyao Deng[1,4], **Jie Cao**[1,4], **Yunfan Liu**[1,4], **Zhenhua Chai**[5], **Qi Li**[1,2,4*], **Zhenan Sun**[1,3,4]
[1]Center for Research on Intelligent Perception and Computing, NLPR, CASIA, Beijing, China
[2]Artificial Intelligence Research, CAS, Qingdao, China
[3]Center for Excellence in Brain Science and Intelligence Technology, CAS, Beijing, China
[4]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[5]Vision Intelligence Center, AI Platform, Meituandianping Group
{qiyao.deng, jie.cao, yunfan.liu}@cripac.ia.ac.cn, {qli, znsun}@nlpr.ia.ac.cn,
chaizhenhua@meituan.com

Abstract

Face portrait editing has achieved great progress in recent years. However, previous methods either 1) operate on pre-defined face attributes, lacking the flexibility of controlling shapes of high-level semantic facial components (e.g., eyes, nose, mouth), or 2) take manually edited mask or sketch as an intermediate representation for observable changes, but such additional input usually requires extra efforts to obtain. To break the limitations (e.g. shape, mask or sketch) of the existing methods, we propose a novel framework termed r-FACE (Reference-guided FAce Component Editing) for diverse and controllable face component editing with geometric changes. Specifically, r-FACE takes an image inpainting model as the backbone, utilizing reference images as conditions for controlling the shape of face components. In order to encourage the framework to concentrate on the target face components, an example-guided attention module is designed to fuse attention features and the target face component features extracted from the reference image. Through extensive experimental validation and comparisons, we verify the effectiveness of the proposed framework.

1 Introduction

Face portrait editing is of great interest in the computer vision community due to its potential applications in movie industry, photo manipulation, and interactive entertainment, etc. With advances in Generative Adversarial Networks [Goodfellow *et al.*, 2014] in recent years, tremendous progress has been made in face portrait editing [Yang *et al.*, 2018; Choi *et al.*, 2018; Liu *et al.*, 2019]. These approaches generally fall into three main categories: label-conditioned methods, geometry-guided methods and reference-guided methods. Specifically, label-conditioned methods [He *et al.*, 2019; Choi *et al.*, 2018] only focus on several pre-defined conspicuous attributes thus lacking the flexibility of controlling shapes of high-level semantic facial components (e.g., eyes, nose, mouth). This is because it is hard to produce results with observable geometric changes merely based on attribute labels. In order to tackle this, geometry-guided methods [Jo and Park, 2019; Gu *et al.*, 2019] propose to take manually edited mask or sketch as an intermediate representation for obvious face component editing with large geometric changes. However, directly taking such precise representations as a shape guide is inconvenient for users, which is laborious and requires painting skills. To solve this problem, reference-guided methods directly learns shape information from reference images without requiring precise auxiliary representation, relieving the dependence on face attribute annotation or precise sketch/color/mask information. As far as we know, reference-guided methods are less studied than the first two methods. ExGANs [Dolhansky and Canton Ferrer, 2018] utilizes exemplar information in the form of a reference image of the region for eye editing (in-painting). However, ExGANs can only edit eyes and requires reference images with the same identity, which is inconvenient to collect in practice. ELEGANT [Xiao *et al.*, 2018] transfers exactly the same type of attributes from a reference image to the source image by exchanging certain part of their encodings. However, ELEGANT is only used for editing obvious semantic attributes, and could not change abstract shapes.

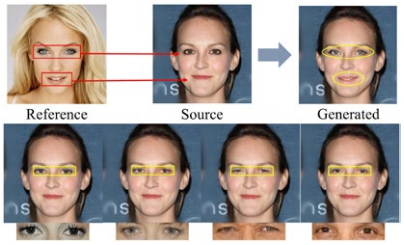


Figure 1: The illustration of reference guided face component editing. The first row is the definition diagram of the task, and the second row is the synthesized result based on the given different reference images.

To overcome the aforementioned problems, we propose a

*Contact Author

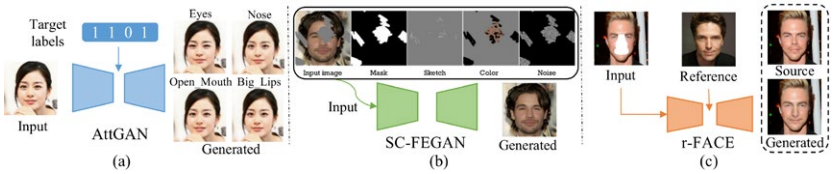


Figure 2: Different methods for face portrait editing. (a) AttGAN, (b) SC-FEGAN and (c) Our network.

new framework: **Reference guided FAce Component Editing** (r-FACE for short), which can achieve diverse and controllable face semantic components editing (e.g., eyes, nose, mouth) without requiring paired images. The ideal editing is to transfer single or multiple face components from a reference image to the source image, while still preserving the consistency of pose, skin color and topological structure(see Figure 1). Our framework breaks the limitations of existing methods: 1) shape limitation. r-FACE can flexibly control diverse shapes of high-level semantic facial components by different reference images; 2) intermediate presentation limitation. There is no need to manually edit precise masks or sketches for observable geometric changes.

Our framework is based on an image inpainting model for editing face components by reference images even without paired images. r-FACE has two main streams including 1) an inpainting network $\mathcal{G}_i$ and 2) an embedding network $\mathcal{E}_r$. As shown in Figure 3, $\mathcal{G}_i$ takes the source image with target face components corrupted and the corresponding mask image as input, and outputs the generated image with semantic features extracted by $\mathcal{E}_r$. To encourage the framework to concentrate on the target face components, an example-guided attention module is introduced to combine features extracted by $\mathcal{G}_i$ and $\mathcal{E}_r$. To supervise the proposed model, a contextual loss is adopted to constrain the similarity of shape between generated images and reference images, while a style loss and a perceptual loss are adopted to preserve the consistency of skin color and topological structure between generated images and source images. Both qualitative and quantitative results demonstrate that our model is superior to existing literature by generating high-quality and diverse faces with observable changes for face component editing.

In summary, the contributions of this paper are as follows:

- We propose a novel framework named reference guided face component editing for diverse and controllable face component editing with geometric changes, which breaks the shape and intermediate presentation (e.g., precise masks or sketches) limitation of existing methods.
- An example-guided attention module is designed to encourage the framework to concentrate on the target face components by combining attention features and the target face component features of the reference image, further boosting the performance of face portrait editing.
- Both qualitative and quantitative results demonstrate the superiority of our method compared with other benchmark methods.

2 Related Work

Face Portrait Editing. Face portrait editing aims at manipulating single or multiple attributes or components of a face image towards given conditions. Depending on different conditions, face portrait editing methods can be classified into three categories: label-conditioned methods, geometry-guided methods and reference-guided methods. Label-conditioned methods change predefined attributes, such as hair color [Choi *et al.*, 2018], age [Liu *et al.*, 2019] and pose [Cao *et al.*, 2019]. However, these methods focus on several conspicuous attributes [Liu *et al.*, 2015; Langner *et al.*, 2010], lacking the flexibility of controlling the shapes of different semantic facial parts. As shown in Figure 2(a), AttGAN [He *et al.*, 2019] attempts to edit attributes with shape changes, such as $'Narrow\_Eyes'$, $'Pointy\_Nose'$ and $'Mouth\_Slightly\_Open'$, but it can only achieve subtle changes hard to be observed. Moreover, lacking of labeled data will extremely limit the performance of these methods. To tackle above problems, geometry-guided methods use an intermediate representation to guide observable shape changes. [Gu *et al.*, 2019] proposes a framework based on mask-guided conditional GANs which can change the shape of face components by manual editing precise masks. As shown in Figure 2(b), SC-FEGAN [Jo and Park, 2019] requires directly taking mask, precise sketch and color as input for editing the shape of face components. However, such precise input is difficult and inconvenient to obtain. Reference-guided methods can directly learn shape information from reference images without precise auxiliary representation, relieving the dependence on face attribute annotation or precise sketch/color/mask information for face portrait editing. Inspired by this, we propose a new framework (see Figure 2(c)), which can achieve diverse and controllable face semantic components editing (e.g., eyes, nose, mouth), which is shape free and precise landmark or sketch free.

Face Completion/Inpainting. Face completion, also known as face inpainting, aims to complete a face image with a masked region or missing content. Early face completion works [Bertalmio *et al.*, 2000; Criminisi *et al.*, 2003; Bertalmio *et al.*, 2003] fill semantic contents based on the overall image and structural continuity between the masked and unmasked regions, which aims to reconstruct missing regions according to the ground-truth image. Recently, some learning-based method [Zheng *et al.*, 2019; Song *et al.*, 2019] are proposed for generating multiple and diverse plausible results. [Zheng *et al.*, 2019] proposes a probabilistically principled framework with two parallel paths, the VAE-based reconstructive path is used to impose smooth priors for the la-

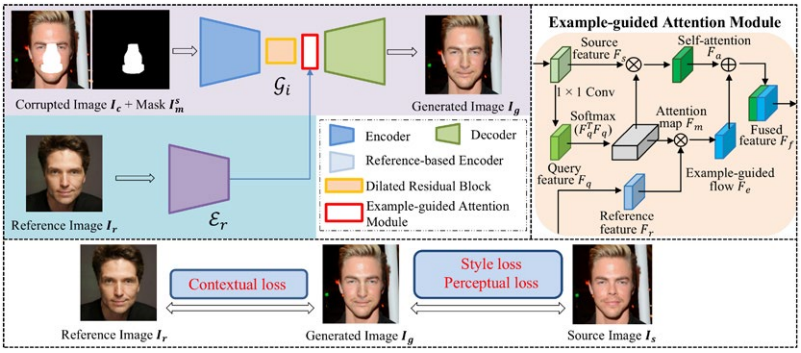


Figure 3: The overall structure of proposed framework. On the top left corner is the generator. The top-right figure shows detailed attention module. The constrains among the source image, the reference image and the generated image is shown on the bottom.

tent space of complement regions, the generative path is used to predict the latent prior distribution for missing regions, which can be sampled to generate diverse results. However, this method lacks controllability for diverse results. In light of this, [Song *et al.*, 2019] generate controllable diverse results from the same masked input by manual modifying facial landmark heatmaps and parsing maps.

3 Method

In this section, we first introduce the framework of reference-based face component editing. Then, the example-guided attention module are presented. Finally, objective functions of the proposed model are provided.

3.1 Reference Guided Face Component Editing

We propose a framework (See Figure 3), named reference guided face component editing, that transfers one or multiple face components of a reference image to corresponding components of the source image. The framework requires three inputs, a source image $\mathbf{I}_s$, a reference image $\mathbf{I}_r$ and the target face component mask of the reference image $\mathbf{I}_m^s$. The source mask $\mathbf{I}_m^s$ merely needs to roughly represent the target face components, which can be obtained by a face parsing or landmark detection network. The corrupted image can be obtained by Equation 1:

$$\mathbf{I}_c = \mathbf{I}_s * \mathbf{I}_m^s, \tag{1}$$

where $*$ is an element-wise multiplication operator. The goal of this framework is to generate a photo-realistic image $\mathbf{I}_g$, in which shape is semantically similar to corresponding face components of the reference image while the face color and topological structure are consistent with the source image.

In this work, we utilize an image inpainting generator $\mathcal{G}_i$ as backbone that can generate the completed image without constraints on shape, while a discriminator $\mathcal{D}$ is used for distinguishing face images from the real and synthesized domains. $\mathcal{G}_i$ is consist of an encoder, seven dilated residual blocks, an attention module and a decoder. To fill missing parts with semantically meaningful face components of a reference image, an reference-guided encoder $\mathcal{E}_r$ is introduced to extract features of the reference image. The encoder of $\mathcal{G}_i$ and $\mathcal{E}_r$ have same structure but parameters are not shared. Attention module, to be described next, is effectively transferring semantic components from high-level features of the reference image to $\mathcal{G}_i$, further improving the performance of our framework. The generated image $\mathbf{I}_g$ can be expressed as:

$$\mathbf{I}_g = \mathcal{G}_i(\mathbf{I}_c, \mathbf{I}_m^s, \mathcal{E}_r(\mathbf{I}_r)), \tag{2}$$

3.2 Example-guided Attention Module

Inspired by the short+long term attention of PICNet [Zheng *et al.*, 2019], we propose an example-guided attention. The short+long term attention uses the self-attention map to harness distant spatial context and the contextual flow to capture feature-feature context for finer-grained image inpainting. The example-guided attention replaces the contextual flow by the example-guided flow which combines the attention features and the reference features for clearly transforming the corresponding face component features of reference images to source images.

The proposed structure is shown in the upper right corner of Figure 3. Following [Zheng *et al.*, 2019], the self-attention map is calculated from the source feature, which can be expressed as $\mathbf{F}_a = \mathbf{F}_s \otimes \mathbf{F}_m$. The attention map $\mathbf{F}_m$ is obtained by $\mathbf{F}_m = Softmax(\mathbf{F}_q^T \mathbf{F}_q)$, in which $\mathbf{F}_q = Conv(\mathbf{F}_s)$ and $Conv$ is a 1×1 convolution filter. To transfer the target face component features of reference images, the reference feature is embedded by multiplying the attention map with the source mask $\mathbf{I}_m^s$. The example-guided flow is expressed as follows:

$$\mathbf{F}_e = \mathbf{I}_m^s * \mathbf{F}_e' + (1 - \mathbf{I}_m^s) * \mathbf{F}_r, \tag{3}$$

where $\mathbf{F}_e' = \mathbf{F}_r \otimes \mathbf{F}_m$. Finally, the fused feature $\mathbf{F}_f = \mathbf{F}_a \oplus \mathbf{F}_e$ is sent to the decoder for generating results with the target components of the reference image.

3.3 Objective

We use the combination of a per-pixel loss, a style loss, a perceptual loss, a contextual loss, a total variation loss and an adversarial loss for training the framework.

To capture fine facial details we adopt the perceptual loss [Johnson *et al.*, 2016] and the style loss [Johnson *et al.*, 2016], which are widely adopted in style transfer, super resolution, and face synthesis. The perceptual loss aims to measure the similarity of the high dimensional features (e.g., overall spatial structure) between two images, while the style loss measure the similarity of styles (e.g., colors). The perceptual loss can be expressed as follows:

$$\mathcal{L}_{perc} = \sum_{l} \frac{1}{C_l H_l W_l} \|\phi_l(\mathbf{I}_g) - \phi_l(\mathbf{I}_s)\|_1, \quad (4)$$

The style loss compare the content of two images by using Gram matrix, which can be expressed as follows:

$$\mathcal{L}_{style} = \sum_{l} \frac{1}{C_l C_l} \| \frac{G_l(\mathbf{I}_g * \mathbf{I}_m^s) - G_l(\mathbf{I}_c)}{C_l H_l W_l} \|_1, \quad (5)$$

where $\| \cdot \|_1$ is the ℓ_1 norm. $\phi_l(\cdot) \in \mathbb{R}^{C_l \times H_l \times W_l}$ represents the feature map of the l-th layer of the VGG-19 network [Simonyan and Zisserman, 2014] pretrained on the ImageNet. $G_l(\cdot) = \phi_l(\cdot)^T \phi_l(\cdot)$ represents the Gram matrix corresponding to $\phi_l(\cdot)$.

The contextual loss [Mechrez *et al.*, 2018] measures the similarity between non-aligned images, which in our model effectively guarantees the consistent shape of the target face components in generated images and reference images. Given an image x and its target image y, each of which is represented as a collection of points(e.g. VGG-19 [Simonyan and Zisserman, 2014] features): $X = \{x_i\}$ and $Y = \{y_j\}$, $|X| = |Y| = N$. The similarity between the images can be calculated that for each feature y_j, finding the feature x_i that is most similar to it, and then sum the corresponding feature similarity values over all y_j. Formally, it is defined as:

$$CX(x, y) = CX(X, Y) = \frac{1}{N} \sum_{j} \max_{i} CX_{ij}, \quad (6)$$

where CX_{ij} is the similarity between features x_i and y_j. At training stage, we need the target components mask of reference images I_m^r for calculating the contextual loss. The contextual loss can be expressed as follows:

$$\mathcal{L}_{cx} = -\log(CX(\phi_l(\mathbf{I}_g * \mathbf{I}_m^s), \phi_l(\mathbf{I}_r * \mathbf{I}_m^r))), \quad (7)$$

The per-pixel loss can be expressed as follows:

$$\mathcal{L}_{pixel} = \|\phi_l(\mathbf{I}_g * \mathbf{I}_m^s) - \phi_l(\mathbf{I}_s * \mathbf{I}_m^s)\|_1, \quad (8)$$

Lastly, we use an adversarial loss for minimizing the distance between the distribution of the real image and the generated image. Here, LSGAN [Mao *et al.*, 2017] is adopted for stable training. The adversarial loss can be expressed as follows:

$$\mathcal{L}_{ad_G} = \mathbb{E}[(\mathcal{D}(\mathbf{I}_g) - 1)^2], \quad (9)$$

$$\mathcal{L}_{ad_D} = \mathbb{E}[\mathcal{D}(\mathbf{I}_g)^2] + \mathbb{E}[(\mathcal{D}(\mathbf{I}_s) - 1)^2], \quad (10)$$

With total variational regularization loss $\mathcal{L}_{tv}$ [Johnson *et al.*, 2016] added to encourage the spatial smoothness in the generated images, we obtain our full objective as:

$$\begin{aligned} \mathcal{L} = \lambda_1 \mathcal{L}_{perc} + \lambda_2 \mathcal{L}_{style} + \lambda_3 \mathcal{L}_{cx} \\ + \lambda_4 \mathcal{L}_{pixel} + \lambda_5 \mathcal{L}_{tv} + \lambda_6 \mathcal{L}_{ad_G} \end{aligned} \quad (11)$$

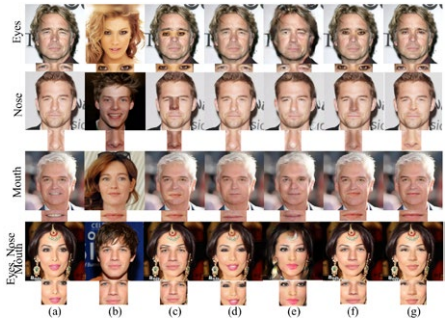


Figure 4: Comparisons among (c) copy-and-paste, (d) AttGAN, (e) ELEGANT, (f) the commercial state of the art algorithm in Adobe Photoshop and (g) the proposed r-FACE technique. The source and reference images are shown in (a) and (b), respectively. AttGAN and ELEGANT edit attributes: $'Narrow\_Eyes'$, $'Pointy\_Nose'$,$'Mouth\_Slightly\_Open'$ and all of them respectively.

4 Experiments

4.1 Dataset and Preprocessing

The face attribute dataset CelebAMask-HQ [Lee *et al.*, 2019] contains 30000 aligned facial images with the size of 1024×1024 and corresponding 30000 semantic segmentation labels with the size of 512×512. Each label in the dataset has 19 classes (e.g., "left eye", "nose", "mouth"). In our experiments, three face components are considered, i.e., eyes ("left eye & right eye"), nose ("nose"), and mouth ("mouth & u_lip & l_lip"). We obtain rough version of face components from semantic segmentation labels by an image dilation operation, which are defined as mask images. We take 2,000 images as the test set for performance evaluation, using rest images to train our model. All images are resized to 256×256.

4.2 Implementation Details

Our end-to-end network is trained on four GeForce GTX TITAN X GPUs of 12GB memory. Adam optimizer is used in experiments with $\beta_1 = 0.0$ and $\beta_2 = 0.9$. The hyperparameters from λ_1 to λ_6 are assigned as 0.1, 250, 1, 0.5, 0.1, 0.01 respectively. For each source image, we remove two or three target face components for training. During testing, our model can change one or more face components.

4.3 Qualitative Evaluation

Comparison with Other Methods. In order to demonstrate the superiority of the proposed method, we compare the quality of sample synthesis results to several benchmark methods. In addition to face editing model AttGAN [He *et al.*, 2019] and ELEGANT [Xiao *et al.*, 2018], we also consider copy-and-paste as a naive baseline and Adobe Photoshop image editing as an interactive way to produce synthesized face images. According to the results presented in Figure 4, although margins of edited facial components in Adobe

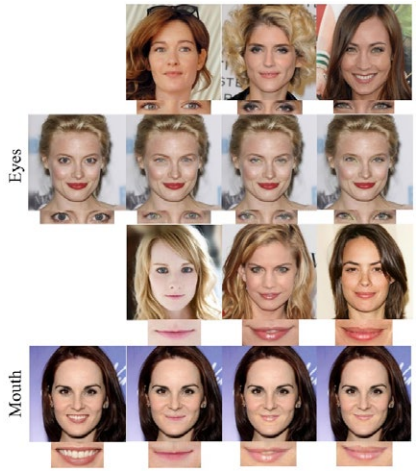


Figure 5: Illustrations of multi-modal face components editing, including 'eyes' and 'mouth'. The first column represents source images, the first and third rows are reference images, and the second and fourth rows are synthesized images according to reference images above.

Photoshop are much smoother than those in results of copy-and-paste, obvious ghosting artifacts and color distortions still exist and more fine-grained manual labour is required to improve the quality. In contrast, AttGAN could generate realistic synthetic images which are indistinguishable to generic ones in an end-to-end manner. However, pre-defined facial attribute labels are used to guide AttGAN to transform facial components thus the diversity of generated images is limited. Moreover, as can be seen in Figure 4, AttGAN only produces subtle changes that could hardly reflect the desired translation. ELEGANT, as a reference-guided face attribute editing method, can learn obvious semantic attributes (e.g., open eyes or close mouth), but could not learn abstract shape information (e.g., nose editing). Moreover, ELEGANT produces large deformation and unexpected artifacts at other attribute-unrelated areas, especially the editing of multiple components. On the contrary, our method takes arbitrary face images as reference, which significantly increases the diversity of generated images.

Multi-Modal Face Components Editing. Reference-guided face components editing improves the diversity and controllability of generated faces, as the stylistic information is designated by arbitrary reference images. As shown in Figure 5, target face components of interest (e.g. eyes and mouths) are transformed to be of the same style as the corresponding reference image. For example, synthesized mouths in faces of the last row accurately simulate the counterpart in reference images, in terms of both overall shape (e.g. raised corners of pursed mouth) and local details (e.g. partially covered teeth). Meanwhile, they are naturally blended in to the source face without observable color distortions and ghosting artifacts, demonstrating the effectiveness of proposed method.

Method	FID ↓	MS-SSIM ↑
GLCIC[Iizuka *et al.*, 2017]	8.09	0.95
AttGAN[He *et al.*, 2019]	6.28	0.96
ELEGANT [Xiao *et al.*, 2018]	15.97	0.82
r-FACE (Ours)	**5.81**	**0.92**
w / o attention	6.25	0.90
w / o contextual loss	5.27	0.90
w / o style loss	8.28	0.90
w / o perceptual loss	8.49	0.89

Table 1: Comparisons of FID and MS-SSIM on the CelebA-HQ dataset.

4.4 Quantitative Evaluation

Following most of face portrait methods [He *et al.*, 2019; Wu *et al.*, 2019], we leverage Fréchet Inception Distance (FID, lower value indicates better quality) [Heusel *et al.*, 2017] and Multi-Scale Structural SIMilarity (MS-SSIM, higher value indicates better quality) [Wang *et al.*, 2003] to evaluate the performance of our model. FID is used to measure the quality and diversity of generated images. MS-SSIM is used to evaluate the similarity of two images from luminance, contrast, and structure.

We compare our method with AttGAN[He *et al.*, 2019], one of label-conditioned methods, and ELEGANT [Xiao *et al.*, 2018], one of reference-guided methods. For AttGAN and ELEGANT, the value of FID or MS-SSIM are the result of averaging three pre-defined attributes for shape changing, including 'Narrow_Eyes', 'Pointy_Nose' and 'Mouth_Slightly_Open'. The backbone of r-FACE is similar to image inpainting task, so we also compare our method with GLCIC[Iizuka *et al.*, 2017], one of popular face inpainting methods. As is shown in TABLE 1, comparing with these methods, the FID of our method is much lower. With the observation that the MS-SSIM of our method is lower than AttGAN and GLCIC, we analyze the reasons: MS-SSIM is sensitive to luminance, contrast, and structure, however, 1) GLCIC does not have any constraints on the structure or shape of components, just requiring the missing regions to be completed; 2) AttGAN edits shape-changing attributes with subtle changes that are hard to be observed as shown in Figure 4, so the change of luminance, contrast, and structure is limited. In contrast, r-FACE imposes a geometric similarity constraint on the components of source images and reference images, which changes the shape or structure drastically and even affects the identity of the face.

4.5 Ablation Study

Example-guided Attention Module. To investigate the effectiveness of the example-guided attention module, we conduct a variant of our model, denoted as "r-FACE w / o attention". In "r-FACE w / o attention", we train r-FACE without example-guided attention module. To learn face components from a reference image, we directly concatenate the fea-

tures of reference images with that of source images, which introduced the features of the whole reference image. TABLE 1 shows the comparison on FID and MS-SSIM between "r-FACE w / o attention" and "r-FACE". Our method outperforms "r-FACE w / o attention" on both FID and MS-SSIM, which indicates that example-guided attention module can explicitly transfer the corresponding face components of a reference image and reduce the impact of other information of the reference image.

Loss Configurations. In this section, we conduct ablation studies on different loss, such as the contextual loss, the style loss and the perceptual loss, for evaluating their individual contributions on our framework. As shown in TABLE 1, the quantitative comparison among "r-FACE w / o contextual loss", "r-FACE w / o style loss", "r-FACE w / o perceptual loss" and "r-FACE" are presented. "r-FACE" outperforms other loss configurations on MS-SSIM, which indicates the effectiveness of the three losses. It is observed that the FID of "r-FACE" is also better than other loss configurations except "r-FACE w/o contextual loss". We argue that contextual loss plays an important role in restricting the shape of face components. The framework has no constraints on the shape after removing the contextual loss, so "r-FACE w / o contextual loss" is easier to generate missing components, merely requiring generated images to look real. Figure 6 compares the visual effects of different loss configurations. We find "r-FACE w / o contextual loss" could not synthesize components with the corresponding shape of reference images, which indicates the contextual loss is significantly important in shape constraints. In the results of "r-FACE w / o style loss" and "r-FACE w / o perceptual loss", color distortions and obvious ghosting artifacts are observed, which show the style loss and the perceptual loss are able to preserve skin color of source image. Above all, we prove that three losses contribute to the performance of our framework and the combination of all losses achieves the best results.

4.6 Discussion and Limitation

Reference guided face component editing has wide real-life applications in interactive entertainment, security and data augmentation. Given whole reference images of any identity, r-FACE performs various face component editing, which can achieve the effect of "plastic surgery". Besides, when all face components of a reference face are transformed to the source face, r-FACE is further extended to face swapping task. Moreover, As the face component is a part of face, editing face components may change the identity of the person. Therefore, r-FACE can be used for data augmentation to generate face images of different identities. Although r-FACE obtains diverse and controllable face component editing results, reference images with significant differences in pose are still challenging for our model. We will continue to explore solving extreme pose problems in further work.

5 Conclusion

In this work, we have proposed a novel framework, reference guided face components editing (r-FACE), for high-level face components editing, such as eyes, nose and mouth.

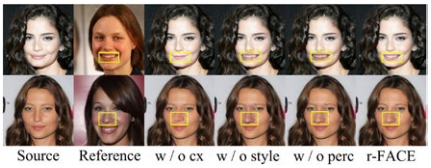


Figure 6: Visual comparisons for different loss configurations, including "r-FACE w / o contextual loss", "r-FACE w / o style loss", "r-FACE w / o perceptual loss" and "r-FACE".

r-FACE can achieve diverse, high-quality, and controllable component changes in shape from given references, which breaks the shape and precise intermediate presentation limitation of existing methods. For embedding the target face components of reference images to source images specifically, an example-guided attention module is designed to fuse the features of source images and reference images, further boosting the performance of face component editing. The extensive experiments demonstrate that our framework can achieve state-of-art face editing results with observable geometric changes.

Acknowledgments

This work was partially supported by the Natural Science Foundation of China (Grant No. U1836217, Grant No. 61702513, Grant No. 61721004, and Grant No. 61427811). This research was also supported by Meituan-Dianping Group, CAS-AIR and Shandong Provincial Key Research and Development Program (Major Scientific and Technological Innovation Project) (NO.2019JZZY010119).

References

[Bertalmio *et al.*, 2000] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *SIGGRAPH*, pages 417–424, 2000.

[Bertalmio *et al.*, 2003] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher. Simultaneous structure and texture image inpainting. *IEEE TIP*, 12(8):882–889, 2003.

[Cao *et al.*, 2019] Jie Cao, Yibo Hu, Bing Yu, Ran He, and Zhenan Sun. 3D aided duet GANs for multi-view face image synthesis. *IEEE TIFS*, 14(8):2028–2042, 2019.

[Choi *et al.*, 2018] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, pages 8789–8797, 2018.

[Criminisi *et al.*, 2003] Antonio Criminisi, Patrick Perez, and Kentaro Toyama. Object removal by exemplar-based inpainting. In *CVPR*, volume 2, pages II–II, 2003.

[Dolhansky and Canton Ferrer, 2018] Brian Dolhansky and Cristian Canton Ferrer. Eye in-painting with exemplar generative adversarial networks. In *CVPR*, pages 7902–7911, 2018.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014.

[Gu *et al.*, 2019] Shuyang Gu, Jianmin Bao, Hao Yang, Dong Chen, Fang Wen, and Lu Yuan. Mask-guided portrait editing with conditional gans. In *CVPR*, pages 3436–3445, 2019.

[He *et al.*, 2019] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. AttGAN: Facial attribute editing by only changing what you want. *IEEE TIP*, 2019.

[Heusel *et al.*, 2017] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash Equilibrium. In *NeurIPS*, pages 6626–6637, 2017.

[Iizuka *et al.*, 2017] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM ToG*, 36(4):107, 2017.

[Jo and Park, 2019] Youngjoo Jo and Jongyoul Park. SC-FEGAN: Face editing generative adversarial network with user's sketch and color. In *ICCV*, October 2019.

[Johnson *et al.*, 2016] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711, 2016.

[Langner *et al.*, 2010] Oliver Langner, Ron Dotsch, Gijsbert Bijlstra, Daniel HJ Wigboldus, Skyler T Hawk, and AD Van Knippenberg. Presentation and validation of the Radboud Faces Database. *Cognition and Emotion*, 24(8):1377–1388, 2010.

[Lee *et al.*, 2019] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. MaskGAN: Towards diverse and interactive facial image manipulation. *arXiv preprint arXiv:1907.11922*, 2019.

[Liu *et al.*, 2015] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, pages 3730–3738, 2015.

[Liu *et al.*, 2019] Yunfan Liu, Qi Li, and Zhenan Sun. Attribute-aware face aging with wavelet-based generative adversarial networks. In *CVPR*, pages 11877–11886, 2019.

[Mao *et al.*, 2017] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, pages 2794–2802, 2017.

[Mechrez *et al.*, 2018] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. The contextual loss for image transformation with non-aligned data. In *ECCV*, pages 768–783, 2018.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Song *et al.*, 2019] Linsen Song, Jie Cao, Linxiao Song, Yibo Hu, and Ran He. Geometry-aware face completion and editing. In *AAAI*, pages 2506–2513, 2019.

[Wang *et al.*, 2003] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *ACSSC*, volume 2, pages 1398–1402, 2003.

[Wu *et al.*, 2019] Po-Wei Wu, Yu-Jing Lin, Che-Han Chang, Edward Y Chang, and Shih-Wei Liao. RelGAN: Multi-domain image-to-image translation via relative attributes. In *ICCV*, pages 5914–5922, 2019.

[Xiao *et al.*, 2018] Taihong Xiao, Jiapeng Hong, and Jinwen Ma. Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. In *ECCV*, pages 168–184, 2018.

[Yang *et al.*, 2018] Hongyu Yang, Di Huang, Yunhong Wang, and Anil K Jain. Learning face age progression: A pyramid architecture of gans. In *CVPR*, pages 31–39, 2018.

[Zheng *et al.*, 2019] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *CVPR*, pages 1438–1447, 2019.

Data Efficient Voice Cloning from Noisy Samples with Domain Adversarial Training

Jian Cong[1], *Shan Yang*[1], *Lei Xie*[1†], *Guoqiao Yu*[2], *Guanglu Wan*[2]

[1]Audio, Speech and Language Processing Group (ASLP@NPU), School of Computer Science, Northwestern Polytechnical University, Xi'an, China
[2]Meituan-Dianping Group, Beijing, China

npujcong@mail.nwpu.edu.cn,{yuguoqiao, wanguanglu}@meituan.com

Abstract

Data efficient voice cloning aims at synthesizing target speaker's voice with only a few enrollment samples at hand. To this end, speaker adaptation and speaker encoding are two typical methods based on base model trained from multiple speakers. The former uses a small set of target speaker data to transfer the multi-speaker model to target speaker's voice through direct model update, while in the latter, only a few seconds of target speaker's audio directly goes through an extra speaker encoding model along with the multi-speaker model to synthesize target speaker's voice without model update. Nevertheless, the two methods need clean target speaker data. However, the samples provided by user may inevitably contain acoustic noise in real applications. It's still challenging to generating target voice with noisy data. In this paper, we study the data efficient voice cloning problem from noisy samples under the sequence-to-sequence based TTS paradigm. Specifically, we introduce domain adversarial training (DAT) to speaker adaptation and speaker encoding, which aims to disentangle noise from speech-noise mixture. Experiments show that for both speaker adaptation and encoding, the proposed approaches can consistently synthesize clean speech from noisy speaker samples, apparently outperforming the method adopting state-of-the-art speech enhancement module.

Index Terms: Speech synthesis, voice cloning, speaker adaptation, speaker encoding, adversarial training.

1. Introduction

Sequence-to-sequence (seq2seq) neural network based text-to-speech (TTS) is able to synthesize natural speech without a complex front-end analyzer and an explicit duration module [1]. However, a sizable amount of high quality audio-text paired data is necessary to build such systems, which limits the model ability to produce natural speech for a target speaker without enough data. Therefore, building target voice with few minutes or even few samples data, or *voice cloning*, has drawn many interests lately [2, 3, 4, 5]. In order to produce target speaker voice in a data efficient manner, there are several attempts to build multi-speaker model to produce target voice from a few clean samples, most of which can be divided into two categories [2]: *speaker adaptation* and *speaker encoding*. In both families, a multi-speaker base model is required to generate target voice.

The core idea for speaker adaptation methods [6, 7] is to fine-tune the pre-trained multi-speaker model with a few audio-text pairs for an unseen speaker to produce target voice. The transcription of target speaker samples can be obtained by speech recognition to fine-tune the base model [8]. The study in [9] demonstrates that the training strategy cannot be fixed for adaptation of different speakers and presents a Bayesian optimization method for fine-tuning the TTS model. As for speaker encoding, it mainly builds an extern speaker encoder model to obtain continuous speaker representations for subsequent multi-speaker training. The same extern speaker encoder is then utilized to obtain the speaker embedding from audio samples of an unseen speaker. Without further fune-tuning, the speaker embedding is directly fed into the multi-speaker model to result in target's voice. As the ability and robustness of speaker representation module directly decides the performance of adaptation, several speaker representation methods have been evaluated for adaptive speech synthesis [4]. Comparing the above two families, speaker adaptation can achieve better speaker similarity and naturalness, while speaker encoding does not need any extra adaptation procedure and audio-text pairs, achieving so-called one/few-shot(s) voice cloning.

Approaching data efficient voice cloning either through speaker adaptation or via speaker encoding, clean speech samples from target speaker is usually necessary to produce clean target voice. However, in practical voice cloning applications, target speaker data is often either acquired in daily acoustic conditions or found data from Internet, with inevitable background noise. It is still challenging generating target voice with noisy target speaker data, especially for systems built upon the current seq2seq paradigm in which attention-based soft alignment is vunarable to interferences [10]. In order to build a robust TTS system, there are several attempts to conduct speech synthesis with noisy data [10, 11, 12]. An alternative method is to de-noise the noisy training data with an external speech enhancement module [11], but the audible or inaudible spectrum distortion may inevitably affect the quality of the generated speech. Besides, we can also try to *disentangle* noise and other attributes in audio. The approach in [13] aims to disentangle speech and non-speech components using variational auto encoders (VAE) [14] to enable the use of found data for TTS applications. And in [15], through speaker and noise attributes disentangling during training, the model is able to control different aspects of the synthesized speech with different reference audios. But prior researches on robust TTS have mainly worked on training on large-scale found or noisy dataset, data efficient voice cloning for noisy data has rarely been considered.

In this paper, we focus on how to produce target voice in both speaker adaptation and speaker encoding scenes with only a few noisy samples of the target speaker under state-of-the-art seq2seq TTS framework. For the speaker adaptation method, we find the model usually cannot converge at adaptation time

†Corresponding author. This research work is supported by the National Key Research and Development Program of China (No.2017YFB1002102).

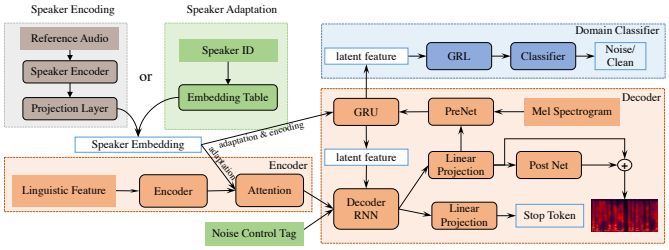


Figure 1: *Basic seq2seq TTS model, speaker adaptation and speaker encoding architecture. The components with dotted orange outlines are common basic model with additional GRU. The components with dotted blue outlines are proposed domain classifier module. Speaker adaptation extends basic model with DAT module, speaker embedding looking up table, and noise control tag, shown as green components. Speaker encoding extends basic model with DAT module and external speaker encoding network, shown as gray components. Note that the speaker embedding only injects to GRU at speaker encoding.*

with noisy speaker data. So we assume the main challenging problem is how to fine-tune the base multi-speaker model with noisy data and produce clean target speech. As for the speaker encoding based synthesis model, the main issue is that the speaker representation usually contains noise information, which directly affects the performance of generated speech as the speaker encoding has deviated because of the interference.

To overcome the above issues in both speaker adaptation and encoding methods, we propose a robust seq2seq framework to conduct target speaker's voice cloning with noisy data. For this purpose, we introduce domain adversarial training (DAT) [16] to both methods to learn noise-invariant latent features. Specifically, we extend the decoder with a domain classifier network with a gradient reverse layer (GRL) for the speaker adaptation method, trying to disentangle the noise condition in acoustic features. For speaker encoding, since the speaker embedding extracted from the speaker encoder network is noise-dependent, we disentangle the noise condition in the speaker embedding with the help of domain adversarial training, leading to noise-invariant speaker embedding. Note that DAT has been previously studied in speech recognition [17, 18, 19], speaker verification [20, 21] as well as speech synthesis [15, 10] tasks with superior performance in learning noise-invariant features and attribute disentanglement. To the best of our knowledge, our study in the first one examining its efficacy in data efficient voice cloning. Our study shows that for both speaker adaptation and encoding, the proposed approach can consistently synthesize clean speech from noisy speaker samples, apparently outperforming the method adopting a speech enhancement module.

2. Proposed Method

Fig. 1 illustrates the proposed seq2seq-based multi-speaker model for data efficient voice cloning in noisy conditions. The proposed architecture contains a CBHG-based text encoder [22], an auto-regressive decoder with GMM-based attention [23], the domain adversarial training module, and the speaker representation module.

For the basic seq2seq framework, the model generates mel-spectrogram $m = (m_1, m_2, \cdots, m_M)$ frame by frame given a text sequence $t = (t_1, t_2, \cdots, t_N)$, where M and N are the length of acoustic features and linguistic features respectively. The text sequence t is firstly fed into the text encoder:

$$x = e(t|\Theta_e) \tag{1}$$

where $e(\cdot)$ represents the text encoder and x is the text representation from the encoder.

During the auto-regressive process, the decoder takes current frame of spectrogram m_t to produce next frame m_{t+1}. In detail, the decoder firstly converts m_t into latent representation z_t through a pre-net $h(\cdot)$, where the z_t acts as an information bottleneck. The z_t is then treated as a query to compute context vector c_t with x through GMM-based attention module $g(\cdot)$. Hence, the next frame m_{t+1} can be calculated from the context vector c_t and z_t through transformation function $f(\cdot)$:

$$z_t = h(m_t|\Theta_h) \tag{2}$$

$$c_t = g(z_t, x|\Theta_g) \tag{3}$$

$$\hat{m}_{t+1} = f(c_t, z_t|\Theta_f) \tag{4}$$

where Θ_h, Θ_g and Θ_f represent the module parameters of pre-net, attention mechanism and transformation, respectively. We minimize the mean square error between predicted $\hat{m}_t$ and ground truth m_t to optimize the whole model:

$$L_{rcon} = ||m - \hat{m}||_1. \tag{5}$$

2.1. Few-shots robust speaker adaptation with DAT

To conduct speaker adaptation for noisy data, we firstly build a multi-speaker model with both noisy and clean speech samples. Based on the basic architecture, we adopt an extra trainable speaker embedding table to bring speaker identity. For each speech sample m, the speaker representation s is obtained from the embedding table indexed by the corresponding speaker label. We concatenate the speaker embedding s with pre-net and encoder output, so Eq. (3) and Eq. (4) become

$$z_t = h(m_t, s|\Theta_h) \tag{6}$$

and

$$c_t = g(z_t, x, s|\Theta_g). \tag{7}$$

In order to build a robust multi-speaker model for few-shots noisy samples in the adaptation stage, we use both clean and noisy speech data during training. As shown in Eq (6), the latent feature z may contain noise interference when m is noisy. In order to encourage z to become noise-independent feature, we inject a GRU layer into the $h(\cdot)$ and then employ a domain classifier with gradient reversal layers (GRL) on the output z of GRU layer at frame level. The proposed latent z is adopted to predict the noisy/clean label for the following domain classifier. We further feed noisy/clean embedding vector into decoder RNN to control the generation process. With the auxiliary classifier, the final loss function in Eq (5) becomes:

$$L = L_{rcon} + \lambda L_{noise\_ce} \tag{8}$$

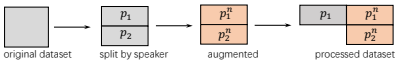


Figure 2: *Data augmentation process for speaker adaptation base model.*

where λ is the tunable weight for domain classifier loss. In order to obtain a multi-speaker corpus for the above domain adversarial training, we apply data augmentation on a clean multi-speaker dataset, as shown in Fig. 2. Specifically, we split the original training set into two subsets p_1 and p_2, both of which contains multiple speakers. Then we add randomly selected background noise at a random signal-to-noise ratio (SNR) in p_1 and p_2 to obtain the noisy counterparts p_1^n and p_2^n. Finally, the subsets p_1, p_1^n and p_2^n are treated as the training set to train the above multi-speaker model. Note that there are no clean speech for the speakers in sub-set p_2^n, which refers to the speaker adaptation scenario with only noisy speech for each speaker.

For few-shots speaker adaptation scene, there are only several noisy audio clips with transcriptions of the target speaker. We utilize the above noisy samples to fine-tune the pre-trained multi-speaker model with the following steps:

1. Set the noise control tag to 'noise' for adaptation data;
2. Remove the domain classifier loss since we assume the latent z is noise-independent;
3. Choose a speaker in the training set whose timbre is the most similar to target speaker, and share its speaker embedding to the target speaker [6];
4. Fine-tune the whole model until convergence;
5. Set the noise control tag to 'clean' and choose the above speaker embedding to generate clean speech of target speaker.

2.2. One-shot speaker encoding with DAT

As discussed above, the proposed few-shots speaker adaptation method requires a few adaptation samples with transcription to fine-tune the model. We further propose a robust one-shot speaker adaptation method for noisy target speaker speech without transcription. To this end, we firstly build an individual text-independent speaker discriminative model trained on speaker verification dataset [4, 5, 24]. The model adopts time delay neural network (TDNN) [24] to extract the speaker representation (so-called *x-vector*) in the latent space. With the speaker recognition model, we can easily obtain the continuous speaker embedding s for both training and adaptation samples.

Different from the above few-shots speaker adaptation, the noisy target speaker's audio is only used to extract speaker embedding. The domain adversarial training module is the same as previous few-shots adaptation. Since the continuous speaker representation s is obtained from noisy speech, it also inevitably contains noise information. In order to avoid introducing noise into c_t, we only inject s in $h(\cdot)$ rather than in both of $g(\cdot)$ and $h(\cdot)$. We train the multi-speaker model with the same objective function in Eq (8).

In order to process domain adversarial training, we still need to augment the training set. Considering a triple of training samples $< aud_{ref}, text, aud_{tgt} >$, we augment the aud_{ref} with random noise and get aud_{ref}^n. Therefore, the processed training set is doubled, consisting of two types of samples (aud_{ref} and aud_{ref}^n) with the same number. And then we apply the same training process as speaker adaptation.

During adaptation, we only need one noisy sample to extract speaker representation s to generate target clean voice. As for a few adaptation samples, we can treat the mean of speaker representations of all sentences as s to control generation, which may be more stable than the s from a single sentence.

3. Experiments and Results

3.1. Basic setups

In our experiments, we use a multi-speaker Mandarin corpus, referred as MULTI-SPK, and a noise corpus from CHiME-4 challenge [25] to simulate noisy speech. The MULTI-SPK dataset consists of 100 different speakers in different ages and genders and each speaker has 500 utterances. The CHiME-4 corpus contains about 8.5 hours of four large categories of background noises. We augment the training set at random signal-to-noise ratio (SNR) ranging from 5 to 25db. We reserve two males (indexed as 001 and 045) and two females (indexed as 077 and 093) as our target unseen speakers for voice cloning experiments. For each target speaker, we select 50 sentences (3-4 minutes of speech) as test samples. The clean test sets for two female and two male speakers are referred as F-C an M-C, respectively. In order to evaluate the performance of noisy target audio, we also add random background noise to F-C and M-C in the way with the training set, resulting in F-N and M-N respectively. As for the de-noising baseline with external speech enhancement module, we use the state-of-the-art speech enhancement model named DCUnet [26] to de-noise F-N and M-N. The internal DCUnet model is trained using over 2000 hours of training data with strong and stable de-noising capacity. The de-noised test sets are referred as F-D and M-D. For clarity, the different parts of test sets are shown in Figure 3.

To evaluate speaker similarity, we extract x-vectors from the synthesized speech and then measure the cosine distance with the x-vector extracted from original speech of the target speaker. We also evaluate speaker similarity and naturalness using subjective mean score option (MOS) tests, where about 20 listeners examining the testing clips. As for objective evaluation, we measure the mel-cepstral distortion (MCD) between generated and real samples after dynamic time warping.

3.2. Model details

All of our models take phoneme-level linguistic features, including phoneme, word boundary, prosody boundary and syllable tone, as input of the CBHG-based encoder [22]. The GMM-based monotonic attention mechanism is employed to align phoneme-level linguistic representations and frame-level acoustic features during training [23]. The architecture of decoder is similar with Tacotron2 [1], and the number of units of additional GRU after pre-net is 256 for latent feature learning. For speaker representation, we adopt straight-forward learnable embedding table for few-shots adaptation, where the dimension of speaker embedding is 256. As for one-shot adaptation, the dimension of x-vector is 512. We concatenate the x-vector with the above latent features.

For the vocoder, we train gender dependent universal mel-LPCNet, which extends LPCNet [27] with mel-spectrogram, using original MULTI-SPK dataset to convert mel-spectrogram to waveform. The audio samples will be available online[1].

3.3. Evaluation on few-shots speaker adaptation

We firstly train a standard multi-speaker system using original training set without domain adversarial module as our baseline,

[1] https://npujcong.github.io/voice_cloning

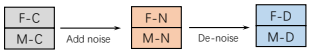


Figure 3: *Different parts of test set. F and M indicate female and male, C and N refer to clean and noisy audio, and D indicates de-noised speech.*

referred as BASE. As for the robust few-shots speaker adaptation, we propose to conduct adversarial training, referred as DAT. For the proposed model, we use noise tag (0/1) to control the acoustic condition and the λ in loss function is set to 0.1. At adaptation time, we adapt the baseline model BASE and proposed model DAT with different test set, where the batch size is set to 8 and initial learning rate is set to 10^{-5}.

Results in terms of various metrics are shown in Table 1 (upper part). For the adaptation with clean target data (F-C, M-C), although we only use half clean speakers in the training set to train the proposed model, the naturalness and similarity of synthesized speech of baseline and proposed model are similar. As for the noisy adaptation data, the BASE model even cannot learn a stable alignment during model fine-tuning, resulting in speech generation failures, i.e. incomplete, mis-pronounced and non-stoping utterances. However, the proposed DAT model still works well to generate target speaker's clean voice, whose performance is close to those samples on clean data in both naturalness and similarity. This result indicates that the proposed approach has ability to produce stable clean target voice under few-shots speaker adaptation scene. We also de-noise the noisy target data to conduct speaker adaptation on the BASE model, but the result indicates that the adaptation with de-noised data suffers from the speech distortion problem, where the MCD is much higher than that of the proposed model. Besides, the similarity is also worse than the proposed model.

Table 1: *The results of speaker adaptation and encoding on different test sets and models. "×" means the model is failed to conduct adaptation. N-MOS and S-MOS denote MOS on naturalness and similarity, and SIM-COS is cosine similarity.*

Few-shots Speaker Adaptation					
TEST SET	MODEL	MCD	N-MOS	SIM-COS	S-MOS
F-C	BASE	3.77	3.42	0.96	3.64
F-C	DAT	3.87	3.42	0.96	3.65
F-N	BASE	×	×	×	×
F-N	DAT	4.18	3.36	0.95	3.65
F-D	BASE	4.72	3.30	0.86	3.45
M-C	BASE	3.66	3.56	0.96	3.64
M-C	DAT	3.95	3.54	0.95	3.71
M-N	BASE	×	×	×	×
M-N	DAT	4.20	3.53	0.93	3.72
M-D	BASE	4.56	3.51	0.91	3.70
One-shot Speaker Encoding					
F-C	BASE	4.30	3.39	0.91	3.51
F-C	DAT	4.31	3.5	0.92	3.54
F-N	BASE	×	×	×	×
F-N	DAT	4.35	3.41	0.92	3.50
F-D	BASE	5.32	3.32	0.89	3.4
M-C	BASE	4.62	3.67	0.85	3.16
M-C	DAT	4.58	3.63	0.88	3.26
M-N	BASE	×	×	×	×
M-N	DAT	4.55	3.67	0.88	3.34
M-D	BASE	4.84	3.57	0.76	2.96

3.4. Evaluation on one-shot speaker encoding

We train an independent x-vector model using internal 3000 hours speaker verification dataset over 2000 speakers. The x-vector is projected to 256 dimension and then used as condition on the TTS model. We train a multi-speaker model with speaker encoder using original dataset as our baseline, referred as BASE and proposed model with DAT using augmented dataset, referred as DAT. During adaptation, we compute mean of x-vectors extracted from 5 sentences randomly selected from test set. Results are shown in the lower half of Table 1.

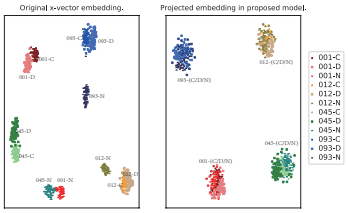


Figure 4: *Visualization of utterance embedding from different speakers at clean and noise conditions. C, N and D stand for clean, noisy and de-noised audio.*

For the clean target speakers, we also find there are no significant difference between proposed DAT model and the BASE model. But for noisy data, it is hard to evaluate the performance of the BASE model since it always crashes with the corresponding noisy x-vectors. As for our proposed model, whether the target audio is clean or noisy, it can produce stable and clean synthesized speech of the target speaker. Similar to the speaker adaptation methods for few-shots adaptation, even we de-noise the noisy target audio to extract x-vector, the naturalness and similarity of generated speech is much worse than the proposed DAT method. When we compare speaker adaptation and speaker encoding, we find that speaker adaptation can produce apparently higher speaker similarity samples than speaker encoding. This observation is consistent with [2] as it's still challenging catching speaker's identity in fine-details using just one shot from the speaker; it is even more challenging using one noisy sample.

To evaluate the effectiveness of proposed model, we also analyze the projected speaker embedding of our proposed model with the original x-vectors from target speakers using t-SNE [28], as shown in Fig. 4. For x-vectors from target speech, whereas the x-vectors have clear distances between speakers, the speaker representations of noisy and clean samples for the same speaker are usually divided into two clusters. It means that the speaker representation is easily affected by noise interferences, which will directly cause the speaker similarity problem in one-shot speaker adaptation. As for the proposed speaker embedding with adversarial training, we find there is no obvious distance between noisy and clean samples from the same speaker. It indicates that the proposed model successfully disentangles the noise condition from the speaker embedding, which alleviates the negative effects from noise in target speech.

4. Conclusions and Future Work

The paper proposes to use domain adversarial training for data efficient voice cloning from noisy target speaker samples. Results indicate that in both few-shots speaker adaptation and one-shot speaker encoding, the proposed approaches can produce clean target speaker's voice with both reasonable naturalness and similarity. Future work will try to handle the more complicated acoustic condition scenarios, e.g., room reverberations.

5. References

[1] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[2] S. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou, "Neural voice cloning with a few samples," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 019–10 029.

[3] S. Yang, Z. Wu, and L. Xie, "On the training of dnn-based average voice model for speech synthesis," in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2016, pp. 1–6.

[4] E. Cooper, C.-I. Lai, Y. Yasuda, F. Fang, X. Wang, N. Chen, and J. Yamagishi, "Zero-shot multi-speaker text-to-speech with state-of-the-art neural speaker embeddings."

[5] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. L. Moreno, Y. Wu *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Advances in neural information processing systems*, 2018, pp. 4480–4490.

[6] Y. Chen, Y. Assael, B. Shillingford, D. Budden, S. Reed, H. Zen, Q. Wang, L. C. Cobo, A. Trask, B. Laurie *et al.*, "Sample efficient adaptive text-to-speech," *arXiv preprint arXiv:1809.10460*, 2018.

[7] Y. Taigman, L. Wolf, A. Polyak, and E. Nachmani, "Voiceloop: Voice fitting and synthesis via a phonological loop," *arXiv preprint arXiv:1707.06588*, 2017.

[8] Y. Huang, L. He, W. Wei, W. Gale, J. Li, and Y. Gong, "Using personalized speech synthesis and neural language generator for rapid speaker adaptation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7399–7403.

[9] H. B. Moss, V. Aggarwal, N. Prateek, J. González, and R. Barra-Chicote, "Boffin tts: Few-shot speaker adaptation by bayesian optimization," *arXiv preprint arXiv:2002.01953*, 2020.

[10] S. Yang, Y. Wang, and L. Xie, "Adversarial feature learning and unsupervised clustering based speech synthesis for found data with acoustic and textual noise," *arXiv preprint arXiv:2004.13595*, 2020.

[11] C. Valentini-Botinhao and J. Yamagishi, "Speech enhancement of noisy and reverberant speech for text-to-speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 8, pp. 1420–1433, 2018.

[12] Q. Hu, E. Marchi, D. Winarsky, Y. Stylianou, D. Naik, and S. Kajarekar, "Neural text-to-speech adaptation from low quality public recordings," in *Speech Synthesis Workshop*, vol. 10, 2019.

[13] N. Gurunath, S. K. Rallabandi, and A. Black, "Disentangling speech and non-speech components for building robust acoustic models from found data," *arXiv preprint arXiv:1909.11727*, 2019.

[14] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[15] W.-N. Hsu, Y. Zhang, R. J. Weiss, Y.-A. Chung, Y. Wang, Y. Wu, and J. Glass, "Disentangling correlated speaker and noise for speech synthesis via data augmentation and adversarial factorization," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5901–5905.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[17] S. Sun, B. Zhang, L. Xie, and Y. Zhang, "An unsupervised deep domain adaptation approach for robust speech recognition," *Neurocomputing*, vol. 257, pp. 79–87, 2017.

[18] Z. Meng, Z. Chen, V. Mazalov, J. Li, and Y. Gong, "Unsupervised adaptation with domain separation networks for robust speech recognition," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 214–221.

[19] S. Sun, C.-F. Yeh, M.-Y. Hwang, M. Ostendorf, and L. Xie, "Domain adversarial training for accented speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4854–4858.

[20] Q. Wang, W. Rao, S. Sun, L. Xie, E. S. Chng, and H. Li, "Unsupervised domain adaptation via domain adversarial training for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4889–4893.

[21] Y. Tu, M.-W. Mak, and J.-T. Chien, "Variational domain adversarial learning for speaker verification," *Proc. Interspeech 2019*, pp. 4315–4319, 2019.

[22] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[23] E. Battenberg, R. Skerry-Ryan, S. Mariooryad, D. Stanton, D. Kao, M. Shannon, and T. Bagby, "Location-relative attention mechanisms for robust long-form speech synthesis," *arXiv preprint arXiv:1910.10288*, 2019.

[24] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[25] E. Vincent, S. Watanabe, A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech and Language*, vol. 46, pp. 535–557, 2017.

[26] H.-S. Choi, J.-H. Kim, J. Huh, A. Kim, J.-W. Ha, and K. Lee, "Phase-aware speech enhancement with deep complex u-net," *arXiv preprint arXiv:1903.03107*, 2019.

[27] J.-M. Valin and J. Skoglund, "Lpcnet: Improving neural speech synthesis through linear prediction," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5891–5895.

[28] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

Delivery Scope: A New Way of Restaurant Retrieval For On-demand Food Delivery Service

Xuetao Ding, Runfeng Zhang, Zhen Mao, Ke Xing, Fangxiao Du,
Xingyu Liu, Guoxing Wei, Feifan Yin, Renqing He, Zhizhao Sun
Meituan-Dianping Group
Beijing, P.R.China
{dingxuetao,zhangrunfeng,maozhen,xingke,dufangxiao,liuxingyu,weiguoxing,yinfeifan,herenqing,sunzhizhao}@meituan.com

ABSTRACT

Recently on-demand food delivery service has become very popular in China. More than 30 million orders are placed by eaters of Meituan-Dianping everyday. Delicacies are delivered to eaters in 30 minutes on average. To fully leverage the ability of our couriers and restaurants, delivery scope is proposed as an infrastructure product for on-demand food delivery area. A delivery scope based retrieval system is designed and built on our platform. In order to draw suitable delivery scopes for millions of restaurant partners, we propose a pioneering delivery scope generation framework. In our framework, a single delivery scope generation algorithm is proposed by using spatial computational techniques and data mining techniques. Moreover, a scope scoring algorithm and decision algorithm are proposed by utilizing machine learning models and combinatorial optimization techniques. Specifically, we propose a novel delivery scope sample generation method and use the scope related features to estimate order numbers and average delivery time in a period of time for each delivery scope. Then we formalize the candidate scopes selection process as a binary integer programming problem. Both branch&bound algorithm and a heuristic search algorithm are integrated in our system. Results of online experiments show that scopes generated by our new algorithm significantly outperform manual generated ones. Our algorithm brings more orders without hurt of users' experience. After deployed online, our system has saved thousands of hours for operation staff, and it is considered to be one of the most useful operation tools to balance demand of eaters and supply of restaurants and couriers.

KEYWORDS

location-based retrieval, on-demand food delivery, spatial computation, machine learning, combinational optimization

ACM Reference Format:
Xuetao Ding, Runfeng Zhang, Zhen Mao, Ke Xing, Fangxiao Du, and Xingyu Liu, Guoxing Wei, Feifan Yin, Renqing He, Zhizhao Sun. 2020. Delivery Scope: A New Way of Restaurant Retrieval For On-demand Food Delivery Service. In *26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394486.3403353

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403353

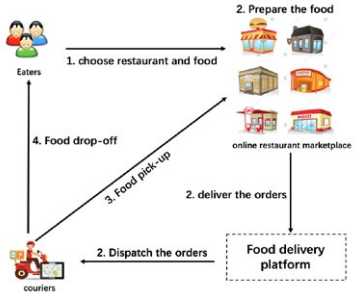


Figure 1: Illustration of 4 phases of on-demand food delivery service. (1) Eaters choose the restaurant and food from the marketplace powered by food delivery service; (2) As soon as order is placed, three things happened almost at the same time, a. restaurant start to prepare the food, b. restaurant online marketplace deliver the order to the food delivery platform, c. food delivery platform dispatch the order to the appropriate courier; (3) Couriers pick up the food from restaurant following the instructions from food delivery platform; (4) Finally, couriers deliver the food to the hands of eaters.

1 INTRODUCTION

Recently on-demand food delivery service has become very popular in the world, especially in China. As shown in Figure 1, eaters could explore the online restaurant marketplace, choose restaurants and order their favorite foods or drinks. After about 30 minutes, eaters would receive them. Everyday more than 30 million orders are placed on Meituan-Dianping platform, one of the world's largest on-demand delivery service provider, and about 1 million couriers work for these orders' delivery. Due to fast growing demand of eaters and limited number of couriers, the way to fully exploit our delivery ability becomes a core factor for our on-demand delivery service. An effective dispatch system to match orders and couriers is indispensable [8]. Besides, a new mechanism of restaurant retrieval, the **delivery scope**, is proposed in the area of on-demand delivery service, which we mainly discuss in this paper.

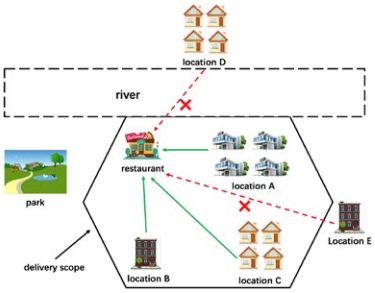


Figure 2: Illustration of a delivery scope. In this figure, the restaurant's delivery scope is the irregular polygon with black solid border. Only eaters inside the polygon could browse the restaurant's menu on our platform and place an order, like location A/B/C. It is difficult for couriers to deliver food across the river or too far away from the restaurant. Thus, location D and E are excluded from the polygon when drawing the delivery scope.

In the area of on-demand delivery service, we must ensure that orders can be delivered to eaters in 30 minutes on average. Due to limited supply of our couriers, it is unrealistic that eaters could order their delicacies from every restaurant in their cities. On our platform, we allocate every restaurant a spatial polygon as the service area. Only eaters whose shipping addresses locate in the service area can order foods and drinks from this restaurant. We define the spatial polygon as **delivery scope**, which is illustrated in Figure 2.

Since there are millions of restaurants on our platform, it means there are at least millions of corresponding delivery scopes. When one eater visits our application, our system should satisfy: 1) all restaurants whose delivery scopes contain the eater's shipping address should be retrieved; 2) one retrieval process should be completed in tens of milliseconds, the same scale of traditional content retrieval systems. To achieve these, a R-tree based index system is designed and implemented. The high performance polygon-based retrieval system is part of the core of this new retrieval paradigm, although we will not focus on it in our paper.

To fully exploit our delivery ability, how to generate delivery scopes becomes very important. In order to generate suitable delivery scopes, both eaters' preferences and couriers' delivery efficiency should be considered. Specifically, delivery scope generation process should follow three principles: 1) restaurant's delivery scope should cover high-demand area; 2) delivery scope should be smaller compared with other restaurants' if its supply ability is relatively lower; 3) delivery scope should not cover faraway locations. Besides, we should gaurantee eaters from a city block could browse same restaurants on our platform. And our restaurant partners and delivery partners could see their own delivery scopes from the management system, they also pay close attention to the shapes. Thus, we make the following rules for our scope generation process: 1) border of a scope should be along the roads; 2) border of a scope should not cut city blocks; 3) delivery scope should not cover or traverse delivery-hard locations, such as seas, rivers, hills, railways etc.

Before our delivery scopes generation system deployed online, our operation staff in different cities drew all delivery scope by hands. It is a painstaking job and unrealistic for them to draw scopes balancing demand of our eaters and supply of restaurants and our couriers. We propose a delivery scope generation algorithm and build an automatic generation system online. Our system frees operation staff from the boring scope drawing jobs. Moreover, the algorithm generated delivery scopes outperform manual ones. Our large-scale online experimental results show that our algorithm significantly increases the number of orders without delivery experience loss.

Specifically, we exploit spatial data mining techniques, machine learning techniques and combinational optimization techniques to generate delivery scopes for millions of restaurants. A unified framework, which we call **delivery scope generation algorithm**, is proposed. First, we generate candidate scopes for target restaurants. Spatial data mining techniques are exploited to ensure that algorithm-generated scopes obey three following rules: 1) candidate scopes should meet distance constraint; 2) scopes should meet requirements for consistent users' experience; 3) scopes should cover more high-demand locations under the constraints of 1) and 2). Second, we utilize regression models to estimate the order number and average order delivery time if a restaurant adopts the scope. Last, we formalize delivery scope allocation process as a binary integer programming problem.

Our contributions are listed as follows: 1) a different retrieval paradigm to utilize the delivery scope for location-based retrieval system is proposed, and we suggest that any similar online location-based applications could utilize this paradigm to improve their service; 2) we proposed a pioneering delivery scope generation framework, results of online experiments demonstrate that our algorithm not only saves manual delivery scope maintenance time, but also brings better purchase and delivery-service experience; 3) we design and build a delivery scope generation system, which has shown success on one of the world's largest on-demand food delivery platform as a pioneering industry-level practice.

2 RESTAURANT RETRIEVAL SYSTEM ARCHITECTURE

Our restaurant retrieval system are demonstrated in Figure 3. The system is designed by two parts. We will introduce each part in the following paragraphs.

Delivery Scope Indexer: The indexer is designed and built for two main functions. First, it supports millions of visits in milliseconds' response time to retrieve the restaurant list via giving the coordinate of eater's shipping address. Second, it supports hundreds of thousands delivery scope update in seconds. The core data structure of the indexer is an improved R-tree. In this paper, details of the indexer will not be mentioned.

Delivery Scope Generation System: This system implements the delivery scope generation framework we proposed. Basically,

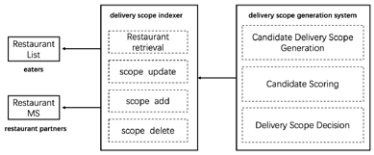


Figure 3: Delivery scope system architecture, it includes two parts: 1) delivery scope index service to update/add/delete restaurant delivery scope and provide the retrieval API for eaters side application and restaurant partners side application; 2) delivery scope generation system.

our system would generate candidate scopes for a group of restaurants using our single delivery scopes drawing pipeline and each candidate would be scored. Then our system will select one final suitable delivery scope for every restaurant in the group to achieve a global optimum. In the following sections, we will focus on these details.

3 DELIVERY SCOPE GENERATION ALGORITHM

3.1 Framework of Delivery Scope Generation

The goal of our algorithm is to generate a reasonable delivery scope for every restaurant in target **station**, a city block of which restaurants' products are deliveried by a specific group of couriers basically. The number of restaurants in one station lies between 200 and 1,000 in our system. The scope generation framework comprises three main stages: 1) candidate delivery scopes generation, 2) candidate delivery scopes scoring and 3) combinatorial optimization of delivery scopes.

In the first stage, several candidate delivery scopes are generated for every restaurant in the target station. In order to generate a reasonable and elegant scope, we propose a novel single scope generation algorithm, as described in Section 3.2.

Then, in the second stage, machine learning models are exploited to get predictions, or scores, about these scopes. Candidate delivery scopes of each restaurants are spatial polygon with different shapes and sizes. We extract a series of features that could capture characteristics of these scopes. Details of the scoring process can be found in Section 3.3.

In the last stage, the combinatorial optimization process selects one delivery scope for each restaurant from its candidates. The target of this stage is to differentiate restaurants in the station. Generally, it is better to allocate larger delivery scopes to restaurants with higher eater conversion rate, greater products' preparation efficiency and more convenient nearby transportation condition. We formalize this as a **binary integer programming** problem (abbreviated as **BIP** in the following sections) and then use the **branch and bound** algorithm [2, 5] or a heuristic search algorithm to solve it.

3.2 How to Draw Single Scope

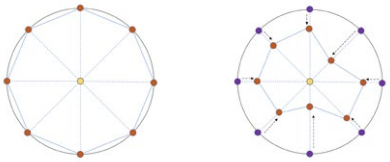
The foundation of our delivery scope generation system is a single delivery scope drawing algorithm. In this algorithm, a single delivery scope would be generated given a target radius and a location point as inputs. Navigation distances between the location point and points on the boundary of the scope are supposed to be (approximately) equal to the input radius. When fed multiple input radii, the algorithm could generate scopes in different sizes and shapes. And, naturally, these scopes cover different numbers of potential eaters. We treat these scopes as candidate scopes which would be used in subsequent scope decision procedures. Besides, our restaurant partners expect the delivery scopes could cover high-demand area as much as possible and that the edges of the delivery scopes could be along city roads. Our single delivery drawing algorithm also takes these business requirements into consideration.

In the system, delivery scope is represented by a polygon with multiple vertices, among which two adjacent vertices are connected by a line segment. The single scope generation algorithm comprises four modules: initialization, business requirements fulfillment, compression and covering high-demand areas. Each module takes the tentative scope processed by the former one as input. In this section, we dive into some details of these modules.

3.2.1 Initialization. In the begining of this module, the algorithm uses the restaurant's location point as the center to draw a circle with input radius. The circle is represented by a fixed number of points on the border at equal intervals. However, navigation distances from these points to the center may be much larger than their straight-line distances. To solve this, our algorithm searches for new points, whose navigation distances to the center are approximately equal to our target radius, on segments between these original points and the center. We adopt a binary search way here. In each step, one boundary point would move towards the center with the length of $\max(\min(naviDist - targetRadius, straightLineDist)/2, minStep)$. Here, $naviDist$ and $straightLineDist$ are the navigation distance and straight-line distance between the boundary point and the center. $targetRadius$ is our input target radius, and $minStep$, as a hyper-parameter, is used to restrict the minimal move length in the search process.

3.2.2 Scope Shape Optimization for Experience. To guarantee better experience of users, restaurants and delivery partners, boundaries of our delivery scopes should be along roads, which is crucial for our platform. A delivery scope whose boundary goes across some buildings or blocks could bring different user experience for eaters in a same building. To be specific, there might be a case that some eaters in the building could place orders on our platform, but some others could not. In order to get boundaries along the road, our algorithm substitutes navigation routes for straight line segments between adjacent boundary points. New boundary points are added in this process. Besides, every station keeps a maximal shipping scope, which marks all places couriers of this station could go. Generated delivery scopes would be intersected with the maximal shipping scope to exclude some improper area.

In order to get elegant delivery scopes, algorithm removes stubs brought in the navigation route process on boundaries: we allocate



(a) Initialized Circle (b) Navigation Distance Modification

Figure 4: Illustration of Scope Initialization: (a) the yellow point marks the restaurant location point, the orange points mark boundary points sampled from initial circle. (b) In the navigation distance modification process, boundary points would move towards the center to meet the navigation distance requirement.

boundary points indexes in order and record GeoHashes [1] that those points belong to. Then, for each of those GeoHashes, our algorithm finds the minimal-index point and the maximal-index one. If most of the intermediate points locate in GeoHashes far from the current GeoHash, the algorithm would treat the route between these two points as a stub, and remove it by connecting the two points directly.

3.2.3 Scope Compression. Plenty of boundary points have been brought in the navigation route process, and in the scope compression module, our algorithm removes redundant points for more efficient data computation and storage. We compute the angle of each point between its subsequent point and its former point. If the angle is close to 180° enough, the point would be removed. An alternate strategy evaluates each points based on both angle and distance between adjacent points, points with higher scores would be retained. Our system also integrates Douglas-Peucker algorithm [15] in the compression module. Comparison between these compression methods can be found in Table 3.

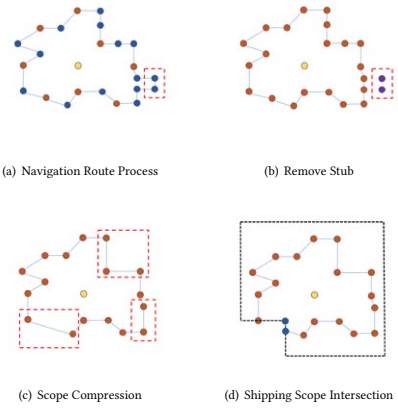
3.2.4 Covering High-Demand Areas. This module modifies delivery scope to cover more high demand areas. In the initialization part mentioned before, the algorithm draws a circle centered at the location point of our target restaurant. Actually, in our system, the center is a deviated one, which is computed based on the location point and the distribution of orders in the area nearby. Beyond that, some high-demand areas are merged directly into the delivery scope. Here, these high-demand areas are the result of GeoHashes clustering. In our system, both K-means and DBSCAN[7] algorithm are reasonable options and adopted. At last, these potential areas are merged to form a new polygon, the final delivery scope, while keeping the restriction of navigation distance.

3.3 Single Scope Scoring

After candidate delivery scopes with different radii for a single restaurant generated, our algorithm measures their scores. Here, the scores include the number of orders and average delivery time in a period of time, which would be used in following optimization process.

We use the estimation of the number of orders to illustrate the process. Specifically, if a target scope to be scored is inside the restaurant's current delivery scope, what needs to do is to count

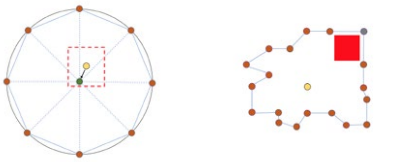
[1] GeoHash is a public domain geocode system and encodes a geographic location into a short string of letters and digits.



(a) Navigation Route Process (b) Remove Stub

(c) Scope Compression (d) Shipping Scope Intersection

Figure 5: Illustration of Scope Shape Optimization and Scope Compression: (a) The boundaries of delivery scopes processed by Navigation Route Process are along roads. (b) A tiny stub in the red bound box is removed. (c) Some redundant points are removed in Scope Compression Process. (d) Delivery scopes are intersected with the maximal shipping scope to remove some improper area.



(a) Center Point Shift (b) Merge High-Demand Areas

Figure 6: Illustration of Covering High-Demand Areas: (a) Yellow point marks a restaurant's location point and the green one marks the deviated scope center computed according to the distribution of orders in the red bouding box. (b) The red block, representing high-demand area, is appended to the original delivery scope.

the number of orders from historical data. However, for a candidate scope larger than the current one, the scoring process would contain two stages: 1) couting the number of orders inside the intersection between the larger scope and the current scope. In Figure 7, the intersection part corresponds to the yellow area. 2) For the area outside the current scope, as shown in pink belt in the figure, regression models would be used. The two scores would be summed to get the final estimation of the number of orders. In our prediction part, we construct samples in the following way: for every restaurant on our platform, we shrink its current delivery scope into smaller ones, which we call "virtual scopes". Then we might treat the virtual scope as the restaurant's real scope and treat the number of orders placed by eaters in the outside belt as the label. By doing so, we get many samples from one restaurant. Among all our features, some are extracted based on the spatial information about virtual scopes and outside belts. We compare several common regression models, such as ridge regression, random forest[11], and XGBoost[4]. Details about metrics of these models could be found in Section 4.2.

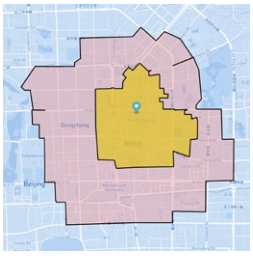


Figure 7: Single Scope Scoring: In this picture, The yellow scope is the current delivery scope of the restaurant in blue marker. The larger one is the scope to be scored. We divide the task into two parts: computing the score in the yellow scope from historical data and predicting the score about the pink belt. Two parts would be conflated into the final score for the targe larger scope.

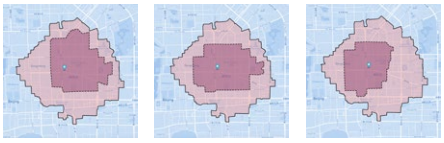


Figure 8: Given the current delivery scope of a restaurant, we shrink the scope border to generate smaller virtual scopes. Here, the three purple scopes with dash line are virtual scopes generated from a same restaurant. They correspond to three samples in the dataset. Labels are the numbers of orders placed by eaters in outside pink belt areas.

The average delivery time is predicted in the same way. Some time related features are designed to feed in our model, like our couriers' delivery speed and average food preparation time for the targe restaurant.

3.4 Delivery Scope Optimization

By adopting methods in Section 3.2 and 3.3 we could get candidate delivery scopes for restaurants and some corresponding scores, like the predicted numbers of orders and the predicted average delivery time in a period of time. Now the problem is how to utilize all this information to allocate each restaurant a reasonable delivery scope in order to achieve a global optimal (or approximate) **GMV** (**Global Merchandise Volume**) or number of orders for a station, and at the same time, to guarantee eaters' experience.

We formalize the problem as: for a target station, there are N restaurants. Each restaurant has M candidate delivery scopes with different sizes. $S_{n,m}$ is used to indicate the m-th candidate delivery scope of the n-th restaurant. $O_{n,m}$ and $T_{n,m}$ are the predicted order num and the average order delivery time if the n-th restaurant adopts candidate scope $S_{n,m}$. What calls for special attention is that we draw candidate scopes of one restaurant from small target radii to large ones in order, so usually scope $S_{n,m+1}$ could cover scope $S_{n,m}$. And when we predict the $O_{n,m}$ and $T_{n,m}$, we also force the model to predict large scores ($O_{n,m}$, $T_{n,m}$) for large scopes. P_n, the average production price of the n-th restaurant, is computed from historical data. In our system, we treat the average delivery time in a station as the very indicator of eaters' experience, and $\bar{T}$ is our target upper bound. We use the binary variable $C_{n,m}$ to indicate whether the m-th candidate scope should be the delivery scope of n-th restaurant, then our problem is converted into a BIP (Binary Integer Programming) problem:

$$\max \sum_{n=1}^{N}\sum_{m=1}^{M} O_{n,m}C_{n,m}P_n \tag{1}$$

$$s.t. \sum_{n=1}^{N}\sum_{m=1}^{M} T_{n,m}O_{n,m}C_{n,m} \le \bar{T}\sum_{n=1}^{N}\sum_{m=1}^{M} O_{n,m}C_{n,m} \tag{2}$$

$$\sum_{m=1}^{M} C_{n,m} = 1,\ n \in \{1,\dots,N\} \tag{3}$$

$$C_{n,m} \in \{0,1\},\ n \in \{1,\dots,N\},\ m \in \{1,\dots,M\} \tag{4}$$

The target of the BIP problem defined above is to maximize the GMV of a station, and we could change the target into maximization of orders by setting all P_n to be a same constant. BIP problem [3] is a typical NP-Hard problem. Several kinds of heuristic algorithms, such as hill climbing algorithm [13] and simulated annealing algorithm [9], could get an approximate solution efficiently. In our system, we usually set M to be 10 (or less than 10) and the number of restaurants is less than 1k in most of our stations. So the total number of variables $C_{n,m}$ is on the order of thousands. The exact algorithm, branch and bound algorithm, could handle the BIP problem with such scale in seconds on our server. A branch and bound solver is integrated in our system, and besides, we propose a heuristic algorithm to solve our BIP problem when we prefer to select a final delivery scope for each restaurant from more than 1k

candidate scopes, in which case it would take serveral minutes to get the exact solution by using branch and bound algorithm directly on our server. Algorithm 1 displays details of our heuristic algorithm. The approximate solution could also be used as the initial feasible solution for the branch and bound algorithm to accelerate the solving process. In Figure 9, we show the framework of our delivery scope generation system, which includes the three parts mentioned before.

Algorithm 1: Delivery Scope BIP Heuristic Search

Data: order prediction $O_{n,m}$, delivery time prediction $T_{n,m}$, restaurant's average production price P_n, number of restaurants N, number of candidate scopes M, target upper bound of average delivery time $\bar{T}$;
Result: approximate solution of the BIP problem $C_{n,m}$

Function heuristicSearch($O, P, T, \bar{T}, N, M$):
  $priority\_queue \leftarrow \emptyset$;
  for $n = 1$ *to* N **do**
    $scope\_idx[n] \leftarrow 1$;
    heapPush($priority\_queue, O, P, T, n, 1, M$)
  end
  $bound \leftarrow$ getBound($O, T, \bar{T}, scope\_idx, N, M$);
  while $priority\_queue$ not empty **do**
    $(priority, n) \leftarrow$ heapPop($priority\_queue$) ;
    $m \leftarrow scope\_idx[n]$;
    $delta \leftarrow O_{n,m+1}(T_{n,m+1} - \bar{T}) - O_{n,m}(T_{n,m} - \bar{T})$;
    $new\_bound \leftarrow bound + delta$;
    if $new\_bound \leq 0$ **then**
      $scope\_idx[n] \leftarrow m + 1$;
      heapPush($priority\_queue, O, P, T, n, m + 1, M$);
      $bound \leftarrow new\_bound$;
    end
  end
  for $n = 1$ *to* N, $m = 1$ *to* M **do**
    $C_{n,m} \leftarrow \mathbb{1}\{m = scope\_idx[n]\}$
  end
  return C
Function getBound($O, T, \bar{T}, scope\_idx, N, M$):
  return $\sum_{n=1}^{N} \sum_{m=1}^{M} (T_{n,m} - \bar{T}) O_{n,m} \mathbb{1}\{m = scope\_idx[n]\}$
Function heapPush($priority\_queue, O, P, T, n, m, M$):
  if $m+1 \leq M$ **then**
    $priority \leftarrow \frac{P_n(O_{n,m+1} - O_{n,m})}{O_{n,m+1}T_{n,m+1} - O_{n,m}T_{n,m}}$;
    push tuple $(priority, n)$ into priority_queue ;
  end

3.5 Computational Complexity

In the single delivery scope drawing procedure, we assign K as the number of sampled points in initial circle. During Navigation Distance Modification, each point steps back to proper position with navigation distance query. Every navigation route query takes the time of T_{navi}, which could be considered as a constant. Since the maximum binary search range is fixed, it takes $O(K)$ to process all points. Navigation Route Process also needs $O(K)$ to query navigation route between adjacent points. Both Remove Stub and Compression take $O(K)$ complexity. So it would be considered to take $O(K)$ to generate one scope. Using the symbols in Section 3.4, it would take $O(KNM)$ to get all candidate delivery scopes for a station. Our algorithm adopts XgBoost as the scope scoring method. The tree depth, the number of stack iteration and the number of features are fixed before the training process. That means it needs to take $O(NM)$ steps to get all scores for these scopes.

The scope allocation stage is a solving process of a BIP problem. To find the exact solution is a NP-Hard problem. Branch and bound algorithm could get an exact solution, but the complexity is relatively difficult to estimate due to its data-dependent procedures inside. Some details about the complexity of branch and bound algorithm could be found in [10]. Our heuristic search algorithm could get an approximate solution in at most NM steps, so the computational complexity of this algorithm is $O(NM)$.

Overall, the total computational complexity of our delivery scope generation algorithm is $O(KNM)$ (by using our heuristic solver). Details about the efficiency of our real-world application can be found in Section 4.4.

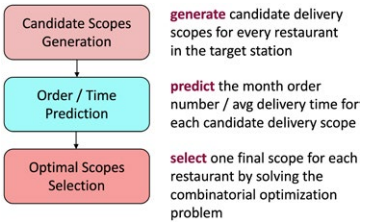


Figure 9: Delivery Scope System Framework.

4 EXPERIMENTS

In this section, we introduce our experiments on Meituan-Dianping food delivery platform. First, we discuss some details of our online evaluation, including experiment settings, results and the reason why our algorithm outperforms humans. Second, we compare scope estimation models adopted in our algorithm. Third, we show the performance of single scope generation algorithm. And lastly, we briefly introduce the efficiency improvement after our algorithm deployed online.

4.1 Online Evaluation

4.1.1 Experiment Settings. We conducted our A/B Test experiment on the 1448 stations of 133 cities in China, which included more than 160,000 restaurants. Specifically, stations in each city were divided into two groups: the experimental group and the control group randomly. Then we put all stations in experimental groups of all cities together into the final experimental group, and all others were combined into the final control group. For restaurants in the experimental group, we substituted algorithm generated delivery

scopes for the old delivery scopes at the launch time. For restaurants in the control group, we kept their old delivery scopes unchanged. The order information data of restaurants two weeks before and after the launch time in both experimental group and control group were collected, which were used to verify the effectiveness of our algorithm.

We record metric for results two weeks before the launch time as Met_{bef} and record metric for results two weeks after the launch time as Met_{aft}. Then we define $\mathcal{R}_{Met}$ as the changing ratio between Met_{aft} and Met_{bef} as follows:

$$\mathcal{R}_{Met} = (Met_{aft} - Met_{bef})/Met_{bef} \tag{5}$$

For metrics which are percentage number themselves, we define the absolute difference Δ_{Met}:

$$\Delta_{Met} = Met_{aft} - Met_{bef} \tag{6}$$

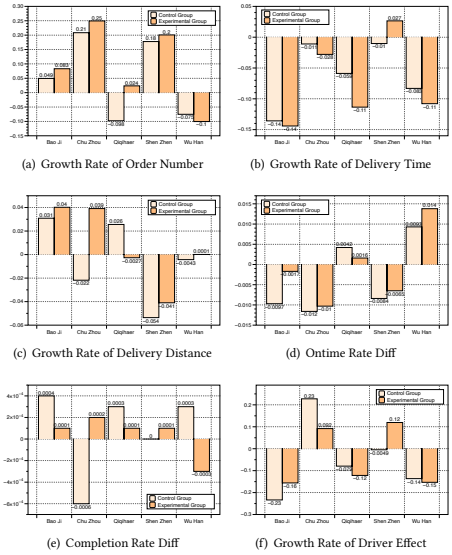
The indicators, $\mathcal{R}_{Met}$ and Δ_{Met}, of control group could reflect the change of metrics without the influence of the delivery scope algorithm. So the divergence between experimental group's Δ_{Met} and control group's Δ_{Met} and the divergence between experimental group's $\mathcal{R}_{Met}$ and control group's $\mathcal{R}_{Met}$ are the impact of our algorithm.

4.1.2 Online Metrics Comparison. In our real-time delivery scenario, the market size of our platform and the delivery efficiency are the two most important aspects that we concerned. Usually, we use GMV or the number of orders on platform to describe the market size. In our experiments, we mainly adopt the maximization of order number (*#order*) as our target. To describe the delivery efficiency, we use the following five metrics:

- **average delivery time (**$\bar{t}$**)**: The average delivery time of orders in a period of time.
- **average delivery distance (**$\bar{d}$**)**: The average delivery distance of orders in a period of time.
- **ontime rate (***OR***)**: The ratio of ontime orders to the whole orders in a period of time.
- **completion rate (***CR***)**: The ratio of completed orders to the whole orders in a period of time.
- **courier efficiency (***CE***)**: The average completed orders in one day for one courier.

Table 1 shows that the new generated delivery scope could bring about 1.68*pp* improvement for the number of orders compared with the control group. At the same time, courier efficiency of experimental group was improved about 1.94*pp*. We can also notice that delivery efficiency was decreased to some extent. On Meituan-Dianping platform, the average delivery time is about 30 minutes, so the average delivery delay caused by algorithm in the experimental group is less than 1.2 minutes. Figure 10 shows part of the experiment results data: in most of the experiment cities, algorithm generated scopes bring much more orders while keeping the delivery efficiency no much worse.

4.1.3 Why Algorithm Outperforms Humans. The reason lies in two aspects: 1) The delivery scope generated by algorithm is better than the manual one. Our operating staff tend to draw a restaurant-centered polygon so as to keep distances between the border points and the restaurant location point to be similar. However, there might be a significant difference between the navigation distance and the straight-line distance. So manual scopes would probably cover blocks far from restaurants. Besides, algorithm-generated scopes include some high-demand area that could not be detected easily. 2) The BIP model in our algorithm customizes scopes for different restaurants according to their own characteristics. Generally, restaurants with higher conversion rate and more convenient traffic condition could get larger scopes. Figure 11 shows three algorithm-generated scopes with significant difference for restaurants in a same station.



(a) Growth Rate of Order Number (b) Growth Rate of Delivery Time

(c) Growth Rate of Delivery Distance (d) Ontime Rate Diff

(e) Completion Rate Diff (f) Growth Rate of Driver Effect

Figure 10: Experiment Results in Part Cities of China: In most cities of China, algorithm generated delivery scopes outperform the manual ones in order number with a little tolerable experience loss.

Figure 11: Algorithm-generated scopes for restaurants in a station: Our algorithm allocate three scopes with significantly different sizes to these restaurants, due to the variations in their conversion rates and traffic condition nearby.

Table 1: Delivery Scope Algorithm Experiment Results in 33 Cities

Group Type	$\mathcal{R}_{\#order}$	$\mathcal{R}_{\bar{t}}$	$\mathcal{R}_{\bar{d}}$	Δ_{OR}	Δ_{CR}	$\mathcal{R}_{CE}$
Experimental Group	12.15%	5.79%	2.02%	-0.93%	-0.05%	5.42%
Control Group	10.47%	2.00%	0.07%	-0.28%	-0.05%	3.48%

Table 2: Performance of ML Methods on Order Estimation

ML method	MAE	RMSE	R^2
Ridge Regression	8.29	18.0873	0.5045
CART	4.08	11.7369	0.7885
Random Forest	3.65	8.0920	0.8621
XgBoost*	3.07	6.5218	0.8813

4.2 Performance of Order Estimation

The experiments dataset is constructed from online delivery scopes and order information. We construct the virtual scopes from our online delivery scope according to different area proportion. In our experiments, we make use of delivery scopes of 2 million online restaurants on Meituan-Dianping platform and data of their corresponding orders in 30 days.

In Table 2 we list the performance of different machine learning models on the estimation of the number of orders. XgBoost outperforms other methods in several evaluation criteria. We get a similar conclusion from experiments on the estimation of average delivery time.

4.3 Performance of Delivery Scope Drawing Algorithm

In this section, we exhibit real intermediate results of the single delivery scope drawing pipeline in our system. Figure 12 displays them step by step from the initial circle to the final delivery scope polygon.

4.3.1 Compression Effects. We evaluate different compression methods in our system. Good compression method of delivery scope keeps original scope shape as much as possible and at the same time use less storage, or points in our scenario. Base on that, we compare the metrics, **compression rate** (CR) and the **normalized symmetric area difference** (NSAD) between different methods. Here, compression rate is the ratio between the number of boundary points in the compressed scope and the number of boundary points in the original scope. We propose the normalized symmetric area difference inspired by [6]. We denote S_{ori} and S_{comp} to the original delivery scope and the compressed one respectively, then the NSAD is defined as follows:

$$NSAD = (Area_{(S_{ori}-S_{comp})} + Area_{(S_{comp}-S_{ori})})/Area_{S_{ori}} \quad (7)$$

We take delivery scopes from 100 online restaurants as our evaluation dataset, and evaluate average performance of CR and NSAD on Angle based compression method and Douglas-Peucker compression method with same hyper-parameters. Details about the comparison can be found in Table 3. Results show that both these two compression methods could keep original scope shape with little deformation, and Douglas-Peucker method could achieve a much higher compression rate.

[2]Figures are generated by mapbox.

Figure 12: Results of Single Delivery Scope Drawing Algorithm: (a) Initialization with Shifted Center Point (b) Navigation Distance Modification (c) Navigation Route Process (d) Remove Stub (e) Scope Compression Process (f) Shipping Scope Intersection (g) Merge High-Demand Areas (h) Remove Improper Location[2]

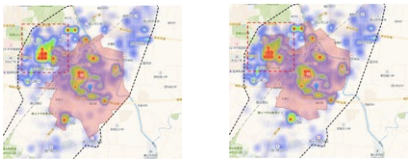
Table 3: Performance of Compression methods

Compression Method	CR (%)	NSAD(m^2)
Angle	66.8	0.000523
Douglas-Peucker*	41.9	0.003

4.3.2 Effect of Covering High-Demand Areas. We exhibit the effect of covering high-demand areas on order distribution heat map by comparing the difference of delivery scopes with and without covering high-demand areas (CHDA) module. Figure 13 shows two generated delivery scopes of the same online restaurant. In the red dot box, we can see the scope generated with CHDA in (b) covers more hot regions compared with the scope generated without CHDA in (a).

4.4 Efficiency for Operations

Before our delivery scope generation system deployed online, operating staff drew all delivery scopes in a manual way. It takes about 3 minutes to draw a delivery scope by hands on average. On our platform, adjustment of delivery scopes is required considering the



(a) Scope before CHDA Process (b) Scope after CHDA Process

Figure 13: CHDA Process merges some hot blocks into the scope generated in the previous procedures. This could increase the number of orders for the corresponding restaurant effectively.

fluctuation of our couriers' number. Due to the large amount of partner restaurants, to draw delivery scopes or to adjust current ones is a time-consuming task. In our delivery scope generation system, 20 servers work for scope generation jobs. A server could generate delivery scopes for restaurants in one station in less than 10 minutes. This enables us to draw delivery scopes for all restaurants on our platform in less than 10 hours.

5 RELATED WORK

Although delivery scope is a special concept in the senario of on-demand delivery service, our work is related to some previous research listed below. [12] and [17] do some analysis on the relationship between the demand&supply of a store and the size of its delivery scope, which they call "service area" or "service outlet". In their work, they just represent a delivery scope as a circle to simplify the problem, which is not accurate in practice. And in some previous location-based retrieval research, authors usually focus on how to represent locations with refined features [16] or how to estimate the user preferences more accurately with some location information [1]. When retrieving the locations for users, previous location-based service also fetch all locations in the circle with specified radius or fetch those locations based on users' interest without considering spatial limitation [14, 18, 19].

6 CONCLUSION

In this paper, we propose a new paradigm for the generation of delivery scopes. In fact, the set of delivery scopes is an important tool to balance the demand of eaters, the supply of restaurants and couriers' capacity in on-demand delivery service. In our framework, at first, a single delivery scope generation algorithm is proposed, which could draw delivery scopes to guarantee experience of users, restaurants and delivery partners. Then in order to achieve a global optimum for a bunch of restaurants, we formalize the delivery scope selection as a binary integer programming problem. Any branch and bound solver could be used to solve this problem to get an exact solution. In order to solve a BIP problem in a large scale efficiently, we propose a heuristic search algorithm to achieve an approximate solution. Results of large scale experiments demonstrate that these algorithm-generated delivery scopes outperform manual ones.

This scope generation framework has been deployed online for several years, and it has been proven to be a successful practice on our on-demand delivery service platform. Now, a next-generation framework has already been built, which we would demonstrate in the future.

REFERENCES

[1] Jie Bao, Yu Zheng, and Mohamed F Mokbel. 2012. Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 20th international conference on advances in geographic information systems*. 199–208.
[2] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.
[3] Stephen P Bradley, Arnoldo C Hax, and Thomas L Magnanti. 1977. Applied mathematical programming. (1977).
[4] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.
[5] Jens Clausen. 1999. Branch and bound algorithms-principles and examples. *Department of Computer Science, University of Copenhagen* (1999), 1–30.
[6] Matt Duckham, Lars Kulik, Mike Worboys, and Antony Galton. 2008. Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. *Pattern recognition* 41, 10 (2008), 3224–3236.
[7] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *Kdd*, Vol. 96. 226–231.
[8] Ying Cha Feng Guo Jinghua Hao Renqing He and Zhizhao Sun Huanyu Zheng, Shengyao Wang. 2019. A Two-Stage Fast Heuristic for Food Delivery RoutePlanning Problem.
[9] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. 1983. Optimization by simulated annealing. *science* 220, 4598 (1983), 671–680.
[10] Hendrik W Lenstra Jr. 1983. Integer programming with a fixed number of variables. *Mathematics of operations research* 8, 4 (1983), 538–548.
[11] Andy Liaw, Matthew Wiener, et al. 2002. Classification and regression by randomForest. *R news* 2, 3 (2002), 18–22.
[12] Shawn Mankad, Masha Shunko, and Qiuping Yu. 2019. How To Find Your Most Valuable Service Outlets: Measuring Influence Using Network Analysis. *Available at SSRN 3366127* (2019).
[13] Stuart J Russell and Peter Norvig. 2016. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,.
[14] Anders Skovsgaard and Christian S Jensen. 2014. Top-k point of interest retrieval using standard indexes. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 173–182.
[15] S-T Wu and MERCEDES ROCio GONZALES Marquez. 2003. A non-self-intersection Douglas-Peucker algorithm. In *16th Brazilian symposium on computer graphics and Image Processing (SIBGRAPI 2003)*. IEEE, 60–66.
[16] Min Xie, Hongzhi Yin, Hao Wang, Fanjiang Xu, Weitong Chen, and Sen Wang. 2016. Learning graph-based poi embedding for location-based recommendation. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 15–24.
[17] Baris Yildiz and Martin Savelsbergh. 2019. Service and capacity planning in crowd-sourced delivery. *Transportation Research Part C: Emerging Technologies* 100 (2019), 177–199.
[18] Shenglin Zhao, Tong Zhao, Haiqin Yang, Michael R Lyu, and Irwin King. 2016. STELLAR: spatial-temporal latent ranking for successive point-of-interest recommendation. In *Thirtieth AAAI conference on artificial intelligence*.
[19] Fan Zhou, Ruiyang Yin, Kunpeng Zhang, Goce Trajcevski, Ting Zhong, and Jin Wu. 2019. Adversarial point-of-interest recommendation. In *The World Wide Web Conference*. 3462–34618.

HeroGRAPH: A Heterogeneous Graph Framework for Multi-Target Cross-Domain Recommendation

Qiang Cui
Meituan
Beijing, China
cuiqiang04@meituan.com

Tao Wei
Meituan
Beijing, China
weitao@meituan.com

Yafeng Zhang
Meituan
Beijing, China
zhangyafeng@meituan.com

Qing Zhang
Meituan
Beijing, China
zhangqing31@meituan.com

ABSTRACT

Cross-Domain Recommendation (CDR) is an important task in recommender systems. Information can be transferred from other domains to target domain to boost its performance and relieve the sparsity issue. Most of the previous work is single-target CDR (STCDR), and some researchers recently propose to study dual-target CDR (DTCDR). However, there are several limitations. These works tend to capture pair-wise relations between domains. They will need to learn much more relations if they are extended to multi-target CDR (MTCDR). Besides, previous CDR works prefer relieving the sparsity issue by extra information or overlapping users. This leads to a lot of pre-operations, such as feature-engineering and finding common users. In this work, we propose a heterogeneous graph framework for MTCDR (HeroGRAPH). First, we construct a shared graph by collecting users and items from multiple domains. This can obtain cross-domain information for each domain by modeling the graph only once, without any relation modeling. Second, we relieve the sparsity by aggregating neighbors from multiple domains for a user or an item. Then, we devise a recurrent attention to model heterogeneous neighbors for each node. This recurrent structure can help iteratively refine the process of selecting important neighbors. Experiments on real-world datasets show that HeroGRAPH can effectively transfer information between domains and alleviate the sparsity issue.

CCS CONCEPTS

• **Information systems** → **Collaborative filtering**; **Recommender systems**.

KEYWORDS

heterogeneous, graph, multi-target, cross-domain

Reference Format:
Qiang Cui, Tao Wei, Yafeng Zhang, and Qing Zhang. 2020. HeroGRAPH: A Heterogeneous Graph Framework for Multi-Target Cross-Domain Recommendation. In *3rd Workshop on Online Recommender Systems and User Modeling (ORSUM 2020), in conjunction with the 14th ACM Conference on Recommender Systems, September 25th, 2020, Virtual Event, Brazil.*

ORSUM@ACM RecSys 2020, September 25th, 2020, Virtual Event, Brazil

1 INTRODUCTION

Collaborative Filtering (CF) has become an effective and efficient technique for recommender systems [13]. However, CF methods often face the sparsity issue as real-world datesets usually have a long tail of users and items with few feedbacks. With the development of CF, Cross-Domain Recommendation (CDR) has been proven to be a promising method to alleviate the sparsity. It can transfer rich information from one domain to another to boost performance.

According to different tasks, previous CDR methods can be roughly divided into two categories, i.e., single-target CDR (STCDR) and dual-target CDR (DTCDR). Most CDR methods belong to the former one, which transfers the information from source domain to target domain and not vice versa. These methods can be based on either the feedbacks [8, 19] or the rich side information [2, 14] to relieve the sparsity. The latter DTCDR has recently been studied. Information from source domain and target domain is mutually utilized to improve the performance of both domains. There are usually two approaches to conduct dual-target modeling. The first way is mostly founded on common users [17, 20] as they can clearly restore information from multiple domains. The second way utilizes mapping function [7, 9] performing as a bridge between domains.

Technically, previous works are good at STCDR and DTCDR, but few people study the multi-target CDR (MTCDR). MTCDR is a generalization of DTCDR. Given at least three domains along with the features and feedbacks, the goal of MTCDR is to boost the performance of all domains. It is a more challenging but more general task in real systems. Previous successful DTCDR methods [7, 20] would have some problems if they were extended to MTCDR. First, DTCDR generally models the pairwise relations between domains. If they directly handle n domains, there will be at least C_n^2 relations. Second, most previous works transfer information by users. It is an indirect way to incorporate cross-domain information, because user behaviors at multiple domains are still processed within each domain. Maybe we can collect all behaviors to devise a shared structure such as graph. Such a structure can directly model within-domain and cross-domain behaviors together, because it can acquire feedbacks from all domains as neighbors for a user or an item.

In this work, we propose a **Hetero**geneous **GRAPH** framework for MTCDR (**HeroGRAPH**). First, we collect ID information of users and items from multiple domains and build a shared graph.

Nodes include users and items. If a user purchases an item, there will be an edge in this graph. Then we use information within each domain to conduct within-domain modeling, and use the shared graph to handle cross-domain information. Besides, we propose a recurrent attention to aggregate neighbors from multiple domains. Last, we combine within-domain embedding and cross-domain embedding to compute user preference and train the model. The main contributions are listed as follows:

- We propose to introduce a shared structure to model information from multiple domains, such as a graph. This structure can greatly simplify the cross-modeling process.
- We propose to aggregate neighbors from all domains for users and items to relieve the sparsity issue. Besides, we introduce a recurrent attention to iteratively refine the aggregation.
- Experiments on real-world datasets reveal that HeroGRAPH outperforms the state-of-the-art methods and is effective in dealing with the sparsity.

2 RELATED WORK

In this section, we review related works including STCDR, DTCDR and graph neural network.

Compared with single-domain recommendation, STCDR can leverage information from source domain to improve the performance of target domain. [19] proposes a deep adaptation-based model to boost the target domain only with ratings. [8] proposes to use spectral convolutions to acquire high-order connectivity and construct domain-invariant user mapping to transfer knowledge. Other works would use text information like description [14] and attribute information [2] to improve performance. STCDR only aims to have a better target domain performance.

Different from STCDR, DTCDR tries to use the information from target domain to boost the source domain. In another word, the goal of DTCDR is mutual improvement. The concept of DTCDR is first proposed in [20]. This work applies common users to obtain the shared data. It needs different methods to obtain pre-trained features. [7] also belongs to DTCDR. It utilizes an orthogonal mapping to connect two models for two domains. The two models can be mutually connected. This work also generates an extension to muti-target CDR but needs orthogonal matrix between every two domains. These representative DTCDR methods only consider the user for building connections between domains.

Graph neural network has become a rising and shining star nowadays. With the success of GCN [6], graph methods quickly catch the eye of researchers. In recent years, GraphSAGE achieves great success as it can acquire inductive embedding [3]. PinSage can be considered a successful industrial practice [18]. Other graph methods such as GAT [15] also promote development in the field. In recommender systems, graph methods are also outstanding. [16] applies meta-path and attention on heterogeneous graph and achieves the state-of-the-art performance. Encouraged by those works, we can also use graph to address problems in CDR, such as alleviating the sparsity by aggregating neighbors.

3 METHODOLOGY

In this section, we propose a heterogeneous graph framework for multi-target cross-domain recommendation (HeroGRAPH) and its diagram is in Fig. 1. We first formulate the problem. Next, we collect feedbacks for each domain and obtain within-domain embedding for each user and item. Then we gather all feedbacks to build a shared graph and acquire cross-domain embedding. Finally, we compute user preference and apply Bayesian Personalized Ranking (BPR) to train the model.

3.1 Problem Formulation

Let $\mathcal{U}_A$ and $\mathcal{I}_A$ be the sets of users and items respectively. The subscript A represents domain A, and so do domain B, domain C, and so on. Refer to u_A, i_A and (u_A, i_A) as user ID, positive item ID, and a positive feedback pair. In this work, we only use these IDs and feedbacks to build model without any side information. Given at least three domains, our target is to improve the performance of all domains.

3.2 Within-Domain Modeling

In the first stage, we collect feedbacks and obtain within-domain embedding for every user and item in each domain. As the only feature we have is ID, we can easily allocate a vector as the initial embedding for each ID. For domain A, this process can be represented as $E_A(\cdot)$ in Fig. 1, and embeddings for u_A and i_A are E_{u_A} and E_{i_A}, respectively.

3.3 Shared Graph and Cross-Domain Modeling

After obtaining the within-domain embedding, we need to acquire cross-domain embedding.

By using feedbacks from all domains, we construct a heterogeneous graph illustrated in Fig. 2 and all domains will use this graph. In real-world, one platform can provide items in many domains for users. If we split user's feedbacks according to domain label, we will obtain many single-domain datasets and user interest will be split. For example, Amazon dataset [10] has many domains, such as Digital Music, Musical Instruments and Amazon Instant Video. Therefore, it is reasonable to reassemble the data from different domains to acquire better user embedding and item embedding.

The cross-domain modeling can be considered as graph modeling, abbreviated as $G(\cdot)$ in Fig. 1. The corresponding embeddings for u_A and i_A can be represented as G_{u_A} and G_{i_A}, respectively. We consider obtaining G_{u_A} as an example to explain how to fuse information from multiple domains. The basic process is conducted by GraphSAGE [3] with max pooling. Now, suppose we have a user in domain A whoes ID is u_A, and its neighbors $N(u_A) = \{i_A, i_B, ..., i_N\}$ are from multiple domains. Their intermediate graph embeddings can be represented as

$$\begin{aligned} q &= h_{u_A} \\ K &= \{h_j \mid j \in N(u_A)\} = \{h_{i_A}, h_{i_B}, ..., h_{i_N}\} \\ V &= \{Ph_j + p \mid h_j \in K\} \end{aligned} \tag{1}$$

where q, K, V are user vector, neighbor's vector and embeded neighbor representation obtained by a fully connected network, respectively. Then, these vectors are used to compute the cross-domain

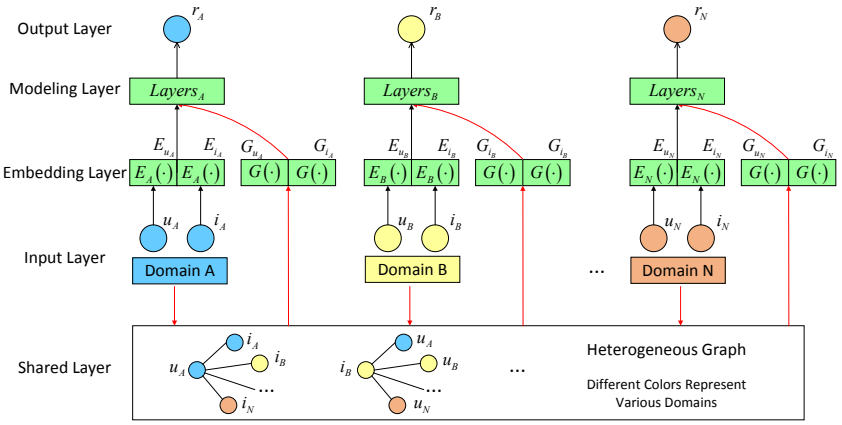


Figure 1: Diagram of the HeroGRAPH model. Black and red arrows between different layers represent the within-domain modeling and cross-domain modeling, respectively. Our model gathers information from multiple domains to construct a heterogeneous graph to transfer knowledge and boost performance of each domain.

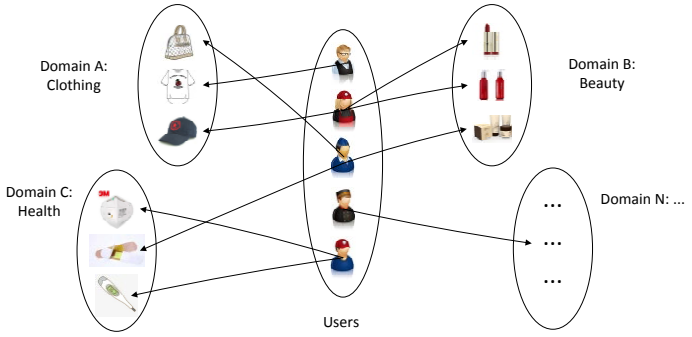


Figure 2: Illustration of the heterogeneous graph. Users may have feedbacks in different domains, and we collect all users and items into one graph as a shared structure. This graph is a bridge among domains. Please note that these domains are limited to one platform, such as Facebook or Amazon.

embedding by

$$\begin{aligned} o_V &= \max(V) \\ G_{u_A} &= \text{ReLU}(W \cdot \text{CONCAT}(q, o_V) + w) \end{aligned} \tag{2}$$

where o_V is the aggregated neighbor representation and the final graph embedding G_{u_A} is a non-linear combination of q and o_V. Please not that although we no longer need to find overlapping users during modeling, graph modeling still depends on overlapping users to incorporate cross-domain informaion.

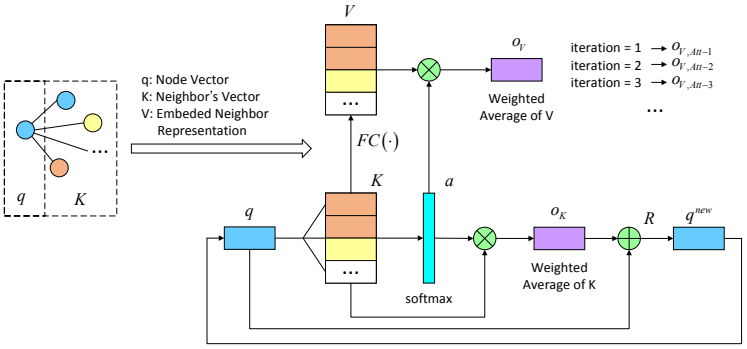


Figure 3: Diagram of the recurrent attention. This attention acts as an aggregator of neighbors of a node. Attention can summarize multiple factors and our work develop a recurrent version to gradually refine this process. The recurrent operation is conducted within node vector q and neighbor's vector K. During each iteration, we compute the output of neighbor's aggregation by attention weight a and embeded neighbor representation V.

3.4 Recurrent Attention for Neighbor Aggregation

In this subsection, we propose a recurrent attention to aggregate neighbors for each node by automatically detecting the importance of each neighbor. The recurrent attention is illustrated in Fig. 3.

Here is the detail of our recurrent attention. First of all, the symbols q, K, V have the same meanings explained in subsection 3.3. Then, the attention weight a is calculated between q and K by Bahdanau Attention [1] and we can obtain a new form of o_V by

$$o_V = V \cdot a \tag{3}$$

As neighbors are from multiple domains, we expect to gradually refine the process of obtaining attention weight a. In order to do this, we aggregate K and update q by

$$\begin{aligned} o_K &= K \cdot a \\ q^{new} &= R \cdot (q + o_K) \end{aligned} \tag{4}$$

where R is a linear mapping and $q + o_K$ is a short-cut connection. Next, q^{new} acts as as new q to recalculate a.

Obviously, we can obtain multiple aggregated neighbor representations as $o_{V,Att-1}, o_{V,Att-2}$ and so on, where subscript $Att-1$ and $Att-2$ means recurrent attention is conducted once and twice respectively. In addition, we add a dropout layer to q and K to avoid overfitting when we first get them.

3.5 Training Framework

In this subsection, we obtain user preference and train the model. Equations are introduced based on domain A.

The positive user preference is calculated based on matrix factorization

$$\hat{x}^t_{ui_A} = E^t_{u_A} \cdot E^t_{i_A} + G^t_{u_A} \cdot G^t_{i_A} \tag{5}$$

where superscript t represents a sample (u_A, i_A) with a certain timestamp. Then we apply the widely-used pair-wise Bayesian Personalized Ranking (BPR) [11] to train the model

$$l^t_{uij_A} = -\ln \sigma \left(\hat{x}^t_{ui_A} - \hat{x}^t_{uj_A} \right) \tag{6}$$

where $\hat{x}^t_{uj_A} = E^t_{u_A} \cdot E^t_{j_A} + G^t_{u_A} \cdot G^t_{j_A}$ is negative preference based on negative feedback pair (u_A, j_A). Finally, the loss function for domain A is

$$\Theta^*_A = \underset{\Theta}{\operatorname{argmin}} \sum_u \sum_{t=1}^{t=|u|} l^t_{uij_A} + \frac{\lambda_\Theta}{2} \|\Theta\|^2 \tag{7}$$

where $|u|$ reprents the number of all samples of user u. The total loss is $\Theta^* = \Theta^*_A + \Theta^*_B + \Theta^*_C + \dots$ and parameters are updated by Adam with default values [5].

4 EXPERIMENTS

In this section, we conduct experiments, analyze the sparsity issue and the proposed recurrent attention.

4.1 Experimental Settings

Datasets. The experiment is conducted on Amazon 5-core dataset [10]. We choose six domains and divide them into two tasks. Each task has three domains. The statistics of each domain are listed in Table 1. Please note that the number of feedbacks is equal to the number of reviews listed on the website [1].

Evaluation Protocols. All datasets are divided into training set, validation set and test set by time. Specifically, the time range of validation set is between 1-Mar.-2014 and 30-Apr.-2014. The ratio of the amount of feedbacks in three sets is approximately 8:1:1. The performance is evaluated on test set by AUC.

[1] http://jmcauley.ucsd.edu/data/amazon

Table 1: Amazon dataset. We divide six domains into two tasks.

Task	Task 1			Task 2		
Domain	Music	Instrument	Video	Clothing	Beauty	Health
# Users	5,541	1,429	5,130	39,387	22,363	38,609
# Items	3,568	900	1,685	23,033	12,101	18,534
# Feedbacks	64,706	10.261	37,126	278,677	198,502	346,355

Baselines. Our model is compared with several baselines. (1) BPR [11]: We choose the BPR-MF to model implicit feedback. This is a popular and powerful single-domain method. (2) DDTCDR [7]: This is a state-of-the-art DTCDR method and we extend it to handle three domains. (3) GraphSAGE-pool [3]: It is a popular graph method and we select its max pooling variant. Besides, as our proposed network has recurrent attention, we generate three variants: HeroGRAPH-Att-1, HeroGRAPH-Att-2 and HeroGRAPH-Att-3.

In this paper, we aim to explore a novel structure for MTCDR. Therefore, we choose widely-used matrix factorization to compute dot product similarity for all methods rather than a learned similarity such as NeuMF [4], as it still needs to be carefully studied [12].

Parameter Settings. Our method is implemented by Tensorflow 2.2 [2] and hyper-parameters are chosen based on validation set. The embedding size for each ID is 8. The regularization parameter λ_Θ is 0.01. As for the graph modeling, we apply a uniform sampling used in GraphSAGE [3] to choose neighbors and the sample sizes for first-order neighbors and second-order neighbors are 10 and 5 respectively. Correspondingly, output embedding sizes of first-layer aggregation and second-layer aggregation are 64 and 16 respectively. These parameters are used across all tasks in our work.

4.2 Hyperparameter Optimization

Dropout is a powerful technique to prevent overfitting. We introduce dropout layers in subsection 3.4 and take our HeroGRAPH-Att-2 as an example to study different dropout rates. The performance on validation set is illustrated in Fig. 4 and we choose the best dropout rate as 0.2 for all tasks in our work.

The best rates for different domains may vary. On Task1, they are 0.4, 0.2 and 0.2 for Music, Instrument and Video respectively. On Task2, Clothing, Beauty and Health have best rates as 0.1, 0.4 and 0.3 respectively. Although best rates vary among domains, we choose the best from a global perspective rather than selecting domain-specific best rates. This strategy will affect performance but reduce the amount of parameters.

4.3 Performance Comparison

The overall performance is listed in Table 2. From an overall point of view, our HeroGRAPH gains the best performance. It achieves significant improvement on task 1, while the performance on task 2 is not big.

On task 1, HeroGRAPH performs good on all three domains. On task 2, we find that some methods are comparative, especially BPR gains best performance on domain Beauty. There are several reasons for this phenomenon. First, three domains of task 2 have much more data than that of task 1. The larger the amount of data, the easier it is to learn a good expression. In this case, the single-domain method can achieve good results and will not be affected by data from other domains. Therefore, it will be difficult to improve on such domains. Second, we treat multiple domains as a whole and use the global optimal hyperparameters. This will cause our model to be unable to achieve optimal performance on each domain. In this unfavorable situation, our model still obtains good results on domain Clothing and Health, which shows the effectiveness of our model.

[2]https://github.com/cuiqiang1990/HeroGRAPH

4.4 Analysis of Sparsity Issue

In this subsection, we analyze the sparsity issue. First we choose a sparse set from the whole test set. We calculate number of occurrence per item in test set and items with no more than 5 feedbacks makes up a sparse set. We count the total numbers of feedbacks of sparse set and test set and divide them to obtain a proportion. The higher the proportion, the more serious the sparsity issue. Statistics and experimental results are listed in Table 3.

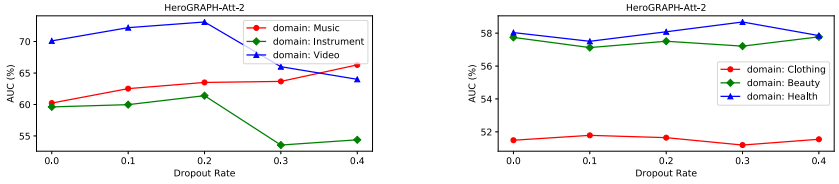
On tasks 1 and 2, our model can achieve great improvement if that domain has a high proportion of sparse items. If there are fewer sparse items, such as in domain Video, Beauty and Health, our HeroGRAPH is not good enough because of the global optimal hyperparameters. This means that by modeling neighbors for users and items, our model can help relieve the sparsity issue.

4.5 Analysis of Recurrent Attention

The analysis of recurrent attention is based on Tables 2 and 3 as our variants are listed in the last three lines of each table. Generally speaking, Att-2 is better than Att-1 and Att-3, and it is also better than GraphSAGE. This means that attention may be more useful and recurrent attention can reduce the variance of data. On the other hand, Att-2 performs comparable with Att-1 on task 2. Nearly all values of Att-3 are smaller than that of Att-2. We can conclude that if we have too many iterations, our model may get overfiting. Therefore, we do not perform more iterations.

5 CONCLUSION

In this work, we propose a heterogeneous graph framework for multi-target cross-domain recommendation (HeroGRAPH). This is a challenging but promising task. We fistly propose to use a shared structure to model information from all the domains such as a graph. Then we propose a recurrent attention to gradually refine the process of neighbor aggregation to relieve the sparsity issue. Experiments show the effectiveness of our model.



(a) Task1 (b) Task2

Figure 4: Performance of HeroGRAPH-Att-2 on validation set with different dropout rates.

Table 2: Performance comparison of baselines and our models.

Task		**Task 1**			**Task 2**		
Domain		Music	Instrument	Video	Clothing	Beauty	Health
Evaluation on test set		AUC (%)			AUC (%)		
Baselines	BPR [11]	65.36	51.15	67.20	53.84	**59.13**	59.68
	DDTCDR [7]	65.21	53.46	67.40	52.50	57.97	60.23
	GraphSAGE-pool [3]	65.50	59.10	71.30	53.17	58.59	60.04
Our HeroGRAPH	HeroGRAPH-Att-1	66.52	60.14	71.20	52.84	58.40	**60.26**
	HeroGRAPH-Att-2	**67.98**	**62.56**	**73.80**	**54.73**	58.18	60.21
	HeroGRAPH-Att-3	66.38	**62.56**	68.80	51.95	57.23	58.63

Table 3: Performance comparison on sparse set. Items in sparse set appear no more than 5 times in test set.

Task - sparse		**Task 1 - sparse**			**Task 2 - sparse**		
Domain		Music	Instrument	Video	Clothing	Beauty	Health
Number of feedbacks in test set	whole set	687	868	4,988	36,002	23,389	41,622
	sparse set	634	647	1,685	23,266	11,363	18,324
	proportion	92.28%	74.53%	33.78%	64.62%	48.58%	44.02%
Evaluation on sparse set		AUC (%)			AUC (%)		
Baselines	BPR [11]	64.51	51.62	**63.32**	52.64	55.25	52.85
	DDTCDR [7]	67.04	51.93	60.50	52.03	54.69	**53.78**
	GraphSAGE-pool [3]	66.26	54.56	61.40	52.74	**55.51**	53.49
Our HeroGRAPH	HeroGRAPH-Att-1	66.25	56.72	63.20	52.11	54.79	53.57
	HeroGRAPH-Att-2	**68.30**	57.34	60.90	**53.26**	53.88	53.75
	HeroGRAPH-Att-2	67.67	**58.73**	59.30	51.84	53.17	52.18

In the future, we will explore other shared structures such as knowledge graph and try to incorporate user profile or item attribute information. Besides, it is promising to add weights to different preferences within one domain, and to weigh losses between multiple domains. Homoscedastic uncertainty may be a good strategy to investigate weight that can be automatically updated during training.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.
[2] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. 2007. Cross-domain mediation in collaborative filtering. In *UM*. Springer, 355–359.
[3] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*. 1024–1034.
[4] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.
[5] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.
[6] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

[7] Pan Li and Alexander Tuzhilin. 2020. DDTCDR: Deep Dual Transfer Cross Domain Recommendation. In *WSDM*. 331–339.

[8] Zhiwei Liu, Lei Zheng, Zhang Jiawei, Jiayu Han, and Philip Yu. 2019. JSCN: Joint Spectral Convolutional Network for Cross Domain Recommendation. 850–859. https://doi.org/10.1109/BigData47090.2019.9006266

[9] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-Domain Recommendation: An Embedding and Mapping Approach.. In *IJCAI*. 2464–2470.

[10] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *ACM SIGIR*. 43–52.

[11] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*. 452–461.

[12] Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. 2020. Neural Collaborative Filtering vs. Matrix Factorization Revisited. *arXiv preprint arXiv:2005.09683* (2020).

[13] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW*. 285–295.

[14] Shulong Tan, Jiajun Bu, Xuzhen Qin, Chun Chen, and Deng Cai. 2014. Cross domain recommendation based on multi-type media fusion. *Neurocomputing* 127 (2014), 124–134.

[15] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. *ICLR* (2018).

[16] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *WWW*. 2022–2032.

[17] Xinghua Wang, Zhaohui Peng, Senzhang Wang, S Yu Philip, Wenjing Fu, and Xiaoguang Hong. 2018. Cross-domain recommendation for cold-start users via neighborhood based feature mapping. In *International Conference on Database Systems for Advanced Applications*. Springer, 158–165.

[18] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *ACM SIGKDD*. 974–983.

[19] Feng Yuan, Lina Yao, and Boualem Benatallah. 2019. DARec: deep domain adaptation for cross-domain recommendation via transferring rating patterns. In *IJCAI*.

[20] Feng Zhu, Chaochao Chen, Yan Wang, Guanfeng Liu, and Xiaolin Zheng. 2019. DTCDR: A Framework for Dual-Target Cross-Domain Recommendation. In *CIKM*. 1533–1542.

Answer-Driven Visual State Estimator for Goal-Oriented Visual Dialogue

Zipeng Xu
xuzp@bupt.edu.cn
Beijing University of Posts and Telecommunications

Fangxiang Feng
fxfeng@bupt.edu.cn
Beijing University of Posts and Telecommunications

Xiaojie Wang*
xjwang@bupt.edu.cn
Beijing University of Posts and Telecommunications

Yushu Yang
yangyushu@meituan.com
Meituan-Dianping Group

Huixing Jiang
jianghuixing@meituan.com
Meituan-Dianping Group

Zhongyuan Wang
wangzhongyuan02@meituan.com
Meituan-Dianping Group

ABSTRACT

A goal-oriented visual dialogue involves multi-turn interactions between two agents, Questioner and Oracle. During which, the answer given by Oracle is of great significance, as it provides golden response to what Questioner concerns. Based on the answer, Questioner updates its belief on target visual content and further raises another question. Notably, different answers drive into different visual beliefs and future questions. However, existing methods always indiscriminately encode answers after much longer questions, resulting in a weak utilization of answers. In this paper, we propose an Answer-Driven Visual State Estimator (ADVSE) to impose the effects of different answers on visual states. First, we propose an Answer-Driven Focusing Attention (ADFA) to capture the answer-driven effect on visual attention by sharpening question-related attention and adjusting it by answer-based logical operation at each turn. Then based on the focusing attention, we get the visual state estimation by Conditional Visual Information Fusion (CVIF), where overall information and difference information are fused conditioning on the question-answer state. We evaluate the proposed ADVSE to both question generator and guesser tasks on the large-scale GuessWhat?! dataset and achieve the state-of-the-art performances on both tasks. The qualitative results indicate that the ADVSE boosts the agent to generate highly efficient questions and obtains reliable visual attentions during the reasonable question generation and guess processes.

CCS CONCEPTS

• **Computing methodologies** → *Computer vision tasks*; *Discourse, dialogue and pragmatics*; *Natural language generation*; *Computer vision representations.*

KEYWORDS

Goal-Oriented Visual Dialogue; Attention Mechanism; Visual State Estimation

ACM Reference Format:
Zipeng Xu, Fangxiang Feng, Xiaojie Wang, Yushu Yang, Huixing Jiang, and Zhongyuan Wang. 2020. Answer-Driven Visual State Estimator for Goal-Oriented Visual Dialogue. In *Proceedings of Proceedings of the 28th ACM International Conference on Multimedia (MM '20).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3413668

1 INTRODUCTION

Goal-oriented Visual Dialogue, which means conducting multi-turn visual-grounded conversations with specific goals, is a comparatively new vision-language task while has attracted increased interests for its research significance and application prospect. As test-beds, image guessing tasks such as Guess-What [6] and Guess-Which [5], i.e. two-player games between Questioner and Oracle to retrieve visual content through dialogue, are proposed. In each round of the dialogue, the Questioner raises a visual-grounded question and gets respond from the Oracle (who predefines the visual target). After several rounds, Questioner is expected to make a right guess at the visual target.

To conduct goal-oriented and vision-coherent dialogue, the AI agent should be able to learn a visual sensitive multimodal representation of the dialogue as well as a dialogue policy. Many works have been done on policy-learning. As Strub et al. [22] first introduce Reinforcement Learning (RL) to explore the dialogue strategy, later works take efforts on reward design [20, 26] or action selection [1, 2]. However, most of them employ a simple way to represent the multimodal dialogue by concatenating the two separately encoded modalities, i.e. language feature encoded by Recurrent Neural Network (RNN) and vision feature encoded by pre-trained Convolutional Neural Network (CNN). To improve the multimodal dialogue representation, various attention mechanisms have been proposed [7, 25, 28], where multimodal interactions are enhanced consequently. Although progresses have been made, unresolved issues still exist.

Firstly, none of the existing representation methods can distinguish among different answers in the dialogue history. The answer is always encoded right after the question without distinction. Since answer is usually a word of yes or no while question contains a longer word string, the effect of answer is relatively weak. However, in fact, answer largely determines the subsequent concerned

*Corresponding author.

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413668

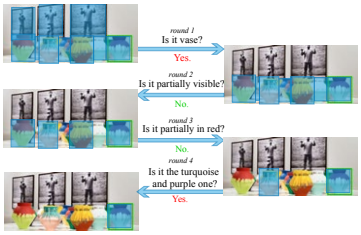


Figure 1: An example of discovering object in a image through dialogue. In which, answer largely determines the subsequent concerned visual information and question. Driven by successive questions and answers, the range of potential objects (highlighted in blue) shrinks and finally focuses on the target one (in green box).

visual information and question. As the object-discovery example in Figure 1, when the answer to the first question "Is it a vase?" is *"yes"*, the questioner continues to pay attention to the vase and asks questions about the features that can best distinguish multiple vases; when the answer to the third question "Is it partially in red?" is *"no"*, the questioner no longer pays attention to the vase in red and instead asks questions about the remaining candidates.

Secondly, the image in previous works is either encoded as a static embedding or attended by the dialogue history, which can hardly capture the influence of different answer on visual information. As mentioned above, different answer results in different concern changes on visual content. Generally, when answer is yes, we will focus on the question-related content for more detailed distinctive information within the confirmed candidates; when answer is no, we need to pay attention to the global area to find new possible candidates. Thus, a proper visual representation should have access to not only the global visual information but also the detailed distinctive information among candidates. Which kind of information is more important is dependent on the current question-answer (QA) state.

To address the above two issues, we propose an Answer-Driven Visual State Estimator (ADVSE), where the visual state is the QA-driven visual information dynamically updated through a dialogue. We formulate the ADVSE process in two steps. We firstly estimate the visual attention with Answer-Driven Focusing Attention (ADFA) and then accordingly estimate the visual state by Conditional Visual Information Fusion (CVIF). ADFA first uses a proposed sharpening operation to polarize the question-guided attention at current round, then inverts or maintains the attention based on different answers, and subsequently accumulates it in the final attention state. The effect of answer on the visual attention state is strengthened in this way. CVIF fuses overall information of the image and the difference information of the current focused candidate from other candidates under the guidance of the current QA, thus obtaining the estimated visual state. We apply ADVSE to build both Question Generator and Guesser for GuessWhat?!, where the specific goal is to discover an undisclosed object in a rich image scene. Experimental results show that both of them achieve state-of-the-art performances.

To conclude, our main contributions are as follows.

- First, we propose an Answer-Driven Visual State Estimator (ADVSE) to capture the influence of different answers in goal-oriented visual dialogue.
- Second, we apply the ADVSE to question generation and guess tasks on the large-scale GuessWhat?! dataset and achieve state-of-the-art performances on both tasks.
- Third, the qualitative results indicate that our ADVSE not only boosts the agent to generate highly efficient questions but also presents reliable visual attention during the reasonable question generation and guess processes.

2 RELATED WORKS

Goal-oriented dialogue requires the agent to complete a related task with a clear goal through multiturn conversations. Although goal-oriented spoken and text-based dialogues have been studied in Natural Language Processing committee for years [4, 15, 24], goal-oriented visual dialogue extends the setting to vision domain and is a relatively new and challenging field. Representatively, GuessWhat?! [6] aims to identify a predefined object in a real-world image through dialogue and GuessWhich [5] is to figure out the referring image among various images. There are typically two dialogue agents, a Questioner and an Oracle, communicating together while the Questioner asks questions to figure out the undisclosed target and the Oracle, who predefines the target, responds accordingly.

Question Generation is a core task in goal-oriented visual dialogue. De Vries et al. [6] first proposed a supervised model, where they extended the Hierarchical Recurrent Encoder Decoder (HRED) [17] by introducing the visual information, which is the image's FC8 feature obtained from a pre-trained VGG [21]. After that, various researches focused on dialogue policy learning. Strub et al. [22] introduced Reinforcement Learning (RL) to explore different dialogue strategy, which regarded question generation as a Markov Decision Process and used whether enabling a right guess as the reward function. Zhao et al. [27] proposed a Temperature Policy Gradient method to make balance of exploration and exploitation while selecting words. Zhang et al. [26] designed a fine-grained reward mechanism based on the information provided by Oracle and Guesser. Some researchers explored the use of information uncertainty or changes to generate valuable questions [2, 11, 20].

In these methods, the multimodal dialogue is encoded in the simplest way, where the CNN-encoded static image embedding is concatenated with the RNN-encoded changing dialogue history embedding to serve as the multimodal representation. However, encoding image as a static embedding is irrational, for the concerned image content changes as the dialogue progresses. Other than the simplest method, some attention-based methods are proposed to model the interaction between dialogue and image, computing dynamic visual information through dialogue. In PLAN network [28], the dialogue history embedding is jointly used with the image embedding to compute the attention on different regions, making it possible to provide dynamic visual information at each round. Deng

et al. [7] proposed Accumulated Attention (A-ATT) mechanism that consists of three kinds of attention (query attention, image attention and objects attention), where the image is attended under the joint effect of dialogue history and object feature. Yang et al. [25] proposed a History-Aware Co-attention Network which includes two co-attention module, feature-wise co-attention module and element-wise co-attention module, while both of the attention are computed under the guidance of question and history feature.

As we can see, none of the existing methods give special consideration to the effect of different answers. Most of the previous works weaken answers' effect by indiscriminately encoding the much shorter answers with a dialogue history encoder. On the contrary, the proposed Answer-Driven Visual State Estimator (ADVSE) explicitly exploits different answers in different ways to update the visual attention at each step and further fuses two types of visual information conditioning on different QA-state.

3 ANSWER-DRIVEN VISUAL STATE ESTIMATOR

This section introduces the proposed Answer-Driven Visual State Estimator (ADVSE). As in Figure 2, the estimator contains three parts, which are Encoders, ADFA-based Attention State Update (ADFA-ASU) and Conditional Visual Information Fusion (CVIF).

In the Encoders, visual information and language information are encoded separately. The ADFA-ASU estimates the visual attention while greatly considering the answer-driven effect with the proposed ADFA. Based on the estimated visual attention, the CVIF estimates the visual state by fusing the attended object overall information and the attended object difference information conditioning on the current QA state. They are introduced in detail below.

3.1 Encoders

Visual feature. Given the input image I, Faster-RCNN [16] is used to encode the image information. According to the static features provided by bottom-up attention [3], the image representation is obtained:

$$I = RCNN(image), \tag{1}$$

of which, top-K region proposals are selected from each image. Here, K is simply fixed as 36, i.e. $I = \{i_1, i_2, \ldots, i_{36}\} \in R^{36\times 2048}$.

Language feature. Given the t rounds dialogue history $H = \{(q_1, a_1), \ldots, (q_t, a_t)\}$, where q_t is the t-th round question and a_t is the t-th round answer, a 2-layer GRU is applied to encode the dialogue. In concrete, the t-th round question q_t, which includes m words and whose word embeddings are $\{w^q_{t,1}, \ldots, w^q_{t,m}\}$, is encoded by GRU^w:

$$h^q_{t,i} = GRU^w(w^q_{t,i}, h^q_{t,i-1}). \tag{2}$$

We use the last hidden state $Q_t = h^q_{t,m}$ as the representation of the question.

Similarly, the representation of current answer A_t can be obtained. By feeding Q_t and A_t to the upper layer GRU^c, the representation of t-th round dialogue history H_t is obtained:

$$H_t = GRU^c([Q_t; A_t], H_{t-1}). \tag{3}$$

3.2 ADFA-ASU

During the visual dialogue process, the attention state to the image dynamically updates, driven by the dialogue history and the current QA. In this section, we formulate the attention updating process by the proposed ADFA-ASU. At t-th round, the attention state att_t is updated by two parts: current QA caused Answer-Driven Focusing Attention (ADFA) att^q_t and history guided attention att^h_t. The concrete modeling of att^q_t and att^h_t are described below:

Firstly, in Answer-Driven Focusing Attention (ADFA), the current turn QA-guided focusing attention state att^q_t is modeled by the following four steps:

Step 1, calculate the question-guided image attention α^q_t according to Eq. 4-7:

$$Q^m_t = \{h^q_{t,i}\}^m_{i=1}, \tag{4}$$

$$\widetilde{Q}^k_t = softmax(Q^m_t W^k_q)^T \odot Q^m_t,\ \widetilde{Q}_t = [\widetilde{Q}^1_t; \widetilde{Q}^2_t], \tag{5}$$

$$F^q_t = W_Q \widetilde{Q}_t \odot W^q_I I, \tag{6}$$

$$\alpha^q_t = Softmax(W^q_F F^q_t + g). \tag{7}$$

In order to extract the important information within a question, a 2-glimpse attention is utilized to extract the current question feature $\widetilde{Q}_t$ as in Eq. 4-5. The textual question feature and visual feature is then fused by Hadamard Product (Eq. 6). To enable end-to-end training with the subsequent discrete decision, we introduce Gumbel-Softmax sampler [8] as well as the Gumbel-Softmax training trick [9, 12] to compute the attention distribution as in Eq. 7. In concrete, we add g (i.i.d. samples from Gumbel distribution) before the softmax activation during the training stage.

Step 2, polarize the α^q_t by a sharpening operation as shown in Eq. 8-9 to figure out the question-correlated objects:

$$norm(\alpha^q_t) = \frac{\alpha^q_t - min(\alpha^q_t)}{max(\alpha^q_t) - min(\alpha^q_t)}, \tag{8}$$

$$P(\alpha^q_{t,k}) = \begin{cases} 1, & if norm(\alpha^q_{t,k}) > \gamma, \\ 0, & else. \end{cases} \tag{9}$$

The attention sharpening operation project the attention weight of each block $\alpha^q_{t,k} \in \alpha^q_t \in (0, 1)$ into a binary value $P(\alpha^q_{t,k}) \in \{0, 1\}$. It first applies the max-min normalization to α^q_t and gets $norm(\alpha^q_t)$ (Eq. 8), then filters the normalized attention by a threshold γ (i.e. a hyperparameter) to get the polarized value $P(\alpha^q_{t,k})$ (Eq. 9), which represents whether the object i_k correlates to what q_t asks.

Step 3, based on $P(\alpha^q_t)$, the answer to q_t is used to determine the direction of the attention mask M^q_t as shown in Eq. 10:

$$M^q_t = \begin{cases} P(\alpha^q_t), & if a_t == YES, \\ 1 - P(\alpha^q_t), & if a_t == NO, \\ 1, & otherwise. \end{cases} \tag{10}$$

If the answer is *"yes"*, the attention mask is $P(\alpha^q_t)$, which means the agent will hold attention on the currently concerned objects. The agent will keep paying close attention to the objects with the $P(.)$ of 1 and keep paying no attention to those objects with the $P(.)$ of 0. If the answer is *"no"*, the attention mask is $1 - P(\alpha^q_t)$, which means the attentions on objects is going to be reversed. The agent will transfer its attentions to other objects whose $P(.)$ is 0 and no longer concern the objects whose $P(.)$ is 1 as they are denied

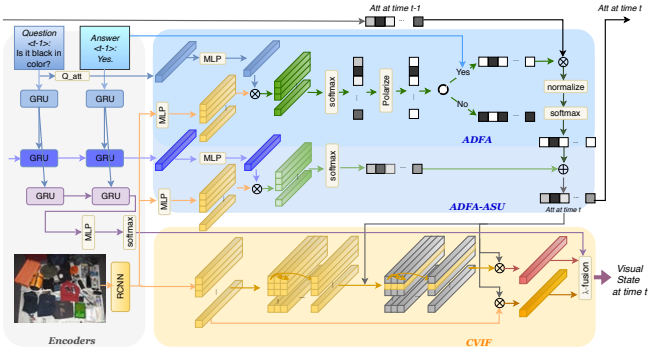


Figure 2: Block Diagram of the proposed Answer-Driven Visual State Estimator (ADVSE).

by the Oracle. Otherwise, if the answer is *"N/A"*, the att_t^q will be kept unchanged, which is achieved by letting the elements in M_t^q be 1 for all candidates. In this way, the answer plays a key role on forming the subsequent visual attention and therefore affects the visual state.

Step 4, after calculating the influence of current round of question and answer, we update the focusing attention state att_t^q. The obtained attention mask M_t^q is applied on the previous attention state att_{t-1} by a Hadamard Product, and is then normalized. A learnable parameter τ and masked softmax are utilized to adjust the updated att_t^q as shown in Eq. 11:

$$att_t^q = maskedSoftmax(\frac{Norm(M_t^q \odot att_{t-1})}{\tau}). \tag{11}$$

Secondly, the history guided attention is calculated as follows:

$$att_t^h = softmax(W_F^H(W_t^H H_t \odot W_I^H I)). \tag{12}$$

Finally, we get the estimated attention state att_t by adding att_t^q and att_t^h:

$$att_t = att_t^h + att_t^q \tag{13}$$

The attention state is dynamically updated and gradually focused in this way as successive QA pair generates.

3.3 CVIF

In CVIF, we firstly compute the attended difference information D_{att}^t and the attended overall information I_{att}^t based on the attention state estimated in ADFA-ASU. Finally, we fuse the two types of visual information conditioning on the current QA to obtain the current visual state estimation V_t.

First, the difference information between the mostly focused object and others is achieved in two steps as follows:

Step 1, select the mostly focused object $i_{selected^t}$ according to the att_t:

$$selected^t = argmax(att_t). \tag{14}$$

Step 2, compute the difference between $i_{selected^t}$ and other object, and then get the focused difference information guided by att_t, as described by the following formulas:

$$D_{selected}^t = \{i_{selected^t} - i_j\}_{j=1}^N, \tag{15}$$

$$D_{att}^t = D_{selected}^t \otimes att_t. \tag{16}$$

Then, the overall feature is calculated by:

$$I_{att}^t = I \otimes att_t. \tag{17}$$

Finally, D_{att}^t and I_{att}^t are fused conditioning on current QA-pair. The QA pair is first encoded as shown in Eq. 18, and then normalized by softmax to obtain the conditioning factor λ_t as shown in Eq. 19. Then the estimated visual state V_t is obtained by weighted summing the D_{att}^t and I_{att}^t with the factor λ_t, as shown in Eq. 20.

$$h_{t,q}^p = GRU^p(q_t, h_0),\ P_t = GRU^p(h_{t,q}^p, a_t), \tag{18}$$

$$(\lambda_t, 1-\lambda_t) = softmax(W_p P_t), \tag{19}$$

$$V_t = \lambda_t \odot D_{att}^t + (1-\lambda_t) \odot I_{att}^t. \tag{20}$$

Visual state estimation is a soft fusion of difference information and overall information conditioned on current QA-pair, which strengthens again the influence of current answer.

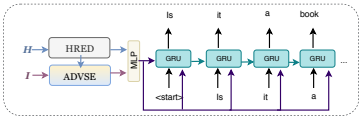
4 USING ADVSE FOR QGEN AND GUESSER

ADVSE is a general framework for goal-oriented visual dialogue. In this section, we apply it to model the Question Generator (QGen) and Guesser in GuessWhat?! game. We firstly combine the ADVSE with an ordinary hierarchical history encoder to get the multimodal dialogue representation:

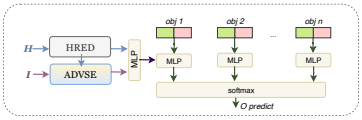
$$F_t = tanh(W_f[H_t; V_t]). \tag{21}$$

In which, H_t is the encoding of dialogue history (as in Eq. 3) and V_t is the visual state estimated by ADVSE (as in Eq. 20). They are concatenated and then projected by an MLP to get the multimodal dialogue representation F_t.

On the basis of F_t, the ADVSE-QGen and ADVSE-Guesser are introduced as follows.



(a) ADVSE-QGen



(b) ADVSE-Guesser

Figure 3: Using ADVSE for QGen and Guesser.

4.1 ADVSE-QGen Model

In the process of visual dialogue, given image I, dialogue history $H = \{(q_1, a_1), \ldots, (q_t, a_t)\}$, the QGen model needs to generate new question $q_{t+1} = (w_0^{t+1}, w_1^{t+1}, \ldots, w_m^{t+1})$, so as to get more information of the target object. As shown in Figure 3(a), the ADVSE-QGen Model is mainly modeled by ADVSE, HRED and a decoder.

Specifically, after ADVSE-based multimodal representation F_t is obtained, the decoder takes F_t as the initial incentive:

$$h_{dec}^{t,0} = F_t. \tag{22}$$

The decoder is a single-layer GRU. The word vector is concatenated with the visual state estimation V_t and the previous state, then used as the input at current state to predict the next word:

$$w_i^{t+1} = GRU_d([w_{i-1}^{t+1}; V_t], h_{dec}^{t,i-1}). \tag{23}$$

When stop token appears, the generation ends.

4.2 ADVSE-Guesser Model

With image I and completed dialogue history $H = \{(q_1, a_1), \ldots, (q_T, a_T)\}$ in hands, a Guesser is expected to select the target object o^* from the candidates $O = \{o_1, o_2, \ldots, o_n\}$ while it has access to the spatial information s_O and the category information c_O in addition. The ADVSE-Guess Model is mainly modeled by ADVSE, HRED and a classifier as shown in Figure 3(b).

The classifier first encodes the object representation r_O from its category and spatial information as in Eq. 24, which is the same as the previous models [22].

$$r_O = ReLU(W_o^2 ReLU(W_o^1[s_O; c_O])). \tag{24}$$

Then, softmax function is applied on the dot product between F_T and r_O to get the probability distribution. At last, the one with the maximum probability is selected:

$$o_{predict} = argmax(softmax(F_T{}' r_O)). \tag{25}$$

5 EXPERIMENTS

We evaluate the models on the GuessWhat?! dataset, which has 155,281 dialogues based on 66,537 images, containing 134,074 different objects. There are 821,955 question-answer pairs in the dataset while the vocabulary size is 4900. We use standard dataset split (train set, validate set, test set).

In this section, we firstly report experimental results of ADVSE-QGen and ADVSE-Guesser respectively. We introduce the training details and evaluation metric, make comparisons with the state-of-the-art models and provide qualitative results. To verify the contribution of each component under different tasks, we conduct ablation study on both ADVSE-QGen and ADVSE-Guesser. Further, we report the experimental results of jointly using ADVSE-QGen and ADVSE-Guesser. The codes of our models are available at https://github.com/zipengxuc/ADVSE-GuessWhat.

5.1 ADVSE-QGen

5.1.1 Training Details. The QGen model is firstly trained in supervised way, and then trained by reinforcement learning.

In supervised learning, we minimized the negative likelihood loss. We use Adam [10] with an initial learning rate of 1e-3, a batch size of 64 to train the QGen model for 20 epochs. Learning rate is decayed by 0.9 per epoch. The hyperparameter γ in Sharpening Operation is set as 0.7.

Further, we train the model using the same reinforcement learning method as the baseline model [22], where the QGen is modeled as a Markov Decision Process and uses the 0-1 reward that depends on whether a right guess can be made. We use Stochastic Gradient Descent (SGD) to train the model for 500 epochs with a learning rate of 1e-3 and a batch size of 64. We set the maximum round T = 8, the maximum length of each sentence m = 12. We use the same standard Oracle and Guesser as [22] while the trained benchmark Oracle and Guesser's errors on the test set are 21.9% and 35.9%, respectively.

5.1.2 Evaluation Metric and Comparison Models. Following existing studies (such as [6]), we use the game success rate as the evaluation metric and evaluate in 3 generating way (i.e., sampling, greedy, and beam search (beam size=20)) by 2 test settings, i.e. New Object (games with seen images in train set but randomly sampled new target) and New Image (games with unseen images in test set).

We make comparisons in supervised training fashion and advanced training fashion (includes reinforcement learning and cooperative learning) respectively. The 3 supervised models are: the baseline SL [6], the DM [18] and the current state-of-the-art model VDST-SL [13]; 9 advanced training models are: baseline RL [22], GDSE-C [19], TPG [27], VQG [26], ISM [1], Bayesian [2], RIG as rewards (RIG-1), RIG as a loss with 0-1 rewards (RIG-2) [20] and the current state-of-the-art model VDST-RL [13].

5.1.3 Quantitative Results. Table 1 shows the comparisons among models in supervised learning and reinforcement learning, respectively. To be fair, all models in comparisons use the standard Oracle and Guesser model in this part.

Supervised learning. As in the upper part of Table 1, our ADVSE-QGen achieves the best performance. With the standard Guesser [22], the model achieves the success rate of 50.66% on New

Table 1: A comparison results of QGen on the task success rate evaluated by two types of Guesser, i.e. the standard Guesser[22] and the proposed ADVSE-Guesser. The upper part shows the results in SL while the bottom part shows the results in RL.

	Approach	(%)New object				(%)New game			
		Sampling	Greedy	Beam-search	Best	Sampling	Greedy	Beam-Search	Best
Guesser[22]	SL	41.6	43.5	47.1	47.1	39.2	40.8	44.6	44.6
	DM	-	-	-	-	-	-	-	42.19
	VDST-SL	45.02	49.49	-	49.49	44.24	45.94	-	45.94
	ADVSE-QGen	**47.55**	**50.66**	**47.47**	**50.66**	**44.75**	**47.03**	**44.70**	**47.03**
ADVSE-Guesser	**ADVSE-QGen**	**48.01**	**54.06**	**50.66**	**54.06**	**46.32**	**50.94**	**47.89**	**50.94**
Guesser[22]	RL	62.8	58.2	53.9	62.8	60.8	56.3	52.0	60.8
	VQG	63.2	63.6	63.9	63.9	59.8	60.7	60.8	60.8
	Bayesian	61.4	62.1	63.6	63.6	59.0	59.8	60.6	60.6
	GDSE-C	-	-	-	63.3	-	-	-	60.7
	ISM	-	64.2	-	64.2	-	62.1	-	62.1
	TPG	-	-	-	-	-	-	-	62.6
	RIG-1	65.20	63.00	63.08	65.20	64.06	59.00	60.21	64.06
	RIG-2	67.19	63.19	62.57	67.19	65.79	61.18	59.79	65.79
	VDST-RL	69.51	70.55	71.03	71.03	66.76	67.73	67.52	67.73
	ADVSE-QGen	**71.26**	**72.73**	**72.24**	**72.73**	**68.82**	**69.88**	**69.88**	**69.88**
ADVSE-Guesser	**ADVSE-QGen**	**72.38**	**73.59**	**73.73**	**73.73**	**70.61**	**71.10**	**71.27**	**71.27**

object and 47.03% on New game, exceeding the state-of-the-art model VDST-SL in all settings.

Reinforcement learning. As can be seen in Table 1 (lower part), the success rate of our ADVSE-QGen is significantly better than the previous methods in any case. Even though we use a simple 0-1 reward, compared with other models that use more finely designed rewards, we still achieve better performance. For example, our model achieves 9.08 points of improvement on New game compared to the VQG model, which designs three fine-grained rewards. Compared with the RIG-1 model that uses informative reward, our model achieves a higher success rate of 5.82 points on New game and 7.53 points on New object. Compared with the current state-of-the-art model VDST-RL, we have improved the success rate in all aspects, gaining an absolute advantage of 2.15 points on New game. In summary, using the benchmark Guesser [22] as the training environment, our model has achieved a maximum success rate of 72.73% on New object and 69.88% on New game, and achieves the new state of the art.

5.1.4 Qualitative Results. Figure 4 shows a visualized example of the question generation process and the changing of visual attention state (att_t) in our model. In each subgraph, the blue box annotates the target object; the red, orange and yellow boxes annotate the candidates with the top-3 largest attention weights at current round. As we can see, at the beginning of the conversation (round 1), the agent asks the *"Is it a person?"*. After getting the answer of *"no"*, the attention shifts to the non-persons and asks, *"Is it a truck?"* (round 2). The Agent keeps on asking new objects until a positive answer to *"Is it a car?"* is received, the attention is then focused on the differences among various cars, such as position, e.g. *"Is it in front?"* is raised (round 4). Driven by following QA pairs, the attention state gradually focuses to the target object that is the more front-end car in the picture (round 8). It can be seen that the questions generated in each round are highly related to the current interested visual

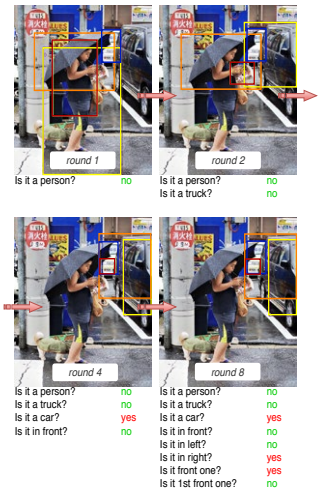
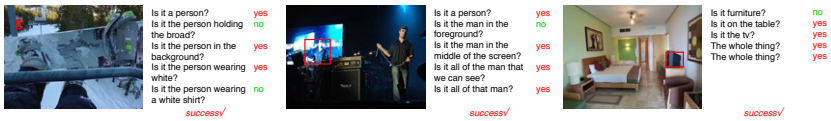


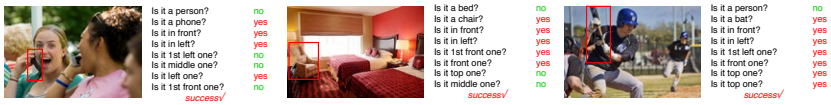
Figure 4: Illustration for the process of question generation.

content, and the attention state changes according to the acquired answer. These phenomena fit well with the designed mechanisms.

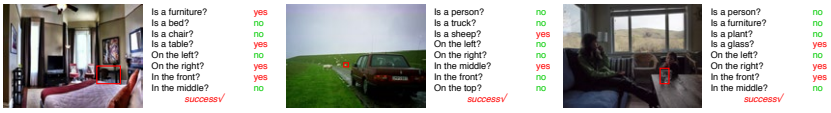
Figure 5 gives additional dialogue examples generated by ADVSE-QGen under different training settings. As can be seen, the generated questions are highly related to the image. As the first example



(a) Generated dialogue samples of the ADVSE-QGen trained by Supervised Learning.



(b) Generated dialogue examples of ADVSE-QGen trained by Reinforcement Learning with the standard Guesser.



(c) Generated dialogue examples of the ADVSE-QGen trained by Reinforcement Learning with the ADVSE-Guesser.

Figure 5: Generated dialogue examples of the ADVSE-QGen under different training settings. The target is annotated in red.

Table 2: Results of ablation study on QGen.

	(%)SL		(%)RL	
	New object	**New game**	**New object**	**New game**
ADVSE-QGen	**50.66**	**47.03**	**72.73**	**69.88**
w/o SO	48.05	45.93	72.04	68.96
w/o ADFA	47.69	45.21	70.48	68.20
w/o CVIF	47.77	45.68	70.90	68.40

in Figure 5(a), the agent raises detailed questions, such as *"Is it the person holding the board?"* and *"Is it the person wearing white?"*, that describe the distinctive object feature comprehensively. Also, the ADVSE-based agents seem to follow some specific strategies. Notably, positive answers always bring about more detailed questions while negative answers lead to questions about the non-excluded objects. Moreover, the model is able to generate questions in a fine-grained differential style, such as *"Is it the 1st front one?"*, which is very efficient for achieving goals.

5.1.5 Ablation Study. We evaluate the individual contribution of the following components: 1) SO: we remove the Sharpening Operation (SO) in ADFA so that the question-guided attention is directly adjusted by the answer without polarizing afore; 2) ADFA: we remove the whole part of ADFA so that the attention is merely guided by history; 3) CVIF: we remove the whole part of CVIF so that only overall visual information can be used. We conduct the ablation study with the standard Oracle and Guesser.

As in Table 2, the result is showed in two training fashions, Supervised Learning (SL) and Reinforcement Learning (RL). It can be seen that without ADFA and CVIF, the performance of QGen model drops significantly, demonstrating their substantial contribution to goal-oriented visual question generation. Besides, the Sharpening Operation (SO) is validated to be an effective step in ADFA.

5.2 Guesser

5.2.1 Training Details. Guesser is trained in supervised way and is optimized by minimizing the negative likelihood loss. We use the Adam [10] optimizer to train the Guesser model for 20 epochs with a learning rate of 1e-3, a batch size of 64. Learning rate is decayed by 0.9 per epoch. The hyperparameter γ in Sharpening Operation in ADFA is set as 0.7.

5.2.2 Evaluation Metric and Comparison Models. Guesser model is evaluated by classification error rate. The 2 baseline models [6]: HRED, HRED-VGG, 3 attention-based models PLAN [28], A-ATT [7], HACAN [25], and 2 Feature-wise Linear Modulation (FiLM) models: single-hop FiLM [14], multi-hop FiLM [23], are compared.

5.2.3 Quantitative Results. Table 3 compares the test error of Guess models. Except for HRED (the first row in the table), all models utilize image feature, dialogue history, object spatial and category feature as input. As HRED+VGG compared to HRED, simply adding image feature will decrease the performance. However, applying appropriate attention mechanism to image helps the model to achieve higher performance, according to the PLAN, A-ATT and HACAN models. FiLM layers take effects either. Overall, it can be seen from the table that the Guesser model with our ADVSE structure achieves the lowest test error of 33.15%, exceeds all the previous models and achieves the new state of the art.

Table 3: Comparison Results of the Guesser.

Model	(%)Test err
HRED	39.0
HRED+VGG	39.6
PLAN	36.6
A-ATT	35.8
Single-hop FiLM	35.7
Multi-hop FiLM	35.0
HACAN	34.1
ADVSE-Guesser	**33.15**
w/o SO	33.45
w/o ADFA	33.50
w/o CVIF	33.65

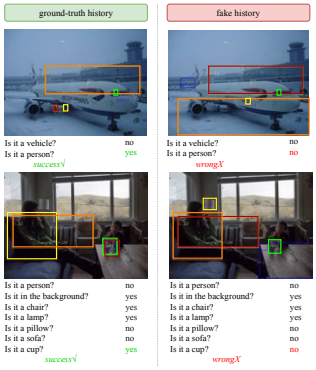


Figure 6: Visualization of the visual attention state in guess process. The left column is provided with the ground-truth history while the right column is provided with the fake history. The green box annotates the target object.

5.2.4 Qualitative Results. Figure 6 illustrates the qualitative examples on Guess model. To illustrate the proposed ADFA mechanism, we visualize the visual attention state in the guess process. The red, orange and yellow boxes annotate the candidates with the top-3 largest attention weight. Further, we substitute the ground-truth history by fake history to make comparisons. It is clear that when current question is answered *"yes"*, our guess model focuses on the question-relevant objects. On the contrary, as in fake examples, when current question is answered with *"no"*, the model immediately transfers the attention to question-irrelevant objects. Moreover, as the right answers are taken place in the fake history, the guess results go wrong. The distinct results reflect the effectiveness of the proposed ADFA.

5.2.5 Ablation Study. We evaluate the individual contribution following the same setting as in section 5.1.5, i.e. SO, ADFA and CVIF. As in Table 3, without ADFA and CVIF, the Guesser results in comparatively worse performances. Besides, SO is still of significance in Guesser. To further illustrate the effect of each part, we conduct Significance Test on the four models. In concrete, we train each model for 10 times with random initialization and then conduct T-test on the collected data. Accordingly, ADFA, CVIF and SO are verified to be significant (with the p-value of 0.001, 0.001, 0.01).

5.3 Joint QGen and Guesser

Further, we combine the proposed QGen and Guess model. Both in the supervised learning and reinforcement learning processes for QGen, we replace the standard Guesser with our ADVSE-Guesser. We show the quantitative results Table 1. In SL, the model achieves the success rate of 54.06% on New object and 50.94% on New game, which are the best performances in supervised training to the best we know. In RL, the model achieves the success rate of 73.73% on New object and 71.27% on New game, which ulteriorly improves the performance. Overall, jointly using the ADVSE-QGen and ADVSE-Guesser, we achieve even better performance on GuessWhat?! task.

We give the generated dialogue examples in Figure 5(c). Jointly using ADVSE-QGen and ADVSE-Guesser generates dialogue in a more concise way. Still, the dialogue strategy is clear. Take the middle in Figure 5(c) as an instance. The agent firstly raises question to figure out the specific category of the target, like *"Is a person?"*, *"Is a truck?"*. Further, as obtained the positive answer *"yes"* to *"Is a sheep?"*, the agent then raises question in a detailed distinctive way to distinguish among many sheep. It successively asks *"On the left?"*, *"On the Right?"*, *"In the middle?"*, *"In the front?"* and finally reaches the target sheep, which is the middle but back one.

We combine the ADVSE-QGen and ADVSE-Guesser in a rather simple way in this section while further explorations for jointly using the two homologous models are expected in the future.

6 CONCLUSIONS

This paper proposes an Answer-Driven Visual State Estimator (ADVSE) to impose the significant effect of different answers on visual information in goal-oriented visual dialogue. First, we capture the answer-driven effect on visual attention by Answer-Driven Focusing Attention (ADFA), where whether to hold or shift the question-related visual attention is determined by different answer at each turn. Further, in Conditional Visual Information Fusion (CVIF), we provide two-types of visual information for different QA state and then conditionally fuse them as the estimation of visual state. Applying the proposed ADVSE to question generation task and guess task in Guesswhat?!, we achieve improved accuracy and qualitative results in comparison to existing state-of-the-art models on both tasks. Moving forward, we will further explore the potential improvements of jointly using the homologous ADVSE-QGen and ADVSE-Guesser.

ACKNOWLEDGMENTS

We thank the reviewers for their comments and suggestions. This paper is partially supported by NSFC (No. 61906018), MoE-CMCC "Artificial Intelligence" Project (No. MCM20190701), the Fundamental Research Funds for the Central Universities and Huawei Noah's Ark Lab.

REFERENCES

[1] Ehsan Abbasnejad, Qi Wu, Iman Abbasnejad, Javen Shi, and Anton van den Hengel. 2018. An Active Information Seeking Model for Goal-oriented Vision-and-Language Tasks. *CoRR* abs/1812.06398 (2018). arXiv:1812.06398 http://arxiv.org/abs/1812.06398

[2] Ehsan Abbasnejad, Qi Wu, Javen Shi, and Anton van den Hengel. 2018. What's to Know? Uncertainty as a Guide to Asking Goal-Oriented Questions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4150–4159.

[3] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 6077–6086.

[4] Antoine Bordes and Jason Weston. 2016. Learning End-to-End Goal-Oriented Dialog. *CoRR* abs/1605.07683 (2016). arXiv:1605.07683 http://arxiv.org/abs/1605.07683

[5] Abhishek Das, Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. 2017. Learning cooperative visual dialog agents with deep reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*. 2951–2960.

[6] Harm De Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. 2017. Guesswhat?! visual object discovery through multi-modal dialogue. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5503–5512.

[7] Chaorui Deng, Qi Wu, Qingyao Wu, Fuyuan Hu, Fan Lyu, and Mingkui Tan. 2018. Visual Grounding via Accumulated Attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7746–7755.

[8] Emil Julius Gumbel. 1948. *Statistical theory of extreme values and some practical applications: a series of lectures*. Vol. 33. US Government Printing Office.

[9] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*.

[10] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014).

[11] Sang-Woo Lee, Yu-Jung Heo, and Byoung-Tak Zhang. 2018. Answerer in questioner's mind: information theoretic approach to goal-oriented visual dialog. In *Advances in Neural Information Processing Systems*. 2579–2589.

[12] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *International Conference on Learning Representations*.

[13] Wei Pang and Xiaojie Wang. 2020. Visual Dialogue State Tracking for Question Generation. In *Association for the Advancement of Artificial Intelligence*.

[14] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. 2018. Film: Visual reasoning with a general conditioning layer. In *Association for the Advancement of Artificial Intelligence*.

[15] Janarthanan Rajendran, Jatin Ganhotra, Satinder Singh, and Lazaros Polymenakos. 2018. Learning End-to-End Goal-Oriented Dialog with Multiple Answers. *CoRR* abs/1808.09996 (2018). arXiv:1808.09996 http://arxiv.org/abs/1808.09996

[16] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*. 91–99.

[17] Iulian Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2015. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. In *Association for the Advancement of Artificial Intelligence*.

[18] Ravi Shekhar, Tim Baumgärtner, Aashish Venkatesh, Elia Bruni, Raffaella Bernardi, and Raquel Fernandez. 2018. Ask No More: Deciding when to guess in referential visual dialogue. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 1218–1233. https://www.aclweb.org/anthology/C18-1104

[19] Ravi Shekhar, Aashish Venkatesh, Tim Baumgärtner, Elia Bruni, Barbara Plank, Raffaella Bernardi, and Raquel Fernández. 2019. Beyond task success: A closer look at jointly learning to see, ask, and GuessWhat. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 2578–2587. https://doi.org/10.18653/v1/N19-1265

[20] Pushkar Shukla, Carlos Elmadjian, Richika Sharan, Vivek Kulkarni, Matthew Turk, and William Yang Wang. 2019. What Should I Ask? Using Conversationally Informative Rewards for Goal-oriented Visual Dialog.. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 6442–6451. https://doi.org/10.18653/v1/P19-1646

[21] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*.

[22] Florian Strub, Harm de Vries, Jérémie Mary, Bilal Piot, Aaron C. Courville, and Olivier Pietquin. 2017. End-to-end optimization of goal-driven and visually grounded dialogue systems. In *Joint Conference on Artificial Intelligence*.

[23] Florian Strub, Mathieu Seurin, Ethan Perez, Harm de Vries, Jérémie Mary, Philippe Preux, and Aaron CourvilleOlivier Pietquin. 2018. Visual reasoning with multi-hop feature modulation. In *Proceedings of the European Conference on Computer Vision*. 784–800.

[24] Jason D. Williams and Steve Young. 2007. Partially Observable Markov Decision Processes for Spoken Dialog Systems. *Comput. Speech Lang.* 21, 2 (April 2007), 393–422. https://doi.org/10.1016/j.csl.2006.06.008

[25] Tianhao Yang, Zheng-Jun Zha, and Hanwang Zhang. 2019. Making History Matter: History-Advantage Sequence Training for Visual Dialog. In *Proceedings of the IEEE International Conference on Computer Vision*. 2561–2569.

[26] Junjie Zhang, Qi Wu, Chunhua Shen, Jian Zhang, Jianfeng Lu, and Anton van den Hengel. 2018. Goal-Oriented Visual Question Generation via Intermediate Rewards. In *Proceedings of the European Conference on Computer Vision*.

[27] Rui Zhao and Volker Tresp. 2018. Learning goal-oriented visual dialog via tempered policy gradient. In *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 868–875.

[28] Bohan Zhuang, Qi Wu, Chunhua Shen, Ian D. Reid, and Anton van den Hengel. 2018. Parallel Attention: A Unified Framework for Visual Object Discovery Through Dialogs and Queries. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4252–4261.

3D Scene Geometry-Aware Constraint for Camera Localization with Deep Learning

Mi Tian[+], Qiong Nie[+], Hao Shen[*]

***Abstract*— Camera localization is a fundamental and key component of autonomous driving vehicles and mobile robots to localize themselves globally for further environment perception, path planning and motion control. Recently end-to-end approaches based on convolutional neural network have been much studied to achieve or even exceed 3D-geometry based traditional methods. In this work, we propose a compact network for absolute camera pose regression. Inspired from those traditional methods, a 3D scene geometry-aware constraint is also introduced by exploiting all available information including motion, depth and image contents. We add this constraint as a regularization term to our proposed network by defining a pixel-level photometric loss and an image-level structural similarity loss. To benchmark our method, different challenging scenes including indoor and outdoor environment are tested with our proposed approach and state-of-the-arts. And the experimental results demonstrate significant performance improvement of our method on both prediction accuracy and convergence efficiency.**

I. Introduction

Camera localization, as a foundation for many applications such as autonomous driving vehicle and mobile robots, estimates camera position and orientation from a query image and a pre-built map with scene information. In traditional localization framework, this scene information is generally presented as sparse key points with 3D information and feature descriptor. Camera poses are then estimated from 2D-3D matching between query images and a map by applying a Perspective-n-Point (PnP) solver accompanied with RANSAC [16, 38] strategies for outlier removal. Different methods are proposed to improve efficiency and effectiveness of such 2D-3D matching. For instance, image-level features like bag-of-words [32, 33], VLAD [34], Fish Vector [36, 37] are usually employed for similarity matching between query images and keyframes stored during mapping. Due to the image-level features retrieval results, matching area can be reduced into top N most similar keyframes and their surrounding points, which means that only a small 3D submap will participate in 2D-3D matching. As an intermediate step, these utilizing keyframes retrieval are categorized into retrieval-based approaches [3, 30, 31]. However direct approaches take advantages of different hashing algorithms to match 2D-3D points for computation acceleration. Specifically, bag-of-words [32, 33] and LSH [23] are two popular hashing methods for camera localization. Although many different efforts are made to improve 2D-3D matching accuracy, the fact that traditional approaches are based on low-level features such as SIFT [11, 1], SURF [25, 9], ORB [18], etc. makes it difficult to deal with challenging environments like illumination change or seasonal change.

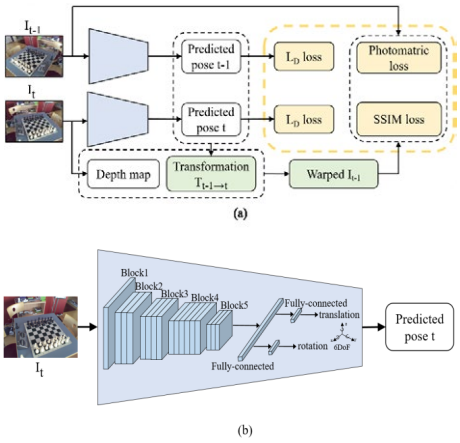


Figure 1: Schematic representation of our proposed self-supervised deep learning for camera localization with 3D scene geometry-aware constraint. (a): Training flow of our proposed algorithm which requires a pair of RGB images and a depth map of one of them. Green rectangles are computational components according to predicted poses and depth map without learnable parameters. Blue rectangles are networks for pose regression to be trained. And yellow rectangles are constraint terms of network. (b): Inference flow of camera pose localization. Blue part is network architecture based on the ResNet-50 that is a detailed description for the blue part in (a).

While learning-based methods aim to regress 6 DoF pose in an end-to-end way [5, 6, 17]. Scene information in this case is described as neural network weights and mapping step turns into a network training process. The first deep learning framework PoseNet [2] retrieves camera pose from a single image. [29] exploits temporal information for pose estimation by utilizing image sequence. [15] introduces the encoder-decoder architecture into camera localization. Some other changes like reasoning about the uncertainty of the estimated poses [22] are also proposed. However, all these methods train their networks by a naive Euclidean distance between prediction and ground truth pose. Inspired from traditional methods utilizing 3D geometry information, recently many geometry relevant loss functions such as geometric consistency error [7, 8], reprojection error [4], relative transform error [24] are built as regularization terms. Such

[+] indicates equal contributions. * indicates corresponding author. Email: ({tianmi02, nieqiong, shenhao04}@meituan.com).

This research is supported by Beijing Science and Technology Project (No. Z181100008918018). All authors are with Meituan-Dianping Group, Beijing, China.

methods perform better than those learned from single image information.

We follow prior works of learning-based camera localization and further search for more geometric information to constraint our model. In addition to standard sensors like GPS and camera that usually provide ground truth poses and images for localization, depth sensors are also very popular in SLAM applications. For indoor situation, we can directly obtain depth information from structured light camera, time-of-flight camera or stereo camera with available depth estimation algorithm. For outdoor environment, 3D LIDAR is usually employed for both localization and scene perception. From 3D geometry knowledge, when a general point in 3D scene is viewed in several images, their corresponding pixel intensities are supposed to be identical. This property we called as photometric consistency. It is the base idea for many direct visual odometry methods [3, 19] or SLAM methods [20, 32 - 35].

In this paper, we immigrate this idea into a neural network. The photometric consistency is described as a photometric loss term accompanied with a structured similarity SSIM [10] loss function to optimize pose regression with self-supervised learning. Meanwhile ground truth pose information and depth information (sparse or dense) from whatever depth sensors are used during training process only to calculate the photometric error loss. It bootstraps the loss function by penalizing pose predictions that contradict 3D scene geometry and helps the convergence of network. Although many traditional stereo methods and learning-based methods can estimate depth information, we prefer to use ground truth depth captured by robust sensors, considering easy availability of the sensor and information accuracy, and also our method does work even with very sparse depth information.

To this end, we make the following contributions compared to other works: (i) We propose a deep neural network architecture to directly estimate an absolute camera pose from an input image. (ii) By utilizing depth sensor information, we applied an additional 3D scene geometry-aware constraint to improve prediction accuracy. As mentioned, sparse depth information will be enough to get remarkable localization precision increment. This means that our method can be adapted with any kind of depth sensors (sparse or dense). (iii) We present extensive experimental evaluations on both indoor and outdoor datasets to compare our approach with state-of-the-art methods. At the same time, we demonstrate that the proposed additional 3D scene geometry-aware constraint can be easily added into other network and make performance improvement.

II. Related Work

Various CNN-based approaches of absolute camera localization have been proposed in the literature. In this section, some of the techniques developed thus far for improving the performance of localization will be discussed.

CNN-based camera localization was first proposed by PoseNet [2] which utilized base architecture of GoogLeNet to directly regress 6DoF camera pose with an input RGB image. By using Bayesian CNN, the authors extended their work to model precision uncertainty [22]. Following approaches, mainly differ in underlying base architecture and loss function used for training. Melekhov et al. [15] proposed Hourglass Network described as a symmetric encoder-decoder structure, which is widely used for applications of semantic segmentation. Rather than using a single image, Walch et al. [13] and Xue et al. [14] introduced Long-Short Term Memory (LSTM) to exploit global information by features learning from constraint of temporal smoothness of the video stream. Valada et al. [7, 8] proposed multitask learning framework for visual localization, odometry estimation and semantic segmentation. This method, which exploits inter-dependencies within multitask for the mutual benefit of each task, is considered as state-of-the-art since it provides higher localization precision than many other CNN-based approaches. However, such multitask training process requires much ground truth information, especially labeled semantic segmentation data causing this approach not flexible in many application domains.

Geometric consistency Constraint is recently used to help improving accuracy of pose regression and proved more effective than that of using Euclidean distance constraint alone. Valada et al. [7, 8] introduced geometric consistency to bootstrap loss function by penalizing pose predictions that contradict the relative motion. MapNet [24] imposed a constraint on relative pose between image pairs for global consistency. This method provided stricter constraints without any additional input information required as relative pose is easily computable by absolute ground truth pose. Kendall et al. [4] introduced another geometric loss named reprojection error defined as the residual of 3D points projected onto 2D image plane using the ground truth and predicted pose. All these works are considered to be state-of-the-art of that time using geometry consistency loss. In our work, we explore a 3D scene geometry-aware constraint called photometric error constraint. 3D structure information is added into this constraint which enforces network not only align predicted poses to camera motion but also aggregate scene structure model. Compared with the above image-level geometry consistency losses, our method makes use of geometry information of every 3D point of the scene and provides much stronger pixel-level constraint.

Photometric error constraint is typically used to deal with relative pose regression, optical flow estimation and depth prediction with supervised or unsupervised learning. For instance, Ma et al. [27] explored temporal relations of video sequences to provide additional photometric supervisions for depth completion network. Zhou et al. [12] built CNNs with unsupervised learning of dense depth and camera pose with photometric error loss to learn the scene level consistent movement governed by camera motion. Yin et al. [26] proposed a multitask unsupervised learning method of dense depth, optical flow and egomotion prediction, where photometric error constraint played an important role to enforce consistency between different tasks. Shen et al. [28] proposed to bridge the gap between geometric loss and photometric loss by introducing the matching loss constrained by epipolar geometry. Since photometric error constraint has been proved effective for relative pose regression and depth prediction, we introduce this photometric error constraint and validate its effectiveness on absolute pose prediction. As our knowledge, this is the first time that photometric error is imposed to solve absolute pose regression problem.

III. Proposed Approach

Our method is dedicated to absolute pose regression. The ground truth pose and depth information will be used during training process. Both information are easily available from sensors like GPS and depth sensors like RGBD cameras or LIDARs. At any inference time, only one image is imported to the network to localize the camera itself. In this section, we will introduce our pose regressing neural network as first. Then we will explain both training and inference framework in detail. At training process, three constraints are applied to help learning process towards a global minimum: a classic Euclidean error to measure distance from prediction to ground truth pose as well as two regularization terms formulated as a photometric loss and a structural similarity loss. Both regularizations try to lead model to obey photometric consistency but respectively by pixel-level and image-level. Finally, a warping process which is an intermediate step for building both terms is also presented.

A. Network architecture

We build a CNN architecture to predict the corresponding absolute pose p = [x, q] for a given image, where x denotes position and q denotes a unit of quaternion representing orientation. We use the first five residual blocks of ResNet-50 as backbone and modify it by introducing a global average pooling layer after the last residual block, and subsequently add three fully connected layers with 2048 neurons, 3 neurons and 4 neurons respectively. The last two fully connected layers separately output the absolute position x and orientation q (see Figure 1(b)). Each convolution layer is followed by batch normalization and Rectified Linear Unit (ReLU)

At inference process, only current image is applied to the network for regressing 6DoF pose directly (see Figure 1(b)). While during training (see Figure 1(a)), two successive images I_{t-1} and I_t as well as a depth map of I_{t-1} and the corresponding ground truth poses of I_{t-1} and I_t are required. The network learns weights and predicts absolute pose for both images by building Euclidean distant constraint as a loss term for each prediction. For a moving camera, two consecutive images are usually overlapped and their absolute poses can be mutually constrained by 3D scene geometry. In this paper, this 3D scene geometry-aware constraint is described as photometric error and SSIM error. Compared to [24] which just employs relative transform as geometry constraint to learn absolute pose, in our work, 3D scene geometry-aware constraint is employed as a pixel-level loss, exploiting more information including relative transform, 3D information and pixel intensity to learn camera localization with a global optimization directly and efficiently.

B. Warping computation

The warping computation from image I_{t-1} to I_t is illustrated in the following:

$$u_t = KT_{t-1}^{t} D_{t-1}(p_{t-1}) K^{-1} u_{t-1} \quad (1)$$

Where u_{t-1} is a static pixel in previous image I_{t-1}, its warped pixel to current time t is defined as u_t. We can easily get intrinsic matrix K by camera calibration. The 3D transform matrix from previous image to the current T_{t-1}^{t} can be computed according to their absolute poses T_w^{t-1} and T_w^{t}:

$$T_{t-1}^{t} = T_w^{t} * (T_w^{t-1})^{-1} \quad (2)$$

In warping computation, depth information $D_{t-1}(p_{t-1})$ is required for reconstructing 3D structure from 2D image pixels. As we explained in the previous section, dense depth information is not necessary. So we can extract it from depth sensors (structural light cameras, Time-of-flight cameras, stereo sensors and 3D LIDAR) or from stereo like depth computation algorithms, for example triangulation method of matched points from two overlapped images with knowing transform between them. However, to make sure not introducing extra depth error into our model. We prefer to choose robust depth information from a sensor.

To facilitate gradient computation for backpropagation, we create a synthetic image $warped_{t-1}$ with the same format of current image I_t by using bilinear interpolation as sampling mechanism for warping. As the warping is fully differentiable, we do not need any pre-computation for training and online running. Furthermore, no learnable weight or additional overhead is required for training and inference.

C. Loss function

In this section, constraint terms used for training network will be discussed in detail. In addition to typical Euclidean distant constraint, we introduce photometric loss term and structure similar loss term based on the warping results.

Euclidean distant constraint Since we input two successive images into the model in parallel during training, the Euclidean distant losses for both images are calculated as:

$$L_D = L_D(I_{t-1}) + L_D(I_t) \quad (3)$$

with

$$L_D(I_i) = \|x_i - \hat{x}_i\|_2 + \beta \|q_i - \hat{q}_i\|_2 \quad for\ i \in \{t-1, t\} \quad (4)$$

Where x_i, q_i are the ground truth position and orientation, $\hat{x}_i$, $\hat{q}_i$ are the predicted position and orientation, and β is a weighted parameter to keep the expected values of position and orientation errors to be nearly equal and to be trained online. This highly strong supervision signal leads pose prediction converge to the approximate ground truth.

Photometric error constraint When there is limited change of viewpoint and the environment is assumed to be light-invariant, the intensity values of a 3D point in different images are supposed to be the same. This photometric consistency is used for solving many problems (both traditional solution and learning-based solutions) like optical flow estimation, depth estimation, visual odometry, etc. Here, we employ it for absolute pose estimation. Here the loss function is designed as the difference between the $warped_{t-1}$ image and current image I_t:

$$L_P = \sum_{i,j} M\left(u_{t-1}^{i,j}\right) \|I_t(i,j) - warped_{t-1}(i,j)\|_1 \quad (5)$$

Where $u_{t-1}^{i,j}$ is the pixel with coordinate (i,j) in image I_{t-1}, $M\left(u_{t-1}^{i,j}\right)$ is an image mask. The idea is to mask pixels without depth information and that do not obey photometric consistency. In our case, we mainly use it to mask two types of pixels: moving pixels and pixels with invalid depth information. The depth validity depends on the acquisition methods. For instance, depth from range sensors like LIDAR usually has satisfactory accuracy even at a long distance, but depth from computation algorithms like stereo-like method is much noisy. And many strategies can be applied to remove dynamic objects. For example, as long as moving objects are

Table1: Comparison of median localization error with existing CNN-based models on 7-Scene dataset

Scene	Spatial extent	PoseNet [2]	LSTM-Pose [13]	VidLoc [29]	Hourglass Pose[15]	PoseNet2 [4]	MapNet [24]	Ours
Chess	3×2×1 m^3	0.32, 8.12°	0.24, 5.77°	0.18, N/A	0.15, 6.53°	0.13, 4.48°	0.08, 3.25°	0.09, 4.39°
Fire	2.5×1×1 m^3	0.47, 14.4°	0.34, 11.9°	0.26, N/A	0.27, 10.84°	0.27, 11.3°	0.27, 11.69°	0.25, 10.79°
Heads	2×0.5×1 m^3	0.29, 12.0°	0.21, 13.7°	0.14, N/A	0.19, 11.63°	0.17, 13.0°	0.18, 13.25°	0.14, 12.56°
Office	2.5×2×1.5 m^3	0.48, 7.68°	0.30, 8.08°	0.26, N/A	0.21, 8.48°	0.19, 5.55°	0.17, 5.15°	0.17, 6.46°
Pumpkin	2.5×2×1 m^3	0.47, 8.42°	0.33, 7.00°	0.36, N/A	0.25, 7.01°	0.26, 4.75°	0.22, 4.02°	0.19, 5.91°
RedKitchen	4×3×1.5 m^3	0.59, 8.64°	0.37, 8.83°	0.31, N/A	0.27, 10.15°	0.23, 5.35°	0.23, 4.93°	0.21, 6.71°
Stairs	2.5×2×1.5 m^3	0.47, 13.8°	0.40, 13.7°	0.26, N/A	0.29, 12.46°	0.35, 12.4°	0.30, 12.08°	0.26, 11.51°

usually vehicles or persons, object detection can be applied in advance to remove these moving objects. Moreover, we can also ignore the pixels with large photometric errors since these pixels are suspectable to violate the consistency principle.

Minimizing the photometric error takes effect only when the warped pixel is very close to the true correspondence. It requires predicted pose not far from ground truth. At the early epochs of training, Euclidean loss determines the gradient direction dominantly as current predicted pose is very different from ground truth and therefore photometric loss produces only a weak or even bad effects. To this end, we propose a self-adaption strategy: a photometric error is used for back-propagation only when the projection point u_{t-1} and u_t satisfies $\|u_t - u_{t-1}\|_1 \leq h$ (h is a threshold value that highly depends on scenes, in our case, h is set as 10). The purpose is to maximize the value of photometric loss for optimizing pose prediction.

Structural similarity constraint This constraint tries to extract structural information from scene, like the way of human visual system. The similarity of two images I_x and I_y is formulated as:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \quad (6)$$

Where C_1 and C_2 are constant to keep SSIM valid. The SSIM value is [0,1] and high similarity corresponds to a big value. An auxiliary constraint that differs $warped_{t-1}$ image and current image I_t is defined by equation (7) in combination with photometric error.

$$L_S = \frac{1-SSIM(I_t,\ warped_{t-1})}{2} \quad (7)$$

The final loss function is defined in formulate (8). It contains three loss terms with different weighted parameters namely λ_D, λ_P, λ_S to balance every loss term, and constrain the weights update together. Both Euclidean distant loss term and SSIM loss term are image-level constraints, while photometric loss term belongs to a pixel-level constraint, which can lead to a more precise accuracy of prediction.

$$L = \lambda_D L_D + \lambda_P L_P + \lambda_S L_S \quad (8)$$

IV. Experiment Evaluation

In this section, we will present experimental results of our proposed method for camera localization in comparison with several state-of-the-art works both on indoor and outdoor datasets. The results demonstrate that our introduced loss terms as well as self-supervised strategy for absolute camera localization task are outstanding in prediction accuracy as well as training convergence.

A. Datasets

Our method is evaluated on a well-known public dataset – Microsoft 7-Scene which is a collection of tracked RGB-D camera frames [21]. Seven different scenes recorded from a handheld Kinect RGB-D camera at 640×480 resolution are proposed for evaluation. The dense depth map is directly obtained from RGB-D sensors and ground truth camera pose is provided by KinectFusion algorithm. The existence of motion blur and weak texture under office environment makes this 7-scene dataset very challenging and widely evaluated by localization and tracking algorithms. To facilitate comparison, we take the same training and testing sequence split of each scene as other methods did.

Oxford robotcar dataset [39] contains 100 repetitions of a consistent route through central oxford captured twice a week over a period of over a year. Different types of data are available from multiple sensors including monocular cameras, LIDAR, GPS, INS measurements as well as stereo cameras. We take sub-dataset LOOP with a total length of 1120m for our evaluation. Two subsets overlapping the whole path with the same motion direction are used for training and test respectively.

B. Implementation details

Since [7, 8] demonstrate that neither synthetic pose augmentation nor synthetic view augmentation techniques yield any performance gain. In some cases, they have even negative impacts on pose accuracy. In our experiments, we only take proven well-performed preprocessing steps like resize of input images into 320×240 and normalization.

We use the Adam solver for optimization with $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\varepsilon = 10^{-10}$. We initialize five residual blocks with weights of ResNet-50 pre-trained on ImageNet and remaining layers with Gaussian distribution, then fine-tuning all layers with mini-batch size of 12 and maximum iterations of 50 epochs. We apply layers-wise learning rate set that is initialized as 8e-4 and 2e-4 for five residual blocks and remaining layers respectively. Polynomial decay for learning rate is adopt with power = 0.9. The weighted parameters β, λ_D, λ_P, λ_S are set as 3, 1, 0.01, 0.1 on all scenes. The work is implemented based on Tensorflow deep learning library and all the experiments are performed on a NVIDIA Titan V GPU with 16GB on-board memory.

C. Comparison with prior methods

Our regression method is tested on all scenes of 7-Scene dataset to compare with prior CNN-based methods namely

PoseNet [2], LSTM-Pose [13], VidLoc [29], Hourglass-Pose [15], PoseNet2 [4] and MapNet [24]. Table 1 shows the quantitative comparisons of median translation and rotation errors for each scene in the datasets. Except that MapNet slightly outperforms on chess scene, our method obtains better results on most scenes. Moreover, compared to MapNet [24] that needs 300 epochs and PoseNet [2] needs more, our method takes only 50 epochs iterations to convergence.

To illustrate our results in detail, several camera pose trajectories on test sequences of heads, fire, pumpkin and stairs scenes are shown in Figure 2. It is obvious that trajectories provided by PoseNet [2] are much noised and even fail sharply in some places. MapNet [24] has a stable prediction globally but the accuracy is unsatisfactory. In this experiment, our method achieves mostly outstanding performances both on translation and rotation accuracy. From above experiments, our proposed network architecture collaborated with introduced photometric error loss term exhibits much better performances considering accuracy-efficiency balance.

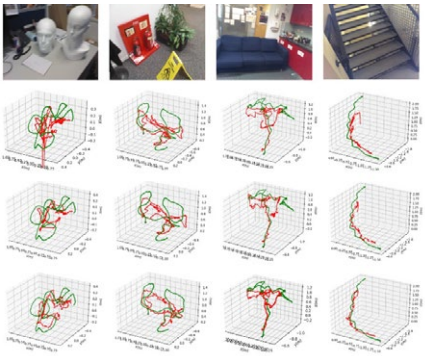


Figure 2: camera localization results on Microsoft 7-Scene. From left to right, the four test sequences are heads-01 sequence, fire-04 sequence, pumpkin-01 sequence and stairs-01 sequence. From top to bottom, the three results are from PoseNet [2], MapNet [24] and our method respectively (green for the ground truth, red for the prediction).

D. Ablation studies

In this section, the performance of our proposed geometric constraint for the absolute camera localization task will be studied. To this end, we employ ablation experiments that we train different network architectures including GoogLeNet of PoseNet and ours with the help of photometric error loss and SSIM loss and then compare them with that without geometric constraint. From the quantitative results shown in Table 2, we can on one hand demonstrate that such 3D scene geometry-aware constraint described by a photometric loss and a SSIM loss is always helpful as it leads to a better performance on prediction accuracy for all scenes. On the other hand, this improvement from 3D scene geometry-aware constraint is applicable to different network architectures. And theoretically we can employ it to any other camera localization networks to help learning process converge towards global minimization during training. Besides, even with one Euclidean loss alone, the results prove that our proposed method performs better than PoseNet2 [4] optimized by geometric loss.

Table 2: comparison of median localization error with different network and loss terms of network on 7-Scene dataset

Network	PoseNet [2]		Ours	
Loss	L_D	$L_D + L_P + L_S$	L_D	$L_D + L_P + L_S$
Chess	0.32, 8.12°	0.11, 5.11°	0.10, 5.38°	0.09, 4.39°
Fire	0.47, 14.4°	0.24, 11.0°	0.26, 13.3°	0.25, 10.79°
Heads	0.29, 12.0°	0.16, 11.8°	0.16, 12.6°	0.14, 12.56°
Office	0.48, 7.68°	0.20, 8.11°	0.22, 8.07°	0.17, 6.46°
Pumpkin	0.47, 8.42°	0.18, 4.83°	0.22 6.80°	0.19, 5.91°
RedKitchen	0.59, 8.64°	0.24, 7.19°	0.23, 8.53°	0.21, 6.71°
Stairs	0.47, 13.8°	0.29, 10.2°	0.30, 11.5°	0.26, 11.51°

E. Influence of depth sparsity

In indoor 7-scene dataset, dense depth maps are available directly from depth sensor. While in an outdoor environment depth information from other type of depth sensors like LIDAR or depth computation algorithms like stereo-like methods is usually sparse. Therefore, we discuss the influence of depth sparsity on our method and show that the proposed approach still works well even with a sparsity of only 20% depth information. We evaluate this property on 7-scene dataset and the results are shown in Table 3. The original depth map generated from Kinect sensor are assumed as 100% depth information. We randomly eliminate 40% of depth and 80% of depth respectively from the initial map, and then test our method using the remaining depth information without changing other network settings.

Table3: comparison of median localization error with different levels of depth sparsity

scene	20%-depth	60%-depth	100%-depth
Chess	0.10, 5.01°	0.10, 4.76°	0.09, 4.39°
Fire	0.25, 12.81°	0.25, 12.38°	0.25, 10.79°
Heads	0.16, 13.31°	0.16, 13.45°	0.14, 12.56°
Office	0.19, 7.79°	0.17, 6.62°	0.17, 6.46°
Pumpkin	0.21, 4.74°	0.20, 4.84°	0.19, 5.91°
RedKitchen	0.23, 10.76°	0.22, 10.18°	0.21, 6.71°
Stairs	0.29, 12.17°	0.28, 12.86°	0.26, 11.51°

Apparently, more depth information means more constraints that will evidently lead to a more precise prediction accuracy. But our method slightly outperforms event with a sparsity of 20% depth information compared to other methods illustrated in Table 1. In summary, our method can collaborate with different kinds of depth sensors or any well-defined depth computation algorithm. and provide more accurate absolute camera pose estimation.

F. Self-supervised learning

Different from [27], we apply self-supervised learning strategy for photometric error and SSIM loss terms at training process. This means that absolute poses of image I_{t-1} and I_t used for building photometric consistency constraint are both predicted by network (see Figure 1(a)). To compared it, we change the pose of image I_t directly from ground truth and use it to compute relative transform between two images for further warping. From the results shown in Table 4, self-supervised learning strategy outperforms both on rotation and translation accuracy. This is partly because we take more advantages of data at training process with self-supervised learning by back-propagating it twice when it is considered as I_{t-1} and also when we treat it as I_t. Furthermore, it helps network to learn in a more natural way because camera poses are never independent to each other and for more overlapped images their corresponding poses are highly relevant by the nature of 3D geometry.

Table 4: comparison of median localization error with different learning strategy

scene	Self-supervised learning	
	w/o	w
Chess	0.10, 5.26°	0.09, 4.39°
Fire	0.26, 11.5°	0.25, 10.79°
Heads	0.16, 13.3°	0.14, 12.56°
Office	0.18, 7.28°	0.17, 6.46°
Pumpkin	0.25, 6.34°	0.19, 5.91°
RedKitchen	0.23, 7.53°	0.21, 6.71°
Stairs	0.29, 12.8°	0.26, 11.51°

G. Outdoor evaluation on Oxford Robotcar Dataset

Our method is also tested in an outdoor dataset Oxford robotcar. Although training subset 2014-05-14-13-59-05 and test subset 2014-05-14-13-53-47 are both captured on the same day, large illumination change between two sequences and motion blur make it very challenging for camera localization.

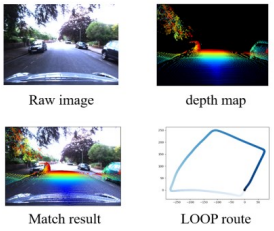


Figure 3: Oxford robotcar dataset raw color image and depth map captured by LIDAR. LOOP route subset with a total length of 1120m.

We firstly align LIDAR with a frontal camera to obtain a sparse depth map for image (see Figure 3). To avoid introducing too much depth error to our system, we choose only nearby 3D points that are less than 20m viewed from camera. The results show that our method significantly outperforms PoseNet (see Table 5) that is learned on training subset using the same network and hyper-parameters setting as [2]. Even utilizing Euclidean distant loss term alone, our method shows an accuracy increase of 15%. After introducing proposed 3D scene geometry-aware constraint, our approach provides an accuracy increase of more than 36% compared to the baseline PoseNet. To be noted that current depth map has a sparsity of less than 5% and it is also suffered from alignment noise.

To sum up, our method is not sensible to environments and it provides an apparent accuracy improvement even with highly sparse depth information. All these properties make the proposed approach suitable for many applications including indoor robots and outdoor autonomous vehicles.

Table 5: comparison of median localization error with different algorithm

Test subset	PoseNet [2]	Ours (L_D)	Ours (L_D+L_P+L_S)
2014-05-14-13-53-47	25.59, 15.96°	22.09, 10.60°	16.28, 7.17°

V. Conclusion

In this paper, we present a novel absolute camera localization algorithm. Rather than building a map whose size is linearly proportional to the scene size, we train a neural network to describe the scene. At the same time, we impose a novel 3D scene geometry-aware constraint as loss terms to supervise the network training. We believe that such network is more representative about 3D scene, motion and image information. The experimental results also show that our method outperforms prior works. Besides, our comparison results illustrate that positive impact is achieved when this 3D scene geometry-aware constraint is added into different networks. Therefore, we believe the effectiveness of this constraint in absolute camera localization algorithms. Last but not the least, our method is suitable for many applications like indoor mobile robots or outdoor autonomous driving vehicles. On these platforms, training data is directly available from different sensors and no additional manual annotation is required. In future work, we aim to pursue further fusion between CNN-based methods and traditional metric-based methods for camera localization. And an integration of different sensor modalities may also improve camera localization.

References

[1] Ke, Yan and R. Sukthankar. “PCA-SIFT: a more distinctive representation for local image descriptors.” Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2004.

[2] A. Kendall, M. Grimes, and R. Cipolla, “Posenet: A convolutional network for real-time 6-dof camera relocalization,” in ICCV, 2015.

[3] I. Melekhov, J. Kannala, and E. Rahtu, “Relative camera pose estimation using convolutional neural networks,” arXiv:1702.01381, 2017.

[4] A. Kendall and R. Cipolla, “Geometric loss functions for camera pose regression with deep learning,” CVPR, 2017.

[5] J. Wu, L. Ma, and X. Hu, “Delving deeper into convolutional neural networks for camera relocalization,” in ICRA, May 2017.

[6] T. Naseer and W. Burgard, “Deep regression for monocular camera-based 6-dof global localization in outdoor environments,” in IROS, 2017.

[7] A. Valada, N. Radwan, and W. Burgard, “Deep auxiliary learning for visual localization and odometry,” in ICRA, 2018.

[8] N. Radwan, A. Valada, W. Burgard, “VLocNet++: Deep Multitask Learning for Semantic Visual Localization and Odometry”, IEEE Robotics and Automation Letters (RA-L), 3(4): 4407-4414, 2018.
[9] Bay, Herbert, et al. “Speeded-up robust features (SURF).” Computer vision and image understanding 110.3 (2008): 346-359.
[10] Wang, Zhou, et al. “Image quality assessment: from error visibility to structural similarity.” IEEE transactions on image processing 13.4 (2004): 600-612.
[11] Ng, Pauline C., and Steven Henikoff. “SIFT: Predicting amino acid changes that affect protein function.” Nucleic acids research 31.13 (2003): 3812-3814.
[12] Zhou, Tinghui, et al. “Unsupervised Learning of Depth and Ego-Motion from Video.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.
[13] F. Walch, C. Hazirbas, et al., “Image-based localization using lstms for structured feature correlation,” in ICCV, 2017.
[14] Xue, Fei, et al. “Beyond Tracking: Selecting Memory and Refining Poses for Deep Visual Odometry.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019
[15] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, “Image-based localization using hourglass networks,” arXiv:1703.07971, 2017.
[16] Brachmann, Eric, et al. “DSAC-differentiable RANSAC for camera localization.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[17] Brachmann, Eric, and Carsten Rother. “Learning less is more-6d camera localization via 3d surface regression.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.
[18] Rublee, Ethan, et al. “ORB: An efficient alternative to SIFT or SURF.” ICCV. Vol. 11. No. 1. 2011.
[19] K. R. Konda and R. Memisevic, “Learning visual odometry with a convolutional network,” in VISAPP, 2015.
[20] Engel, Jakob, Thomas Schöps, and Daniel Cremers. “LSD-SLAM: Large-scale direct monocular SLAM.” European conference on computer vision. Springer, Cham, 2014.
[21] Shotton Jamie, et al. “Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013.
[22] A. Kendall and R. Cipolla, “Modelling uncertainty in deep learning for camera relocalization,” ICRA, 2016.
[23] Andoni, Alexandr, et al. “Practical and optimal LSH for angular distance.” Advances in Neural Information Processing Systems. 2015.
[24] Brahmbhatt, Samarth, et al. “Geometry-aware learning of maps for camera localization.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.
[25] Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. “Surf: Speeded up robust features.” European conference on computer vision. Springer, Berlin, Heidelberg, 2006.
[26] Yin, Zhichao, and Jianping Shi. “Geonet: Unsupervised learning of dense depth, optical flow and camera pose.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.
[27] Ma, Fangchang, Guilherme Venturelli Cavalheiro, and Sertac Karaman. “Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera.” 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019.
[28] Shen, Tianwei, et al. “Beyond Photometric Loss for Self-Supervised Ego-Motion Estimation.” 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019.
[29] Clark, Ronald, et al. “Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.
[30] Wolf, Jürgen, Wolfram Burgard, and Hans Burkhardt. "Robust vision-based localization by combining an image-retrieval system with Monte Carlo localization." IEEE Transactions on Robotics 21.2 (2005): 208-216.
[31] Sattler, Torsten, et al. "Image Retrieval for Image-Based Localization Revisited." BMVC. Vol. 1. No. 2. 2012.
[32] Chai, Zheng, and Takafumi Matsumaru. "ORB-SHOT SLAM: Trajectory Correction by 3D Loop Closing Based on Bag-of-Visual-Words (BoVW) Model for RGB-D Visual SLAM." Journal of Robotics and Mechatronics 29.2 (2017): 365-380.
[33] Jiachen, Zhang, et al. "Bag-of-words based loop-closure detection in visual SLAM." Advanced Optical Imaging Technologies. Vol. 10816. International Society for Optics and Photonics, 2018.
[34] Huang, Yao, Fuchun Sun, and Yao Guo. "VLAD-based loop closure detection for monocular SLAM." 2016 IEEE International Conference on Information and Automation (ICIA). IEEE, 2016.
[35] Yousif, Khalid, Yuichi Taguchi, and Srikumar Ramalingam. "MonoRGBD-SLAM: Simultaneous localization and mapping using both monocular and RGBD cameras." 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017.
[36] Perronnin, Florent, and Christopher Dance. "Fisher kernels on visual vocabularies for image categorization." 2007 IEEE conference on computer vision and pattern recognition. IEEE, 2007.
[37] Perronnin, Florent, Jorge Sánchez, and Thomas Mensink. "Improving the fisher kernel for large-scale image classification." European conference on computer vision. Springer, Berlin, Heidelberg, 2010.
[38] Li, Hao, et al. "An efficient image matching algorithm based on adaptive threshold and RANSAC." IEEE Access 6 (2018): 66963-66971.
[39] W. Maddern, G. Pascoe, C. Linegar and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset", The International Journal of Robotics Research (IJRR), 2016.

Robust Trajectory Forecasting for Multiple Intelligent Agents in Dynamic Scene

Yanliang Zhu[1] , **Dongchun Ren**[1] , **Mingyu Fan**[1,2] , **Deheng Qian**[1] , **Xin Li**[1] , **Huaxia Xia**[1]

[1]Meituan-Dianping Group, Beijing 100102 China
[2]School of Computer Science, Wenzhou university, Wenzhou 325035, China
{zhuyanliang, rendongchun}@meituan.com, fanmingyu@wzu.edu.cn, {qiandeheng, lixin125, xiahuaxia}@meituan.com

Abstract

Trajectory forecasting, or trajectory prediction, of multiple interacting agents in dynamic scenes, is an important problem for many applications, such as robotic systems and autonomous driving. The problem is a great challenge because of the complex interactions among the agents and their interactions with the surrounding scenes. In this paper, we present a novel method for the robust trajectory forecasting of multiple intelligent agents in dynamic scenes. The proposed method consists of three major interrelated components: an interaction net for global spatiotemporal interactive feature extraction, an environment net for decoding dynamic scenes (i.e., the surrounding road topology of an agent), and a prediction net that combines the spatiotemporal feature, the scene feature, the past trajectories of agents and some random noise for the robust trajectory prediction of agents. Experiments on pedestrian-walking and vehicle-pedestrian heterogeneous datasets demonstrate that the proposed method outperforms the state-of-the-art prediction methods in terms of prediction accuracy.

1 Introduction

Trajectory prediction of agents in dynamic scene is a challenging and essential task in many fields, such as social-aware robotic systems [Van den Berg *et al.*, 2011], autonomous driving [Ma *et al.*, 2019b] and behavior understanding [Liang *et al.*, 2019]. Intelligent agents, such as humans, vehicles, and independent robots, are supposed to be able to understand and forecast the movement of the others to avoid collisions and make smarter movement plans. Trajectory prediction has been studied extensively. Traditional prediction methods, such as the Gaussian process regression [Rasmussen and Williams, 2005], the kinematic and dynamic method [Toledo-Moreo and Zamora-Izquierdo, 2009] and the Bayesian networks method [Lefèvre *et al.*, 2011], ignore the interactions among the agents and are only able to make short term predictions. Recently, Recurrent Neural Network (RNN) and its variants [Alahi *et al.*, 2016], such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), have shown promising ability in capturing the dynamic interactions of agents and a great number of trajectory prediction methods have been proposed based on them.

However, trajectory prediction is still a challenging task because of the several properties of it: 1) When intelligent agents move in public, they often interact with other agents such as human or obstacles in the scene, which is named as the **social behavior**. Actions, including collision avoidance and moving in groups, require the ability to forecast the possible movements or actions of the other agents. The social interactions may not be confined to nearby agents or obstacles. 2) The movement of agents is not only dependent on the nearby agents, but is also influenced by the surrounding physical scene, i.e., the **dynamic scene**. One important factor of the scene is the road topology, such as intersections, turns, and slip lanes. Certain road topology can significantly influence the speed and direction of the moving agents. An autonomous agent should be always moving on a feasible terrain. 3) The **multi-modal motion** property illustrates that the interactive agents may follow several viable trajectories as there is a rich choice of reasonable movements. When two independent agents move toward each other, there are many possible different future trajectories that could avoid collision, such as moving to the left, to the right, or stop.

In this study, we propose a novel robust trajectory forecasting method for multiple intelligent agents in dynamic scene. The main contributions of this paper are summarized as follows.

- We model the global spatio-temporal interaction through an interaction net with a soft agent-tracking module. The interaction net not only considers the current locations and interaction of the agents, but also the temporal interactions among the agents by the hidden states of LSTMs on past trajectories.
- An environmental net is introduced to encode the dynamic scene. The surrounding road topology, such as interactions, turns and slip lanes, is firstly transformed into an high-definition map and then the map is encoded by a pre-trained convolutional neural network.
- Our trajectory prediction net combines the feature of spatio-temporal interaction, the environment feature, and the past trajectory to forecast the future trajectory of all the agent. Attention model is used to adaptively encode the spatio-temporal interaction of an agent with

the others.

The rest of this paper is structured as follows: in Section 2, some related work is reviewed. The proposed robust trajectory prediction method is introduced in Section 3. Experimental comparisons with the state-of-the-art trajectory prediction methods on benchmark datasets are presented in Section 4. Finally, the conclusion is drawn in Section 5.

2 Related Work

2.1 RNN networks and trajectory prediction

Recurrent neural networks (RNN) and its variants, such as LSTM and GRU, have been shown very successful in many sequence forecasting tasks [Chung *et al.*, 2014]. Therefore, many researches focusing on using RNN and its variants for trajectory prediction. A simple and scalable RNN architecture for human motion prediction is proposed by [Martinez *et al.*, 2017]. The CIDNN method [Xu *et al.*, 2018] uses the inner product of the motion features, which are obtained by LSTMs, to encode the interactions among agents and feeds the interaction features into a multi-layer perceptron for prediction. By using separate LSTMs for heterogeneous agents on road, the VP-LSTM method [Bi *et al.*, 2019] is designed to learn and predict the trajectories of both pedestrians and vehicles simultaneously. In [Choi and Dariush, 2019], a relation gate module is proposed to replace the LSTM unit for capturing a more descriptive spatio-temporal interactions, and both human-human and human-scene interactions from local and global scales are used for future trajectory forecast. These studies indicate that RNN alone is unable to handle complex scenarios, such as interactions, physical scene and road topology. Additional structure and operations are always required for accurate, robust and long term prediction.

2.2 Social behaviors and interactions

Based on handcraft rules and functions, the social force models [Helbing and Molnár, 1995; Pellegrini *et al.*, 2010] use attractive and repulsive forces to describe the interactions of pedestrians in crowd. However, the handcraft rules and functions are unable to generalize for complex interaction scenarios. Instead of handcraft parameters, recent methods use RNN and its variants to learn the parameters directly from data. Social-LSTM [Alahi *et al.*, 2016] proposes a social pooling layer to model interactions among nearby agents, where the pooling layer uses LSTMs to encode and decode the trajectories. In [Su *et al.*, 2017], the method uses LSTMs with social-aware recurrent Gaussian processes to model the complex transitions and uncertainties of agents in crowd. The SoPhie method [Sadeghian *et al.*, 2019] uses the information from both the physical scene context and the social interactions among the agents for prediction. The TraPHic method [Chandra *et al.*, 2019] proposes to use both the horizon-based and the heterogeneous-based weights to describe interactions among road agents. A Generative Adversarial Network (GAN) is applied in the social-ways method [Amirian *et al.*, 2019] to derive plausible future trajectories of agents, where both generator and discriminator networks of the GAN are built by LSTMs.

2.3 Graph models for trajectory prediction

Many previous studies formulate interactions of agents as graphs, where nodes refers to the agents, and edges are used to represent the pairwise interactions. Edge weights are used to quantify the importances of the agents to each other. The social-BiGAT method [Kosaraju *et al.*, 2019] proposes a graph attention network to encode the interactions among humans in a scene and a recurrent encoder-decoder architecture to predict the trajectory. A dynamic graph-structured model for multimodal trajectories prediction, which is named as the Trajectron, is proposed in [Ivanovic and Pavone, 2019]. Constructed on a 4-D graph, the TrafficPredict method [Ma *et al.*, 2019a] consists of two main layers, an instance layer to learn interactions and a category layer to learn the similarities of instance of the same type. TrafficPredict has shown promising results for trajectory prediction of heterogeneous road agents such as bicycles, vehicles and pedestrians. The STGAT method [Huang *et al.*, 2019] first uses an LSTM to capture the trajectory information of each agent and applies the graph attention network to model interactions in agents at every time step. Then STGAT adopts another LSTM to learn the temporal correlations for interactions explicitly.

3 The Proposed Method

3.1 Problem Formulation

In this study, we consider two types of mobile agents: the ego-agent and the other agents. The spatial coordinates of all agents from time step 1 to T_{obs} are given to predict their future locations from time step T_{obs+1} to T_{pred}. The general formulation of trajectory prediction is represented as,

$$\mathbf{Prediction}_{\{\theta\}} : \left\{\{X_i\}_{i=1}^{N}, X_{ego}, Y_{ego}\right\} \longmapsto \{Y_i\}_{i=1}^{N},$$

where X_i and Y_i denote the past and future trajectories of the i-th agent, respectively, X_{ego} and Y_{ego} stand for the trajectories of the ego-agent, and θ denotes the model parameters. Different from previous studies, we consider the prediction problem on the real autonomous driving system where the planned trajectory for the ego-agent Y_{ego} is given for reference. The planned trajectory can improve the prediction accuracy because it brings some prior knowledge on the future. Specifically, either an observed or a future trajectory can be expressed as a set of temporal coordinates $X_i(\text{or } \mathrm{Y_i}) = \left\{\left(\mathrm{x_i^{(1)}}, \mathrm{y_i^{(1)}}\right), \left(\mathrm{x_i^{(2)}}, \mathrm{y_i^{(2)}}\right), \cdots, \left(\mathrm{x_i^{(t)}}, \mathrm{y_i^{(t)}}\right)\right\}$. We use $p_i^{(t)} = \left(x_i^{(t)}, y_i^{(t)}\right)^T$ to denote the location of the i-th agent at time step t, and set the id of the ego-agent be 0.

As is shown in Fig. 1, our proposed approach consists of three interrelated components: an interaction net for spatio-temporal interactive feature extraction, an environment exploration network for decoding dynamic physical scene (i.e., the surrounding road topology), and a trajectory prediction network. Each component and the implementation details of the proposed method is described in details as follows.

3.2 Interaction Net

The Agent Interaction Network (AIN) is designed to encode the interaction feature among all the agents in the dynamic

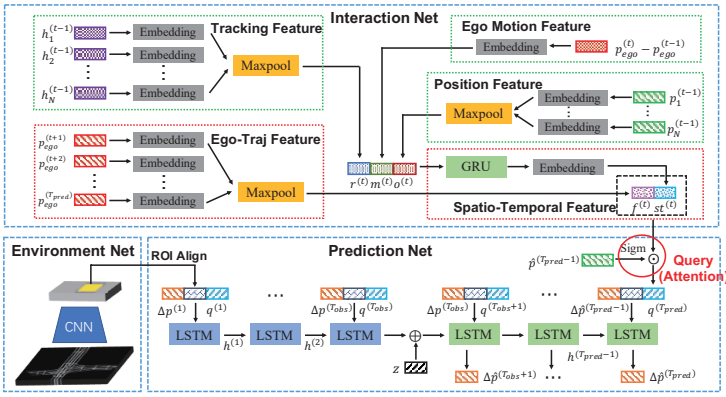


Figure 1: Overview of the proposed method. The proposed method contains three components, a spatio-temporal interaction network, an environment feature extraction network, and a trajectory prediction network.

scenario. As opposed to the pairwise interactive feature by attention models in previous studies, our method is able to capture the collective influence among the agents. Besides, our method could consider the future movement of the ego-agent for reference. The AIN takes three information sources of all agents as input: the past trajectories, the hidden states of LSTMs and the planned trajectory of the ego-agent. Given these data, AIN computes the global spatio-temporal inter-agents interactions and the future ego-others interaction.

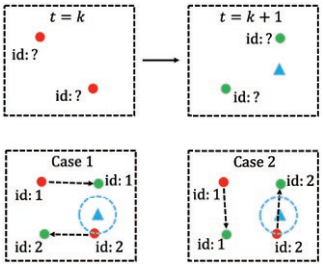


Figure 2: An example to show global interaction with or without tracking information. The top subfigures present the locations of two agents at adjacent time steps without tracking information. The bottom subfigures show the locations of two agents at adjacent time steps with tracking information.

Global spatio-temporal inter-agents interaction

The past trajectories of all agents contain the latent patterns of the interactive movement in dynamic scenario. In this module, we intend to learn the latent patterns through a neural network. The learned latent feature represents the global spatio-temporal interaction of all agents on road.

Given the locations of all agents at a time step t, we utilize the linear and maxpooling functions to produce the global position feature of size $1 \times d_o$, which is given as below:

$$e_{o,i}^{(t)} = W_o p_i^{(t)} + b_o, \quad \forall i \in \{0, \ldots, N\}, \quad (1)$$

$$o^{(t)} = \mathbf{Maxpool}\left(\mathbf{Cat}\left(\left[e_{o,0}^{(t)\,T}, \cdots, e_{o,N}^{(t)\,T}\right], 1\right)\right), \quad (2)$$

where $W_o \in \mathbb{R}^{d_o \times 2}$ and $b_o \in \mathbb{R}^{d_o}$ are the weight matrix and bias of the embedding layer. $\mathbf{Cat}\left([\cdot], 1\right)$ denotes the concatenation function which joints all the inputs along the first dimension. The $\mathbf{Maxpool}\left(\cdot\right)$ function squeezes the spliced data along the same dimension, i.e., the batch dimension.

Moreover, a key problem for the position feature given in Eq. (2) is the temporal issue. Process without temporal information ignores the past interaction and may lead to performance drop. As can be seen in top subfigures of Fig. 2, the locations of two agents (two circles) at two adjacent time steps are shown. Without tracking information (the agent id), it is impossible to know which agent and how the agent interacts the blue triangle in the time span. There are two different possible motion behavior of the agents from time step k to $k+1$, as is shown in the bottom subfigures of Fig. 2. In case 1, both the two agents could interact with the blue triangle. And in case 2, the blue triangle is more likely to interact with the agent 2.

To address the temporal issue, we use the hidden states of the LSTMs in the prediction network to track the locations of all agents. The global tracking feature $r^{(t)} \in \mathbb{R}^{1 \times d_r}$ is obtained as follows,

$$e_{r,i}^{(t)} = W_r h_i^{(t)} + b_r, \quad \forall i \in \{0, \ldots, N\}, \quad (3)$$

$$r^{(t)} = \mathbf{Maxpool}\left(\mathbf{Cat}\left(\left[e_{r,0}^{(t)\,T}, \cdots, e_{r,N}^{(t)\,T}\right], 1\right)\right), \quad (4)$$

where W_r and b_r are the layer parameters, and $h_i^{(t)} = h_{e,i}^{(t)}$ when $t \leq T_{obs}$, $h_i^{(t)} = h_{h,i}^{(t)}$ when $t \geq T_{obs}+1$.

On real autonomous driving system, one is given the planned trajectory of ego-agent to address the issue of coordinate system transformation (from the world coordinate system to relative coordinate system where the ego-agent centers the origin). From ego-perspective, the global inter-agents interaction module can be mathematically expressed as below:

$$m^{(t)} = W_m \left(p_0^{(t)} - p_0^{(t-1)} \right) + b_m, \tag{5}$$

$$h_{gru}^{(t)} = \mathbf{GRU} \left(\mathbf{Cat} \left(\left[o^{(t)}, r^{(t)}, m^{(t)} \right], 2 \right) \right), \tag{6}$$

$$st^{(t)} = W_s h_{gru}^{(t)} + b_s, \tag{7}$$

where the linear layer with parameter W_m and b_m embeds the displacement of the ego-vehicle at two adjacent times into a feature in $\mathbb{R}^{1\times d_m}$. It is worth noting that here we concatenate three kinds of features along the 2nd dimension and produce a comprehensive representation of size $1 \times d_{st}$. Dimension length d_{st} equals to $(d_o + d_r + d_m)$. $\mathbf{GRU}$ is similar to LSTM except that it uses less parameters and converge faster with fewer training samples. GRU is used here because the number of ego-agents is much smaller than the number of other mobile agents in our problem.

Future ego-trajectory interaction

As the planned trajectory is given, the surrounding agents are inclined to adjust their future motion for collision avoidance. Given ego-trajectory Y_0, we first map it into a high-dimension space using an embedding layer and then pass the obtained embedded feature through a maxpooling function to generate the integrated representation $f^t \in \mathbb{R}^{1\times d_f}$ of the ego-agent. This representation is what we call a future ego-trajectory feature as it can influence the trajectories of the other on-road agents. The overall process is formulated as follows.

$$e_{f,0}^{(k)} = W_f p_0^{(k)} + b_f, \ \ \forall k, k \in [t+1, T_{pred}],$$

$$f^{(t)} = \mathbf{Maxpool} \left(\mathbf{Cat} \left(\left[{e_{f,0}^{(t+1)}}^T, \cdots, {e_{f,0}^{(T_{pred})}}^T \right], 1 \right) \right),$$

Finally, as show in Fig. 1, the output of the AIN $fst^{(t)} \in \mathbb{R}^{1\times(d_f+d_{st})}$ is obtained as below:

$$fst^{(t)} = \mathbf{Cat} \left(\left[f^{(t)}, st^{(t)} \right], 2 \right). \tag{10}$$

3.3 Environment Network

Road topology, such as intersections, turns, and slips, have significant influence on both the speed and directions of the agents. Therefore, it is an important factor in predicting the trajectories of agents. Here we use a network to encode the road topology, where the network is named as the Environment Network (EN). In our method, EN explicitly extracts the drivable area from a High Definition (HD) map. The center lines of the roads are normalized by subtracting the location of the ego-agent for the ego-perspective. And then, we transform the processed lines of roads into a semantic image $\mathbf{I}$ of the map of resolution $H \times W$. That is to say, the ego-agent always locates at the center of the image. Besides, to ensure the consistency of the image and the map, the road areas is trimmed with a fixed size of $h \times w$ meters from the HD map around the ego-agent. Then, the resolution of the semantic image is $[h/H, w/W]$ meters per pixel. At any time step, EN takes the road image $\mathbf{I}$ as input and encodes the environment through a pre-trained ResNet18 network [He *et al.*, 2016]. The output of the 2nd block of ResNet18 is used as the map feature. Compare to the size of image $\mathbf{I}$, the map feature is downsampled by a factor of 8.

Given the location of an agent, we pool the local road representation at its current location from the computed map feature. The environmental information within R_s meters around the agent is extracted from the obtained map feature. Thus, the corresponding Region Of the Interest (ROI) on the feature maps has a spatial window of $[HR_s/4h, WR_s/4w]$. We apply ROIAlign on the receptive field to generate a fixed size representation $G_i^{(t)} \in \mathbb{R}^{C\times K\times K}$, where C is the number of output channels in the last layer, and K is the pooling size. As environment feature $G_i^{(t)}$ is produced, we feed it to an embedding layer for dimension reduction and feature extraction. The computation of the embedding operation is written as:

$$g_i^{(t)} = \mathbf{Reshape} \left(G_i^{(t)}, [C \times K \times K, 1] \right), \tag{8}$$

$$v_i^{(t)} = W_v g_i^{(t)} + b_v, \tag{9}$$

where the function $\mathbf{Reshape}(\cdot)$ is used to adjust shape size of the target tensor, W_v and b_v are the layer parameters, and the dimension of the $v^{(t)}$ is d_v.

3.4 Trajectory Prediction Network

The global spaio-temporal interaction and the environment information are encoded by the AIN and the EN, respectively. Besides, given the location of a moving agent, i.e., the i-th one, we first compute the local area interaction around the agent through an attention model. This is because an individual always focuses on the surrounding regions as it moves. The attention model is presented below:

$$e_{c,i}^{(t)} = \sigma \left(W_c p_i^{(t)} + b_c \right), \tag{10}$$

$$q_i^{(t)} = \left(fst^{(t)} \right)^T \odot e_{c,i}^{(t)}, \tag{11}$$

where W_c and b_c are the parameters of the embedding layer, $\sigma(\cdot)$ is the sigmoid activation function, and $\odot$ is the element-wise vector-vector or matrix-matrix operation. This layer maps the input into an attention weight $e_{c,i}^{(t)}$ which has the same dimensionality as $fst^{(t)}$.

Following previous works, we utilize an LSTM-based sequence-to-sequence model to solve the prediction problem. For each obstacle, the encoder takes the observed trajectory as input at the first T_{obs} time steps:

$$e_{p,i}^{(t)} = W_p \left(p_i^{(t)} - p_i^{(t-1)} \right) + b_p, \tag{15}$$

$$h_{e,i}^{(t)} = \mathbf{LSTM_E} \left(h_{e,i}^{(t-1)}, \left[e_{p,i}^{(t)}, v_i^{(t)}, q_i^{(t)} \right] \right), \tag{16}$$

where W_p and b_p are the weights and bias respectively. $h_{e,i}^{(t)}$ is the hidden states of the encoder $\mathbf{LSTM_E}$. Here we set the

Table 1: Experimental Results (ADE/FDE) on the ETH and UCY datasets.

Method\Dataset	ETH-univ	ETH-hotel	UCY-univ	UCY-zara01	UCY-zara02	AVG
Linear (Single)	1.33/2.94	**0.39/0.72**	0.82/1.59	0.62/1.21	0.79/1.59	0.79/1.59
LSTM (Single)	1.09/2.41	0.86/1.91	0.61/1.31	0.41/0.88	0.52/1.11	0.70/1.52
Social LSTM	1.09/2.35	0.79/1.76	0.67/1.40	0.47/1.00	0.56/1.17	0.72/1.54
Social GAN (20V-20)	0.81/1.52	0.72/1.61	0.60/1.26	0.34/0.69	0.42/0.84	0.58/1.18
Social GAN (20VP-20)	0.87/1.62	0.67/1.37	0.76/1.52	0.35/0.68	0.42/0.84	0.61/1.21
Social Way	**0.39/0.64**	**0.39/0.66**	0.55/1.31	0.44/0.64	0.51/0.92	0.45/0.83
Sophie	0.70/1.43	0.76/1.67	0.54/1.24	**0.30/0.63**	0.38/0.78	0.54/1.15
Our method	**0.39/0.79**	0.51/1.05	**0.25/0.56**	**0.30/0.61**	**0.36/0.73**	**0.36/0.75**

hidden state to zero vector at the time step 0. The decoding process from time step $T_{obs}+1$ to T_{pred} has a similar flow with the encoding phase, and we use the predicted coordinates as input:

$$e_{p,i}^{(t)} = W_p\left(\widehat{p}_i^{(t)} - \widehat{p}_i^{(t-1)}\right) + b_p, \tag{17}$$

In practice, we set $\widehat{p}_i^{(T_{obs})}$ to $p_i^{(T_{obs})}$. Our approach forecasts obstacle's position at each prediction moment using the hidden states of the deocder $\mathbf{LSTM_D}$:

$$h_{d,i}^{(t)} = \mathbf{LSTM_D}\left(h_{d,i}^{(t-1)}, \left[e_{p,i}^{(t)}, v_i^{(t)}, q_i^{(t)}\right]\right), \tag{18}$$

$$\Delta\widehat{p}_i^{(t+1)} = W_u h_{d,i}^{(t)} + b_u, \tag{19}$$

$$\widehat{p}_i^{(t+1)} = \widehat{p}_i^{(t)} + \Delta\widehat{p}_i^{t+1}. \tag{20}$$

Furthermore, to capture the multi-modal distribution of the movement and increase the robustness of the proposed method, we introduce a gaussian random noise into the decoder to generate multiple plausible trajectories. Specifically, we initialize the hidden state of the $\mathbf{LSTM_D}$ using the last state of the $\mathbf{LSTM_E}$:

$$e_{h,i}^{(T_{obs})} = \mathbf{Cat}\left(\left[h_{e,i}^{(T_{obs})}, z\right], 1\right), \tag{21}$$

$$h_{d,i}^{(T_{obs})} = W_\phi e_{h,i}^{(T_{obs})} + b_\phi, \tag{22}$$

where z is some gaussian random noise.

3.5 Implementation Details

Our network is trained end-to-end by minimizing the mean square error as below:

$$\mathcal{L} = \frac{1}{NT}\min_{i\in\{1\ldots H\}}\sum_{j=1}^{N}\sum_{t=T_{obs}+1}^{T_{pred}}\left(\widehat{p}_{j,i}^{(t)} - p_j^{(t)}\right)^2, \tag{23}$$

where T is the prediction time steps which equals $T_{pred} - T_{obs}$. H is the number of modalities (predicted trajectories). We only back propagate the gradient to the modality with the minimum error.

We set the output dimensions of all the embedding layers (exclude attention and noise embedding layer in Trajectory Prediction Network) to be 64. The GRU in the AIN has 128 cells, while the LSTMs in the Trajectory Prediction Network have 64 cells. In the EN, the local area size R_s and the pool size K are set as 20 and 3 respectively. Meanwhile, both the height H and width W of the road semantic image are set to 224. The road area size h and w are set to be 100 meters. Our network is trained with a batch size of 8 for 20000 steps using Adam optimizer with an initial learning rate of 0.0005. The entire training process is finished in the platform with an NVIDIA GeForce RTX2080 GPU.

4 Experiments

In this section, we evaluate the proposed method on four benchmark datasets for future trajectory prediction and demonstrate our method performs favorably against state-of-the-art prediction methods. The codes and pre-trained models of our method will be released to the public.

4.1 Datasets Description

ETH [Pellegrini *et al.*, 2009] and UCY [Lerner *et al.*, 2007] are two common benchmarks for pedestrian trajectory prediction. These two datasets consists of 5 scenes, including ETH-univ, ETH-hotel, UCY-zara01, UCY-zara02 and UCY-univ. There are totally 1536 pedestrians in total with thousands of nonlinear trajectories. The same leave-one-set-out strategy as in previous study [Alahi *et al.*, 2016] is used to evaluate the compared methods.

Besides pedestrian walking datasets, the ApolloScape [Ma *et al.*, 2019a] and the Argoverse [Chang *et al.*, 2019] datasets are utilized to demonstrate the performance of the compared methods. ApolloScape dataset is comprised of different kinds of traffic agents which include cars, buses, pedestrians and bicycles. This dataset is very challenging because it is a heterogeneous multi-agent system. On the other hand, Argoverse dataset contains 327790 sequences of different scenarios. Each sequence follows the trajectory of ego-agent for 5 seconds while keeping track of all other agents (cars, pedestrians etc.). The dataset is split into a training data with 208272 sequences and a validation data with 79391 sequences. For ApolloScape, the trajectories of 3 seconds (6 time steps) are observed and the prediction methods are required to predict the trajectory in the following 3 seconds (6 time steps). And for Argoverse, 2 seconds with 20 time steps are observed and the methods are required to predict the trajectories in the following 3 seconds with 30 time steps.

4.2 Experimental setup

The experimental results are reported in terms of two evaluation metrics, the Average Displacement Error (ADE) and Final Displacement Error (FDE). ADE is defined as the mean square error over all prediction points of a trajectory and the

Table 2: Experimental Results (ADE/FDE) on the ApolloScape Dataset.

Dataset\Method	Linear	KF	LSTM	Noise-LSTM	Social LSTM	Social GAN	Our method
ApolloScape	1.95/3.39	2.48/4.33	1.63/2.85	1.49/2.58	1.23/2.11	1.21/2.14	**1.11/1.91**

Table 3: Experimental Results (ADE/FDE) on the Argoverse Dataset.

Dataset\Method	Linear	KF	LSTM	Noise-LSTM	Social LSTM	Social GAN	Our method
Relative coordinate	1.84/3.91.	2.53/5.84	1.47/3.18	1.48/3.17	1.23/2.49	1.31/2.63	**1.18/2.45**
World coordinate	1.57/3.31	2.56/5.96	1.24/2.63	1.25/2.61	1.22/2.45	1.32/2.67	**1.15/2.29**

Table 4: Ablation Study on the Argoverse Dataset.

Method	Component					Metric	
	PF	TF	EMF	ETF	EF	ADE	FDE
Baseline	-	-	-	-	-	1.48	3.17
Our-v1	✓	-	-	-	-	1.32	2.70
Our-v2	✓	✓	-	-	-	1.24	2.56
Our-v3	✓	✓	✓	-	-	1.22	2.51
Our-v4	✓	✓	✓	✓	-	1.19	2.47
Our-full	✓	✓	✓	✓	✓	**1.18**	**2.45**

ground truth points of the trajectory, whereas FDE is the distance between the predicted final location and the ground truth final location at the end of the prediction time period.

We use the the linear regressor, the extended Kalman Filter (KF), and the vanila-LSTM as the baselines. Moreover, many state-of-the-art trajectory prediction methods are compared. Social-LSTM [Alahi *et al.*, 2016] is a prediction method that combines LSTMs with a social pooling layer. Social-GAN [Gupta *et al.*, 2018] applies a GAN model to social LSTMs for prediction. Social-Way [Amirian *et al.*, 2019] utilizes a GAN model to propose plausible future trajectories and train the predictor. Sophie [Sadeghian *et al.*, 2019] introduces a social and physical attention mechanism to a GAN predictor.

Because there is no HD map information and the planned trajectory for the ego-agent, there is no ego-trajectory feature, ego-motion feature and environment feature in the proposed method for the ETH & UCY, and the ApolloScape datasets. For Argoverse dataset, the proposed method with all the features and components is implemented for comparisons.

4.3 Performance Evaluation

ETH & UCY The experimental results on the ETH and UCY datasets are presented in Table 1. As expected, the baselines, the Linear and LSTM, are incapable in capturing the complexity patterns in the trajectories of pedestrians. Our method outperforms the other methods on the UCY-univ and UCY-zara02 subsets and shows competitive results on the ETH-univ and UCY-zara01 subsets. On the ETH-hotel, both linear and social way methods show lower prediction errors than other methods. This indicate that the trajectories in ETH-hotel are linearly distributed and thus are simpler than the other 4 subsets. As the methods are all trained on the other 4 subsets, these nonlinear predictors, such as Social LSTM, Social-GAN, Sophie, show poor generalization ability on ETH-hotel. On the other hand, our method still outperforms Social LSTM, Social-GAN, and Sophie on the ETH-hotel subset.

ApllоScape The performance of the compared methods on the ApolloScape is shown in Table 2. As can be seen, the proposed method outperforms the runner-up in term of ADE/FDE with about 10% improvement in accuracy. It means that our interaction net has faithfully learn the intrinsic interactive patterns and the attention module could extract the specialized feature for each category of the heterogeneous traffic-agents

Argoverse Argoverse provides the HD road map and the planned path for ego-car. The proposed method with all the components is implemented. The experimental results are shown in Table 3. As can be seen, the prediction error of the proposed method is significantly lower than the other methods. We observe 11% and 4% improvement in ADE comparing the Social GAN and Social LSTM methods when the relative coordinate system (ego-perspective) is used. The improvement in ADE is 14% and 6% comparing the Social GAN and the social LSTM methods when the world coordinate system is used.

Ablation Study The proposed method is comprised of multiple separate components, each with different functions. To show the effectiveness of the components, we perform ablation study of the components of the proposed method and the results are presented in Table 4. We use PF, TF, EMF, ETF, EF to represent the position feature, tracking feature, ego motion feature, ego trajectory feature, and environment feature of the proposed method, respectively. The results indicate that PF and TF are contributive to the significant improvement, and the values of the reduction in prediction error are 0.47 and 0.14, respectively. And EMF, ETF, EF all show some level of contributions such that the values of the reduction in prediction error are 0.5, 0.4 and 0.2, respectively. The value of the reduction in error is dropping because it is harder to reduce the prediction error when the error is lower.

5 Conclusion

We propose a new method for trajectory forecasting of multiple agents in dynamic scenes. The method is able to extract the global spatio-temporal interaction feature from the past trajectories, and consider the temporal interactions among agents by soft tracking. An environment net is introduced in our method to encode the road topology for accurate prediction. And the prediction net combines the features of spatio-temporal interactions and environment to prediction the future trajectories of agents. Experiments on four benchmark datasets are presented and the ablation study is implemented to show the effectiveness of each component of the method.

References

[Alahi *et al.*, 2016] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, F. Li, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[Amirian *et al.*, 2019] J. Amirian, J. B. Hayet, and J. Pettre. Social ways: Learning multi-modal distributions of pedestrian trajectories with gans. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[Bi *et al.*, 2019] Huikun Bi, Zhong Fang, Tianlu Mao, Zhaoqi Wang, and Zhigang Deng. Joint prediction for kinematic trajectories in vehicle-pedestrian-mixed scenes. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[Chandra *et al.*, 2019] R. Chandra, A. Bhattacharya, U.and Bera, and D. Manocha. Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[Chang *et al.*, 2019] M. Chang, J. W. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays. Argoverse: 3d tracking and forecasting with rich maps. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[Choi and Dariush, 2019] C. Choi and B. Dariush. Looking to relations for future trajectory forecast. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[Chung *et al.*, 2014] J. Chung, C. Gulcehre, K.H. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

[Gupta *et al.*, 2018] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[He *et al.*, 2016] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[Helbing and Molnár, 1995] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51:4282–4286, May 1995.

[Huang *et al.*, 2019] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[Ivanovic and Pavone, 2019] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[Kosaraju *et al.*, 2019] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In *Advances in Neural Information Processing Systems 32*, pages 137–146. 2019.

[Lefèvre *et al.*, 2011] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmán. Exploiting map information for driver intention estimation at road intersections. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 583–588, June 2011.

[Lerner *et al.*, 2007] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. *Computer Graphics Forum*, 26(3):655–664, 2007.

[Liang *et al.*, 2019] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G. Hauptmann, and Li Fei-Fei. Peeking into the future: Predicting future person activities and locations in videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[Ma *et al.*, 2019a] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:6120–6127, 07 2019.

[Ma *et al.*, 2019b] Yuexin Ma, Xinge Zhu, Sibo Zhang, Ruigang Yang, Wenping Wang, and Dinesh Manocha. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. In *The AAAI Conference on Artificial Intelligence (AAAI)*, pages 6120–6127, February 2019.

[Martinez *et al.*, 2017] J. Martinez, M. J. Black, and J. Romero. On human motion prediction using recurrent neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4674–4683, July 2017.

[Pellegrini *et al.*, 2009] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268, Sep. 2009.

[Pellegrini *et al.*, 2010] S. Pellegrini, A. Ess, and L. Van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *European Conference on Computer Vision (ECCV)*, pages 452–465, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[Rasmussen and Williams, 2005] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, Cambridge, MA, USA, 2005.

[Sadeghian *et al.*, 2019] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[Su *et al.*, 2017] H. Su, J. Zhu, Y. Dong, and B. Zhang. Forecast the plausible paths in crowd scenes. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2772–2778, 2017.

[Toledo-Moreo and Zamora-Izquierdo, 2009] R. Toledo-Moreo and M. A. Zamora-Izquierdo. Imm-based lane-change prediction in highways with low-cost gps/ins. *IEEE Transactions on Intelligent Transportation Systems*, 10(1):180–185, March 2009.

[Van den Berg *et al.*, 2011] Jur Van den Berg, Stephen J. Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In Cédric Pradalier, Roland Siegwart, and Gerhard Hirzinger, editors, *Robotics Research*, pages 3–19, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[Xu *et al.*, 2018] Y. Xu, Z. Piao, and S. Gao. Encoding crowd interaction with deep neural network for pedestrian trajectory prediction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Stereo Visual Inertial Odometry with Online Baseline Calibration

Yunfei Fan, Ruofu Wang, and Yinian Mao

***Abstract*— Stereo-vision devices have rigorous requirements for extrinsic parameter calibration. In Stereo Visual Inertial Odometry (VIO), inaccuracy in or changes to camera extrinsic parameters may lead to serious degradation in estimation performance. In this manuscript, we propose an online calibration method for stereo VIO extrinsic parameters correction. In particular, we focus on Multi-State Constraint Kalman Filter (MSCKF [1]) framework to implement our method. The key component is to formulate stereo extrinsic parameters as part of the state variables and model the Jacobian of feature reprojection error with respect to stereo extrinsic parameters as sub-block of update Jacobian. Therefore we can estimate stereo extrinsic parameters simultaneously with inertial measurement unit (IMU) states and camera poses. Experiments on EuRoC dataset and real-world outdoor dataset demonstrate that the proposed algorithm produce higher positioning accuracy than the original S-MSCKF [2], and the noise of camera extrinsic parameters are self-corrected within the system.**

I. INTRODUCTION

In recent years, high-precision positioning technologies have progressed significantly, propelling the advancements in multiple application scenarios such as autonomous driving, robotics and unmanned aerial vehicles (UAVs), and augmented and virtual reality (AR and VR). In outdoor environments, GNSS such as GPS and RTK can be employed. In indoor and GPS-denied environments, Lidar and visual SLAM can be used. For applications that are limited by device size and weight requirements, the applicable positioning technology is rather limited in the absence of GPS. Since VIO only requires IMU and one or two camera modules to estimate ego-motion, it is naturally suitable for such scenarios. It has been reported that stereo-vision VIO system can improve the overall estimation accuracy over single-vision VIO system (S-MSCKF [2], VINS-Fusion [3,4]). A good stereo calibration ensures the epipolar lines of stereo images being parallel, which is the foundation for most stereo matching algorithms. However, in stereo VIO systems, the estimation accuracy heavily depends on camera extrinsic parameters calibration. With a poor calibration or slight changes in camera parameters during operation, stereo VIO positioning accuracy will drop sharply. Even with rigid and bulky frames, most stereo cameras cannot ensure that extrinsic parameters are unchanged during long course of operations. Within this context, an accurate calibration algorithm that is robust to changes in camera extrinsic parameters is highly desired.

In this paper, we propose a stereo VIO algorithm with online calibration to overcome the above issues. The core method is to formulate stereo camera extrinsic parameters (rotation and translation) into the set of state variables and model the relevance between feature reprojection error and stereo extrinsic parameters in update Jacobian, so that the stereo extrinsic parameters can be calibrated online as part of the state estimation. To accelerate the self-calibration process, the initial covariance of stereo extrinsic parameters is set to a large value. In addition, during the initial phase of the estimation, the threshold of the outlier rejection rule based on stereo extrinsic constraint on the algorithm frontend is relaxed to avoid too many inliers being mistakenly taken out.

Using EuRoC dataset and real-world outdoor dataset, we compare the proposed scheme with other state-of-the-art stereo VIO algorithm, specifically S-MSCKF. The experiments show that, without calibration errors, the proposed method performs similarly to S-MSCKF. Besides, when artificial noises are involved in the calibrated parameters, the proposed scheme can achieve rapid self-calibration and outperforms S-MSCKF in position estimation.

The rest of this paper is organized as follows: Section II introduces related works. Section III introduces system framework and derives analytical formulations. Section IV compares experimental results of the proposed scheme with those of VINS-Fusion [3,4] and S-MSCKF using EuRoC dataset and real-world outdoor datasets collected by UAVs as well as by a handheld device. Finally, the conclusions are summarized in section V.

II. RELATED WORK

The current scholarly works in VIO could be roughly divided into loosely-coupled [5,6] and tightly-coupled [1- 4,7] methods. Tightly-coupled methods put IMU information into state variables and optimize with vision information simultaneously, which is a mainstream direction currently. Tightly-coupled methods can be divided further into filter-based and optimization-based.

VIO methods based on non-linear optimization utilize all measurements, including IMU measurements and visual measurements, to find the optimal state variables to minimize the measurement error. Stereo VIO based on non-linear optimization includes OKVIS [7], VINS-Fusion [3,4], etc. Both OKVIS and VINS-Fusion perform online estimation of extrinsic parameters between the IMU and each camera, separately. However, due to the large number of state variables, even the current mainstream sliding window based VIO methods using non-linear optimization have a considerable demand for computational resource, and it is still difficult to run in real time on embedded platforms.

Filter based VIO methods are mainly based on Extended Kalman Filter (EKF) [1]. Generally, IMU is used for prediction, while visual information is used for update. They achieve almost the same level of accuracy as optimiza-

This work was supported by the Meituan-Dianping Group.

Yunfei Fan and Yinian Mao are with the Meituan-Dianping Group, Beijing, China (e-mail: {fanyunfei | maoyinian}@meituan.com).

Ruofu Wang is with University of Southern California, Los Angeles, CA 90007 USA (e-mail: ruofuwan@usc.edu)

tion-based methods using relatively low computational resources. Thus they can run in real time on embedded platforms. S-MSCKF [2] is one of filter-based stereo VIO frameworks, which only estimates the extrinsic parameters between IMU and left camera online. In order to achieve real-time performance, our method is also based on MSCKF framework.

P Hansen et al. [8] and Yonggen Ling et al. [9] proposed approaches to estimate stereo extrinsic parameters online. They are all based on epipolar geometric constraints for online self-calibration of stereo extrinsic. However, because pure vision-based methods cannot self-calibrate the baseline fully, they can only achieve estimation of stereo extrinsic parameters with 5-DOF, while the length of the baseline cannot be estimated. Therefore, we use the IMU and the cameras jointly, to self-calibrate the 6-DOF stereo extrinsic parameters online.

III. MSCKF ALGORITHM FRAMEWORK

A. State definition

Following the definition of MSCKF in [1], IMU state is defined below:

$$\mathrm{X_I} = \left[{}^{\mathrm{G}}\mathrm{p_I}^T \ {}^{\mathrm{G}}\mathrm{v_I}^T \ {}^{\mathrm{I}}_{\mathrm{G}}\bar{\mathrm{q}}^{\mathrm{T}} \ \mathrm{b_a}^{\mathrm{T}} \ \mathrm{b_g}^{\mathrm{T}} \right]^{\mathrm{T}} \tag{1}$$

In this paper, different from [1, 2], both extrinsic parameters $\mathrm{E_0}$ and $\mathrm{E_1}$ of stereo VIO system shown in Fig. 1, are added into IMU states and calibrated online. The extended IMU states are defined:

$$\mathrm{X_I} = \left[{}^{\mathrm{G}}\mathrm{p_I}^T \ {}^{\mathrm{G}}\mathrm{v_I}^T \ {}^{\mathrm{I}}_{\mathrm{G}}\bar{\mathrm{q}}^{\mathrm{T}} \ \mathrm{b_a}^{\mathrm{T}} \ \mathrm{b_g}^{\mathrm{T}} \ {}^{\mathrm{C^0}}_{\mathrm{I}}\bar{\mathrm{q}}^{\mathrm{T}} \ {}^{\mathrm{I}}\mathrm{p_{C^0}}^T \ {}^{\mathrm{C^1}}_{\mathrm{C^0}}\bar{\mathrm{q}}^{\mathrm{T}} \ {}^{\mathrm{C^0}}\mathrm{p_{C^1}}^T \right]^{\mathrm{T}} \tag{2}$$

In these expressions, $\{\mathrm{G}\}$ and $\{\mathrm{I}\}$ are the global and inertial frame respectively, $\{\mathrm{C^0}\}$ and $\{\mathrm{C^1}\}$ are frame of $\mathrm{C^0}$ and $\mathrm{C^1}$ respectively. ${}^{\mathrm{G}}\mathrm{p_I}$ and ${}^{\mathrm{G}}\mathrm{v_I}$ are position and velocity of IMU expressed in $\{\mathrm{G}\}$, respectively. 4×1 ${}^{\mathrm{I}}_{\mathrm{G}}\bar{\mathrm{q}}$ represents the rotation from $\{\mathrm{G}\}$ to $\{\mathrm{I}\}$ (in this paper, quaternion obeys JPL rules). The vectors $\mathrm{b_g}$ and $\mathrm{b_a}$ are the biases of the measured angular velocity and linear acceleration from the IMU, separately. ${}^{\mathrm{C^0}}_{\mathrm{I}}\bar{\mathrm{q}}$ represents the rotation from $\{\mathrm{I}\}$ to $\{\mathrm{C^0}\}$, and ${}^{\mathrm{I}}\mathrm{p_{C^0}}$ is the position of $\mathrm{C^0}$ based on frame $\{\mathrm{I}\}$ (${}^{\mathrm{C^0}}_{\mathrm{I}}\bar{\mathrm{q}}$ and ${}^{\mathrm{I}}\mathrm{p_{C^0}}$ are the rotation and translation of extrinsic parameter $\mathrm{E_0}$ respectively). Finally, ${}^{\mathrm{C^1}}_{\mathrm{C^0}}\bar{\mathrm{q}}$ represents the rotation from frame $\{\mathrm{C^0}\}$ to frame $\{\mathrm{C^1}\}$, and ${}^{\mathrm{C^0}}\mathrm{p_{C^1}}$ is the position of $\mathrm{C^1}$ based on frame $\{\mathrm{C^0}\}$ (${}^{\mathrm{C^1}}_{\mathrm{C^0}}\bar{\mathrm{q}}$ and ${}^{\mathrm{C^0}}\mathrm{p_{C^1}}$ are the rotation and translation of stereo extrinsic parameter $\mathrm{E_1}$ respectively. We treat stereo extrinsic parameters as ${}^{\mathrm{C^1}}_{\mathrm{C^0}}\bar{\mathrm{q}}$ and ${}^{\mathrm{C^0}}\mathrm{p_{C^1}}$ later).

The EKF error-state of $\mathrm{X_I}$ is defined accordingly:

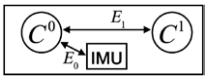


Figure 1. Structure diagram of sensor, and definition of extrinsic parameters

$$\tilde{\mathrm{X}}_\mathrm{I} = \left[{}^{\mathrm{G}}\tilde{\mathrm{p}}_\mathrm{I}^{\mathrm{T}} \ {}^{\mathrm{G}}\tilde{\mathrm{v}}_\mathrm{I}^{\mathrm{T}} \ {}^{\mathrm{I}}_{\mathrm{G}}\tilde{\theta}^{\mathrm{T}} \ \tilde{\mathrm{b}}_\mathrm{a}^{\mathrm{T}} \ \tilde{\mathrm{b}}_\mathrm{g}^{\mathrm{T}} \ {}^{\mathrm{C^0}}_{\mathrm{I}}\tilde{\theta}^{\mathrm{T}} \ {}^{\mathrm{I}}\tilde{\mathrm{p}}_{\mathrm{C^0}}^{\mathrm{T}} \ {}^{\mathrm{C^1}}_{\mathrm{C^0}}\tilde{\theta}^{\mathrm{T}} \ {}^{\mathrm{C^0}}\tilde{\mathrm{p}}_{\mathrm{C^1}}{}^{T} \right]^{\mathrm{T}} \tag{3}$$

Except for quaternions, other states can be used with standard additive error (e.g. $\mathrm{x} = \hat{\mathrm{x}} + \tilde{\mathrm{x}}$). the extended additive error of quaternion is defined in [10] (in this paper, quaternion error is defined in frame $\{\mathrm{I}\}$, see details in [11])

$${}^{\mathrm{I}}_{\mathrm{G}}\bar{\mathrm{q}} = \delta_{\mathrm{G}}^{\mathrm{I}}\bar{\mathrm{q}} \otimes {}^{\mathrm{I}}_{\mathrm{G}}\hat{\bar{\mathrm{q}}}, \ \ \delta_{\mathrm{G}}^{\mathrm{I}}\bar{\mathrm{q}} = \left[1 \ \tfrac{1}{2} \ {}^{\mathrm{I}}_{\mathrm{G}}\tilde{\theta}\right]^{\mathrm{T}} \tag{4}$$

similarly, the extended additive error of rotation matrix is defined:

$$\mathrm{R}\left({}^{\mathrm{I}}_{\mathrm{G}}\bar{\mathrm{q}}\right) = {}^{\mathrm{I}}_{\mathrm{G}}\mathrm{R}, \ {}^{\mathrm{I}}_{\mathrm{G}}\mathrm{R} = (1 - \left[{}^{\mathrm{I}}_{\mathrm{G}}\tilde{\theta}\right]_\times) {}^{\mathrm{I}}_{\mathrm{G}}\hat{\mathrm{R}} \tag{5}$$

B. State Propagation

Similar to EKF state propagation, MSCKF framework uses IMU data to propagate states. The difference is state augmentation at the moment of new image arrival. As can be seen from [1], The time evolution of IMU states are described below:

$$\begin{aligned} {}^{\mathrm{I}}_{\mathrm{G}}\dot{\bar{\mathrm{q}}}(\mathrm{t}) &= \frac{1}{2}\Omega\left(\omega(\mathrm{t})\right) {}^{\mathrm{I}}_{\mathrm{G}}\bar{\mathrm{q}}(\mathrm{t}) \\ \dot{\mathrm{b}}_\mathrm{g}(\mathrm{t}) &= \mathrm{n_{wg}}(\mathrm{t}) \\ {}^{\mathrm{G}}\dot{\mathrm{v}}_\mathrm{I}(\mathrm{t}) &= {}^{\mathrm{G}}\mathrm{a}(\mathrm{t}) \\ \dot{\mathrm{b}}_\mathrm{a}(\mathrm{t}) &= \mathrm{n_{wa}}(\mathrm{t}) \\ {}^{\mathrm{G}}\dot{\mathrm{p}}_\mathrm{I}(\mathrm{t}) &= {}^{\mathrm{G}}\mathrm{v}_\mathrm{I}(\mathrm{t}) \end{aligned} \tag{6}$$

where ${}^{\mathrm{G}}\mathrm{a}$ represents the body acceleration in frame $\{\mathrm{G}\}$. $\omega = \left[\omega_\mathrm{x} \ \omega_\mathrm{y} \ \omega_\mathrm{z}\right]^{\mathrm{T}}$ represents angular velocity of IMU expressed in frame $\{\mathrm{I}\}$. And:

$$\Omega(\omega) = \begin{bmatrix} -[\omega]_\times & \omega \\ -\omega^{\mathrm{T}} & 0 \end{bmatrix}, \ [\omega]_\times = \begin{bmatrix} 0 & -\omega_\mathrm{z} & \omega_\mathrm{y} \\ \omega_\mathrm{z} & 0 & -\omega_\mathrm{x} \\ -\omega_\mathrm{y} & \omega_\mathrm{x} & 0 \end{bmatrix} \tag{7}$$

ω_m and $\mathrm{a_m}$ are the gyroscope and accelerometer measurements separately. Ignored the effects of the planet's rotation, they are given by [1]:

$$\begin{aligned} \omega_\mathrm{m} &= \omega + \mathrm{b_g} + \mathrm{n_g} \\ \mathrm{a_m} &= \mathrm{R}\left({}^{\mathrm{I}}_{\mathrm{G}}\bar{\mathrm{q}}\right)\left({}^{\mathrm{G}}\mathrm{a} - {}^{\mathrm{G}}\mathrm{g}\right) + \mathrm{b_a} + \mathrm{n_a} \end{aligned} \tag{8}$$

where ${}^{\mathrm{G}}\mathrm{g}$ is gravitational acceleration, expressed in frame $\{\mathrm{G}\}$. Applying Eq. (6) in Eq. (8), continuous dynamic model of IMU states can be obtained:

$$\begin{aligned} {}^{\mathrm{I}}_{\mathrm{G}}\dot{\hat{\bar{\mathrm{q}}}} &= \tfrac{1}{2}\Omega(\hat{\omega}) {}^{\mathrm{I}}_{\mathrm{G}}\hat{\bar{\mathrm{q}}}, \ \dot{\hat{\mathrm{b}}}_\mathrm{g} = 0_{3\times 1}, \\ {}^{\mathrm{G}}\dot{\hat{\mathrm{v}}}_\mathrm{I} &= \mathrm{R}\left({}^{\mathrm{I}}_{\mathrm{G}}\hat{\bar{\mathrm{q}}}\right)^{\mathrm{T}}\hat{\mathrm{a}} + {}^{\mathrm{G}}\mathrm{g} \\ \dot{\hat{\mathrm{b}}}_\mathrm{a}(\mathrm{t}) &= 0_{3\times 1}, \qquad {}^{\mathrm{G}}\dot{\hat{\mathrm{p}}}_\mathrm{I} = {}^{\mathrm{G}}\hat{\mathrm{v}}_\mathrm{I} \end{aligned} \tag{9}$$

moreover, $\hat{\mathrm{a}} = \mathrm{a_m} - \hat{\mathrm{b}}_\mathrm{a}$, $\hat{\omega} = \omega_\mathrm{m} - \hat{\mathrm{b}}_\mathrm{g}$, continuous dynamic model of IMU error-state is defined by:

$$\dot{\tilde{\mathrm{X}}}_\mathrm{I} = \mathrm{F}\tilde{\mathrm{X}}_\mathrm{I} + \mathrm{G}\mathrm{n}_\mathrm{I} \tag{10}$$

where $n_I = \begin{bmatrix} n_g^T & n_{\omega g}^T & n_a^T & n_{\omega a}^T \end{bmatrix}^T$ is the system noise. It depends on the IMU noise characteristics. Finally, the matrices F and G that appear in Eq. (10) are given by:

$$F = \begin{bmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times9} \\ 0_{3\times3} & 0_{3\times3} & -R(\hat{{}_G^I q})^T[\hat{a}]_\times & -R(\hat{{}_G^I q})^T & 0_{3\times3} & 0_{3\times9} \\ 0_{3\times3} & 0_{3\times3} & -[\hat{\omega}]_\times & 0_{3\times3} & -I_{3\times3} & 0_{3\times9} \\ 0_{18\times3} & 0_{18\times3} & 0_{18\times3} & 0_{18\times3} & 0_{18\times3} & 0_{18\times9} \end{bmatrix} \tag{11}$$

$$G = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & -R(\hat{{}_G^I q})^T & 0_{3\times3} \\ -I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{12\times3} & 0_{12\times3} & 0_{12\times3} & 0_{12\times3} \end{bmatrix} \tag{12}$$

Following Euler integration [4] of Eq. (10), discrete-time system matrix is given by:

$$\Phi = I + F\delta t \tag{13}$$

Moreover, the propagation of covariance is given by:

$$P = \Phi P \Phi^T + (\Phi G) Q (\Phi G)^T \delta t$$

In this paper, covariance structure is defined as:

$$P_{k|k} = \begin{bmatrix} P_{II_{k|k}} & P_{IC_{k|k}} \\ P_{IC_{k|k}}^T & P_{CC_{k|k}} \end{bmatrix} \tag{14}$$

Since the current state of IMU propagation doesn't change the pose of sliding window, we can formulate the covariance propagation method:

$$P_{k+1|k} = \begin{bmatrix} P_{II_{k+1|k}} & \Phi P_{IC_{k|k}} \\ P_{IC_{k|k}}^T \Phi^T & P_{CC_{k|k}} \end{bmatrix} \tag{15}$$

where, $P_{II_{k+1|k}} = \Phi P_{II_{k|k}} \Phi^T + (\Phi G) Q (\Phi G)^T \delta t$, and P_{II} represents covariance of IMU states. P_{IC} represents covariance of IMU states with respect to pose of cameras. P_{CC} represents covariance of pose of augmented cameras.

When a new image arrives, current state of system should be augmented (in this paper, we augment the left camera state similarly to [2]). Including augmented states, the extended states are defined as:

$$\tilde{X}_k = \begin{bmatrix} \tilde{X}_I^T & \tilde{X}_{C_0^0}^T & \tilde{X}_{C_1^0}^T & \tilde{X}_{C_2^0}^T & \cdots & \tilde{X}_{C_N^0}^T \end{bmatrix}^T \tag{15}$$

where $\tilde{X}_{C_j^0} = \begin{bmatrix} {}_G^{C_j^0}\tilde{\theta}^T & {}^G\tilde{p}_{C_j^0}^T \end{bmatrix}^T, j = (0, \dots, N-1)$ represents the pose of augmented camera C^0. It is derived from extrinsic parameter E_0 and IMU states:

$$\begin{aligned} {}_G^{C^0}\hat{q} &= {}_I^{C^0}\hat{q} \otimes {}_G^I\hat{q} \\ {}^G\hat{p}_{C^0} &= {}^G\hat{p}_I + R(\hat{{}_G^I q})^T \cdot {}^I\hat{p}_{C^0} \end{aligned} \tag{16}$$

Hence, in Error State Kalman Filter (ESKF [12]) framework, error-state of system (including augmented cameras) is defined by:

$$\tilde{X}_k = \begin{bmatrix} \tilde{X}_I^T & \tilde{X}_{C_0^0}^T & \tilde{X}_{C_1^0}^T & \tilde{X}_{C_2^0}^T & \cdots & \tilde{X}_{C_N^0}^T \end{bmatrix}^T \tag{17}$$

where, $\tilde{X}_{C_j^0} = \begin{bmatrix} {}_G^{C_j^0}\tilde{\theta}^T & {}^G\tilde{p}_{C_j^0}^T \end{bmatrix}^T, j = (0, \dots, N-1)$ represents the error of j^{th} augmented camera C^0. Moreover, augmented covariance is defined by:

$$P'_{k|k} = \begin{bmatrix} P_{k|k} & P_{21}^T \\ P_{21} & P_{22} \end{bmatrix} \tag{18}$$

Note that $P_{21} = JP_{k|k}$, $P_{22} = JP_{k|k}J^T$ are the augmented covariance with respect to j^{th} augmented state, and J is the Jacobian of $\tilde{X}_{C_j^0}$ with respect to the error-state vector.

$$J = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & {}_G^{C^0}\hat{R} & 0_{3\times6} & I_{3\times3} & 0_{3\times3} & 0_{3\times(6N+6)} \\ I_{3\times3} & 0_{3\times3} & -{}_G^I\hat{R}^T[{}^I\hat{p}_{C^0}]_\times & 0_{3\times6} & 0_{3\times3} & {}_G^I\hat{R}^T & 0_{3\times(6N+6)} \end{bmatrix} \tag{19}$$

C. State Update

Similar to [2], we can formulate the reprojection of features from stereo. Different from [2], the extrinsic parameters E_1 employed in this paper is calibrated online. $\left({}_G^{C_i^0}\bar{q}\ {}^G p_{C_i^0}\right)$ and $\left({}_G^{C_i^1}\bar{q}\ {}^G p_{C_i^1}\right)$ are i^{th} left and right camera pose at the same time instance respectively. Employing the stereo extrinsic, the pose of the right camera C^1 can be easily derived in terms of the left camera augmented(e.g. ${}_G^{C^1}\bar{q} = {}_{C^0}^{C^1}\bar{q} \otimes {}_G^{C^0}\bar{q}$, ${}^G p_{C^1} = {}^G p_{C^0} + R({}_G^{C^0}\bar{q})^T \cdot {}^{C^0}p_{C^1}$). The reprojection of stereo measurement, $\hat{z}_i^j$ in i^{th} pose is defined as:

$$\hat{z}_i^j = \begin{pmatrix} \hat{u}_{i,0}^j \\ \hat{v}_{i,0}^j \\ \hat{u}_{i,1}^j \\ \hat{v}_{i,1}^j \end{pmatrix} = \begin{pmatrix} \frac{1}{{}_i^{C_0}\tilde{z}_j} & 0_{2\times2} \\ 0_{2\times2} & \frac{1}{{}_i^{C_1}\tilde{z}_j} \end{pmatrix} \begin{pmatrix} {}_i^{C_0}\hat{X}_j \\ {}_i^{C_0}\hat{Y}_j \\ {}_i^{C_1}\hat{X}_j \\ {}_i^{C_1}\hat{Y}_j \end{pmatrix} \tag{20}$$

Note that $\begin{bmatrix} {}_i^{C^k}\hat{X}_j & {}_i^{C^k}\hat{Y}_j & {}_i^{C^k}\hat{X}_j & {}_i^{C^k}\hat{Y}_j \end{bmatrix}$ is the coordinate of j^{th} feature in frame $\{C^k\}$ in i^{th} camera pose of sliding window (k=0,1 represents left and right camera respectively). Measurement residual is defined as:

$$r_i^{j,k} = z_i^{j,k} - \hat{z}_i^{j,k} \tag{21}$$

We can formulate least-squares system to optimize the coordinates of features. See details in [13]. Then, the reprojection error of j^{th} feature observation in i^{th} camera pose in sliding window is derived as:

$$\begin{aligned} r_i^{j,k} = z_i^{j,k} - \hat{z}_i^{j,k} &\approx H_{X_i}^{j,k}\tilde{X} + H_{f_i}^{j,k}\,{}^G\tilde{p}_{f_j} + n_i^{j,k} \\ H_{X_i}^{j,0} &= \begin{bmatrix} 0_{2\times(27+6i)} & H_1 & 0_{2\times6(N-i-1)} \end{bmatrix} \\ H_{X_i}^{j,1} &= \begin{bmatrix} 0_{2\times21} & H_2 & 0_{2\times6i} & H_3 & 0_{2\times6(N-i-1)} \end{bmatrix} \end{aligned} \tag{22}$$

where, $H_{X_i}^{j,0}$ and $H_{X_i}^{j,1}$ represents the Jacobian of $r_i^{j,0}$ and $r_i^{j,1}$ with respect to error-state. And H_1, H_2, H_3 are derived respectively by:

$$\begin{aligned} H_1 &= \begin{bmatrix} J_i^{j,0}\left[{}_i^{C^0}\hat{p}_{f_j}\right]_\times & -J_i^{j,0}\,{}_G^{C^0}\hat{R} \end{bmatrix} \\ H_2 &= \begin{bmatrix} J_i^{j,1}\left[{}_i^{C^1}\hat{p}_{f_j}\right]_\times & -J_i^{j,1}\,{}_{C^0}^{C^1}\hat{R} \end{bmatrix} \end{aligned}$$

$$H_3 = \left[J_i^{j,1} {}^{C_1}_{C_0}\hat{R} \left[{}^{C_i^0}\hat{p}_{f_j} \right]_\times \; -J_i^{j,1} {}^{C_i^1}_{G}\hat{R} \right]$$

where $J_i^{j,0}$ and $J_i^{j,1}$ are defined as:

$$J_i^{j,k} = \frac{1}{\left({}^{C_i^k}\hat{Z}_j\right)^2} \begin{bmatrix} {}^{C_i^k}\hat{Z}_j & 0 & -{}^{C_i^k}\hat{X}_j \\ 0 & {}^{C_i^k}\hat{Z}_j & -{}^{C_i^k}\hat{Y}_j \end{bmatrix}, (k = 0,1)$$

Similar to original S-MSCKF [2], $H_{f_i}^j$ represents the Jacobian with respect to the error of feature coordinate. $H_{X_i}^j$ represents the Jacobian with respect to error-state. The core point in this paper is, different with S-MSCKF, in the Jacobian of reprojection error in right camera with respect to error-state, the sub-Jacobian of the reprojection error in right camera with respect to the error-state of stereo extrinsic E_1 is a non-zero block. It just models the reprojection error with respect to E_1. During state update, the E_1 will be calibrated online iteratively. n_i^j represents observation noise of j^{th} feature in i^{th} pose. We can stack Eq. (22) of all the observations with respect to the same feature:

$$r^j = z^j - \hat{z}^j \approx H_X^j \tilde{X} + H_f^j {}^G\tilde{p}_{f_j} + n^j \tag{23}$$

As EKF state variables are formulated regardless of feature coordinates, we can project Eq. (23) into the left null space of H_f^j, and marginalize the formula of feature error [14]:

$$r_o^j = V^T r^j = V^T(z^j - \hat{z}^j) \approx V^T H_X^j \tilde{X} + n_o^j \tag{24}$$

where, V represents the left null space of H_f^j, $n_o^j = V^T n^j$. Hence, Eq. (24) becomes the same as standard EKF update, and QR decomposition can be employed to accelerate the standard EKF update [1].

Similar to original S-MSCKF [2], the Observability Constrained EKF [15] is applied in our method for maintaining the consistency of the filter. And the strategy of feature update also comes from S-MSCKF.

D. Vision Frontend

In our implementation, for efficiency, FAST [16] corners are extracted as landmarks. Similar to [2-4], the KLT optical flow algorithm [17] is employed in feature matching of front and rear frames, as well as left and right frames. In stereo matching, essential matrix constraint is used to eliminate outliers. Different from [2-4], since stereo extrinsic parameters are calibrated online in this work, the stereo extrinsic parameters used in the frontend will also be time-varying. Since the initial extrinsic parameters may be inaccurate, the outlier rejection algorithm may incorrectly remove inliers during the initial phase of system start-up. Therefore, the constraint of outlier rejection using essential matrix relation should be weakened during the initial period of system startup to prevent serious errors. After the system runs for a period of time (i.e. 30 seconds in this paper), the essential matrix constraint could be set to the normal threshold.

IV. EXPERIMENTS

Two experiments are performed to evaluate the proposed algorithm. Firstly, we compare our method with state-of-the-art stereo VIO [2-4] on EuRoC dataset and a large scale dataset. Secondly, another experiment is performed with the stereo extrinsic containing initial noise to show the robustness and the validity of the proposed algorithm. All of the following algorithms run on Intel i9-9900k (3.6GHZ) desktop platform.

A. Dataset

EuRoC dataset is a visual-inertial dataset [18] produced by ASL team of ETH. Collected by UAV, the dataset includes stereo images of 20 FPS and IMU data of 200 Hz. It also provides ground truth trajectories from Leica MS50 lidar and Vicon motion capture system. The dataset consists of three scenarios and 11 sequences. Five of them are randomly selected for comparison.

Our large scale dataset includes 30 Hz stereo images and 500 Hz IMU collected by Mynteye S1030 camera shown in Fig. 2. The cameras is calibrated by Kalibr toolkit [19], and the ground truth is collected by 5Hz GPS of UBLOX NEO-M8N.

Our real-world dataset contains two scenes. The first dataset is the outdoor flight scene of UAV at 10m, 25m and 30m altitude. The horizontal trajectory distances are 1km, 1.1km and 2.1km separately, and the trajectories look like rectangles. The second dataset is an outdoor hand-held scene with a total distance of 1.5km. Therefore, all sequences are from large scale scenes. Fig. 3 shows sample images of the self-collected dataset.

B. RMSE comparison

RMSE, root mean square error, is a popular measurement to evaluate estimation accuracy. In the experiment, we compare proposed method with S-MSCKF and VINS-Fusion. The former is also based on MSCFK framework which cannot estimate stereo extrinsic online. The latter is an optimization based stereo VIO, which can estimate extrinsic between IMU and every camera. For fairness, we turn off the loop closure mode of VINS-Fusion.

In some cases, as the proposed algorithm has a relatively large initial threshold in frontend, we only compare trajectories after 30s.

Figure 2. The device we used for our dataset. It contains oblique top-down global shutter stereo camera(AR0135, 30Hz) with 752×480 resolution and it contains a build-in IMU (ICM20602, 500Hz).

1) In EuRoC dataset

In Table 1, when the initial stereo extrinsic is normal, VINS-Fusion performs the best, and our method performs similarly to original S-MSCKF. Although VINS-Fusion has higher accuracy, it consumes more computational resource because of too many variables optimized at the same time (see the detail comparison in [2]), and the average CPU load on our machine is about 146%. On the contrary, the proposed method is filter-based. Thus, it has the advantage of both high efficiency and lightweight. The average CPU load of our method is only about 57%, which is less than 1/2 of VINS-Fusion. Our method is similar with S-MSCKF in terms of CPU load.

In Table 2, as expected, benefited from the online stereo extrinsic calibration, the estimation results with the initial noise in stereo extrinsic of our algorithm and VINS-Fusion are not degraded significantly compared with no noise situations. However, without stereo extrinsic estimation, S-MSCKF performs badly or even diverges.

2) In large scale environment

In large scale environment, the estimation accuracy of proposed method is similar with VINS-Fusion, both of which are superior to S-MSCKF. Especially in the handheld data (Fig. 4), because there is no stereo baseline estimation, the estimated scale of S-MSCKF has a large error. It indicates that in the large scale environment, online stereo extrinsic estimation is crucial for scale estimation.

Similarly, with stereo extrinsic containing initial noise, the proposed algorithm and VINS-Fusion still work well in most cases, but S-MSCKF diverges. Hence, the robustness of our method is validated.

C. Stereo extrinsic estimation result

As can be seen from Table 3, with different perturbations to X and Y direction of the initial stereo extrinsic parameters, our method can converge to be approximately the same as the off-line calibration results. Specifically, for errors in translation in X or Y axis, most of the final errors are limited to below 0.5 mm. For errors in any direction of rotation, the final error is controlled under 0.1 degree. It should be noted that in Z axis of translation, all final errors are around 5 mm, including the case with normal initial stereo extrinsic parameters. An intuitive explanation is that the Z axis of stereo camera device is aligned with UAV heading direction in EuRoC dataset. Almost all the time UAV moves towards the heading direction, which leads to a bigger error in the estimated offset along the depth direction.

Fig. 5 shows that, with different initial artificial perturbations, the estimated translation in X axis and rotation in yaw direction between two cameras change with time. The figure shows that in about 30 seconds the estimated translation in X axis converges, and in 5 seconds the estimated rotation in yaw converges. These results indicate that the proposed algorithm can effectively estimate the stereo extrinsic in a timely manner.

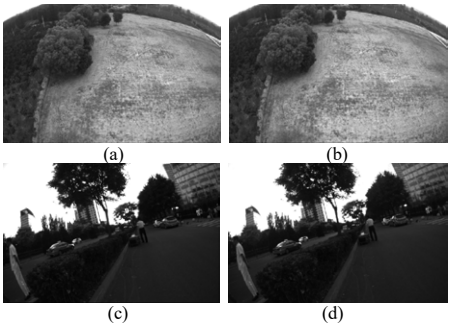


Figure 3. Sample images of large scale dataset. (a) and (b) are the images when the UAV is flying on 30 meters height. (c) and (d) are the images when the device is held by hand.

TABLE I. RMSE (m) comparison with normal initial stereo extrinsic. For EuRoC dataset, only trajectories after 30s were considered.

Data sequences	VINS-Fusion	S-MSCKF	Our Method
MH_03	0.080	0.211	0.223
MH_04	0.110	0.373	0.315
V1_03	0.129	0.260	0.195
V2_01	0.079	0.110	0.091
V2_02	0.035	0.139	0.163
UAV_10m	2.935	4.417	3.737
UAV_25m	4.068	4.674	4.768
UAV_30m	15.976	19.328	13.866
Hand_Held	14.223	41.969	9.480

TABLE II. RMSE (m) comparison with bad initial stereo extrinsic (added 2 deg. error in Z axis for rotation and 5mm error in baseline for translation). Only trajectories after 30s were considered, as it took time to estimate appropriate stereo extrinsic for filter-based method.

Data sequences	VINS-Fusion	S-MSCKF	Our Method
MH_03	0.087	-	0.302
MH_04	0.102	1.659	0.337
V1_03	0.195	0.585	0.235
V2_01	0.154	0.724	0.127
V2_02	0.067	0.454	0.165
UAV_10m	2.950	-	4.930
UAV_25m	4.076	-	11.213
UAV_30m	-	-	-
Hand_Held	18.610	-	24.861

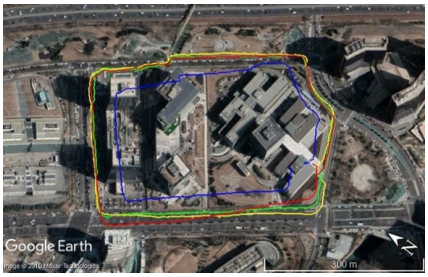
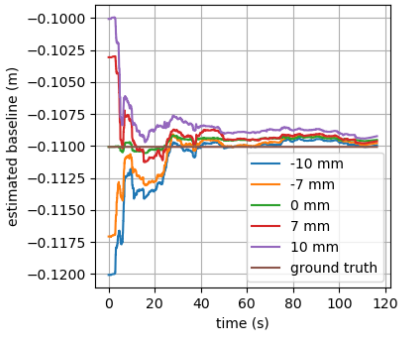


Figure 4. The estimated trajectories with good initial stereo extrinsic in hand held outdoor environment aligned to Google Map. VINS-Fusion (red), S-MSCKF (blue), our method (yellow) and GPS (green).

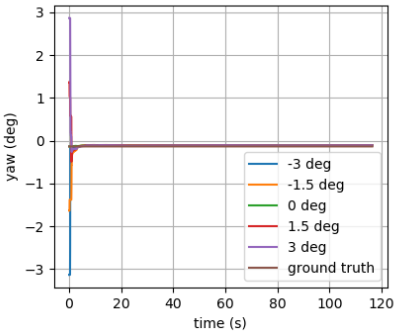
V. CONCLUSION

In this paper, we have presented an approach for online estimation of stereo extrinsic parameters based on S-MSCKF framework. The key component of our formulation is that the stereo extrinsic parameter E_1 is explicitly included in state variables, and the model between E_1 error and feature reprojection error is formulated. The resulting stereo VIO system significantly reduces the dependency on accurate offline stereo calibration. At the same time, the robustness and accuracy of the system are improved. Based on the experiments using EuRoC and real-world datasets, our scheme significantly outperforms the original S-MSCKF when there are perturbations to camera parameters. Especially, given inaccurate extrinsic parameters, our method can converge to an accurate estimation of extrinsic parameters over a few dozens of seconds. Since our method is filter-based, the computational requirement is much lower than those of optimization-based methods (e.g. VINS-Fusion), without significantly degrading the accuracy and robustness of the algorithm.

In future work, we will focus on real-time evaluation of the certainty of stereo extrinsic parameters.



(a)



(b)

Figure 5. With different initial artifical perturbations, estimated baseline (translation in X axis) and rotation in yaw between two cameras changing with time compared with offline calibration results using V2_02_medium data of EuRoC. (a) shows the translation and (b) shows the rotation.

TABLE III. Given different artificial initial perturbations, the final estimation errors of translation (mm) and rotation (Euler Angles in degree) between two cameras compared to the offline calibration ground truth. V2_02_medium data sequence of EuRoC is used in this experiment.

Errors in translation	**-10 mm**	**-7 mm**	**0 mm**	**+7 mm**	**+10 mm**
X	0.127	0.295	0.547	0.442	0.838
Y	0.248	-0.026	-0.204	-0.020	-0.040
Z	5.425	5.720	5.253	5.547	5.554
Errors in rotation	**-3 deg**	**-1.5 deg**	**0 deg**	**+1.5 deg**	**+3 deg**
Roll	0.096	0.097	0.093	0.096	0.087
Pitch	-0.078	-0.077	-0.080	-0.080	-0.086
Yaw	0.027	0.026	0.027	0.025	0.026

REFERENCES

[1] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," inProceedings 2007 IEEE International Conference on Robotics and Automation, pp. 3565 – 3572,2007.

[2] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar,C. J. Taylor, and V. Kumar, "Robust stereo visual inertial odometry forfast autonomous flight," IEEE Robotics and Automation Letters, vol. 3,pp. 965 – 972, April 2018.[3] M. Brossard, S. Bonnabel, and J.

[3] T. Qin, J. Pan, S. Cao, and S. Shen, "A general optimization-based framework for local odometry estimation with multiple sensors", arXiv preprint arXiv:1901.03638, 2019.

[4] T. Qin and S. Shen, "Online temporal calibration for monocular visual-inertial systems", in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2018, pp. 3662–3669.

[5] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments," Robotics and Automation (ICRA), 2012 IEEE International Conference on, pp. 957-964, 2012.

[6] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft MAV," Robotics and Automation (ICRA), 2014 IEEE International Conference on, pp. 4974-4981, 2014.

[7] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart and Paul Timothy Furgale. Keyframe-based visual-inertial odometry using nonlinear optimization. The International Journal of Robotics Research, 2015.
[8] Hansen P, Alismail H, Rander P, et al. Online continuous stereo extrinsic parameter estimation[C]//2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2012: 1059-1066.
[9] Ling Y, Shen S. High-precision online markerless stereo extrinsic calibration[C]//2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016: 1771-1778.
[10] N. Trawny and S. Roumeliotis, "Indirect Kalman filter for 6D pose estimation," University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep, vol. 2, 2005.
[11] Li, M. and Mourikis, A. I. (2011). Consistency of EKF-based visual-inertial odometry. Technical report, University of California Riverside. www.ee.ucr.edu/~mourikis/tech reports/VIO.pdf.
[12] J. Sola`, "Quaternion kinematics for the error-state kalman filter," CoRR, vol. abs/1711.02508, 2017.
[13] L. Clement, V. Peretroukhin, J. Lambert, and J. Kelly. The battle for filter supremacy: A comparative study of the multi-state constraint kalman filter and the sliding window filter. In Computer and Robot Vision (CRV), 2015 12th Conference on, pages 23–30, 2015.
[14] Y. Yang, J. Maley, and G. Huang, "Null-space-based marginalization: Analysis and algorithm," in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, Vancouver, Canada, Sep. 24-28, 2017, pp. 6749-6755.
[15] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Observability-constrained vision-aided inertial navigation," University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab, Tech. Rep, vol. 1, 2012.
[16] M. Trajkovic´ and M. Hedley, "Fast corner detection," Image and vision computing, vol. 16, no. 2, pp. 75–87, 1998.
[17] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision (ijcai)," Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81), pp. 674–679, April 1981.
[18] Burri M, Nikolic J, Gohl P, et al. The EuRoC micro aerial vehicle datasets[J]. The International Journal of Robotics Research, 2016, 35(10): 1157-1163.
[19] Rehder J, Nikolic J, Schneider T, et al. Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes[C]//2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2016: 4304-4311.

Learn with Noisy Data via Unsupervised Loss Correction for Weakly Supervised Reading Comprehension

Xuemiao Zhang[1], **Kun Zhou**[3], **Sirui Wang**[4], **Fuzheng Zhang**[4], **Zhongyuan Wang**[4], **Junfei Liu**[2] *
[1]School of Software and Microelectronics, Peking University, Beijing, China
[2]National Engineering Research Center for Software Engineering, Peking University, Beijing, China
[3]Renmin University of China, Beijing, China
[4]Meituan-Dianping Group
{zhangxuemiao,liujunfei}@pku.edu.cn,francis_kun_zhou@163.com
{wangsirui,zhangfuzheng}@meituan.com,wzhy@outlook.com

Abstract

Weakly supervised machine reading comprehension (MRC) task is practical and promising for its easily available and massive training data, but inevitablely introduces noise. Existing related methods usually incorporate extra submodels to help filter noise before the noisy data is input to main models. However, these multistage methods often make training difficult, and the qualities of submodels are hard to be controlled. In this paper, we first explore and analyze the essential characteristics of noise from the perspective of loss distribution, and find that in the early stage of training, noisy samples usually lead to significantly larger loss values than clean ones. Based on the observation, we propose a hierarchical loss correction strategy to avoid fitting noise and enhance clean supervision signals, including using an unsupervisedly fitted Gaussian mixture model to calculate the weight factors for all losses to correct the loss distribution, and employ a hard bootstrapping loss to modify loss function. Experimental results on different weakly supervised MRC datasets show that the proposed methods can help improve models significantly.

1 Introduction

Machine reading comprehension (MRC) (Rajpurkar et al., 2016) is a well-known NLP task, and has made significant progress in recent years (Yu et al., 2018; Devlin et al., 2019; Gong et al., 2020; Yuan et al., 2020). To learn a well-performed MRC system, large amount of human annotated data is required. However, human annotation is high-cost in real-world application, and it is hard to control the quality for some of hard instances. Recent approach (Joshi et al., 2017) utilized a distantly supervised method to collect the excerpts for answers. It greatly scales up the dataset and reduces the cost, but introduces more harmful noisy samples inevitably. There are many of approaches proposed to filter noise for question answering (QA) recently. Lin et al. (2018) and Lee et al. (2019a) adopted a paragraph selector to calculate confidences of paragraphs to help filter noisy ones before they are input into the main model. Niu et al. (2020) designed a submodel to generate labels to supervise the training of the selector. Back to MRC, Lee et al. (2019b) further proposed to generate labels for unlabeled samples, then train an extra *Refinery* model to refine the overall labels for multilingual MRC task with limited training data.

Admittedly, these multistage methods have achieved certain improvements, but rely heavily on the selector, retriever or refinery. The qualities of these complementary models are hard to be controlled, and make training difficult. In fact, we can explore another novel idea that exploits the essential characteristics of noise itself to help alleviate its effect for MRC task. Inspired by the idea of learning with noisy labels in image classification (Arazo et al., 2019), we explore and find that the loss distribution of weakly supervised MRC training data has inspiring characteristics. As shown in Figure 1 (a), at the beginning of training, losses of noisy samples are generally greater than losses of clean samples significantly. And in Figure 1 (b), during training, the losses of all samples roughly converge into two clusters according to values. In addition, we have noticed that without correction, noise tends to attract more attention due to

*Corresponding author: Junfei Liu (liujunfei@pku.edu.cn)

the produced larger loss, which causes the model optimized into wrong direction easily. We argue that it is one of the essential reasons why the performance will be hurt by much noise.

In this paper, our main idea for improving original models is to correct the loss distribution based on the above findings, which reduces losses of noisy samples thus avoiding fitting noise and pushes models to pay more attention to the supervision signals from clean data, as shown in Figure 2. Specifically, instead of modifying the original structures of previous well-performing models, we first choose to fit a 2-component Gaussian mixture model (GMM) to loss distribution unsupervisedly, then infer the probabilities of samples being clean or noisy through the posterior probability provided by GMM. Note that we have verified in Section 5.2 that the noise recognition accuracy can even exceed 90% by GMM. Based on the inferred results of GMM, we then automatically produce weight factors for all losses. Specifically, we assign larger factors to losses that have higher probabilities of being clean samples by GMM, while assign lowers ones to losses that are more likely to be noisy. Note that in the traditional case without correction, the weight factors of all losses can be regarded as an uniform distribution. In addition, we also propose to use the hard bootstrapping loss to replace standard cross-entropy loss to further correct loss values of the individual samples to further avoid fitting noise.

Our contributions are summarized as: (1) We explore the essential characteristics of noise in weakly supervised MRC from the perspective of loss distribution, and offer new ideas for this task and other related NLP tasks in weakly supervised manner; (2) We propose the hierarchical loss correction method to avoid fitting noise and strengthen the supervision from clean samples, which uses unsupervisedly fitted GMM to calculate weight factors for correcting loss distribution, and uses hard bootstrapping loss to modify loss function; (3) We conduct ample experiments on two types of multiple weakly supervised datasets, and experimental results show that the proposed method can improve models significantly.

2 Preliminaries

2.1 Problem Formulation

The typical machine reading comprehension (MRC) task focuses on learning a model $h_\theta(x)$ to answer a question q given the excerpt evidence e derived from excerpt set $\mathcal{E}$. The training set can be formalized into a set of triple examples $\mathcal{D} = \{(q_i, e_i, a_i)|i = 1, ..., N\}$, where N is the number of examples in $\mathcal{D}$, $q_i = \{w_1^{q_i}, w_2^{q_i}, \ldots, w_n^{q_i}\}$ is the question with n tokens, $e_i = \{w_1^{e_i}, w_2^{e_i}, \ldots, w_m^{e_i}\}$ is the excerpt evidence with m tokens, $a_i = \{w_i^{e_i}, w_{i+1}^{e_i}, \ldots, w_{i+s-1}^{e_i}\}$ is a substring from e_i, and defines the golden answer to q_i. Following Devlin et al. (2018) and Joshi et al. (2017), this task can be formulated as to predict an answer span, i.e., the start and end indices of answer a_i in excerpt e_i.

TriviaQA (Joshi et al., 2017) contains a distantly supervised MRC dataset, whose evidences are gathered automatically, with the assumption of distant supervision that the presence of the answer string in an evidence document implies that the document does answer the question. Formally, in the distantly supervised MRC task, e_i is set to a set of excerpts, and training data is formalized as $\mathcal{D}_{ds} = \{(q_i, \{e_{ij}\}_{j=1}^M, a_i)|i = 1, ..., N\}$, where M is the number of excerpts. Although all excerpts in the set contain answer strings, there is no guarantee that answers to questions will be derived from the excerpts. When aligned to standard MRC data, a sample of distant supervised data $(q_i, \{D_i^1, D_i^2, \ldots, D_i^M\}, a_i)$ can be expanded into M samples in standard format $\{(q_i, D_i^1, a_i), (q_i, D_i^2, a_i), \ldots, (q_i, D_i^M, a_i)\}$. Obviously this automated operation can easily obtain a large number of training data, but inevitably introduces a lot of noise, which will hurt the model's performance.

In this paper, we consider such a more common and general weakly supervised MRC scenario, which extends from the distantly supervised MRC task (Joshi et al., 2017): in the training set $\mathcal{D}$, both the excerpts and the answer spans may be noisy. That is, not only do the excerpt e_i not guaranteed to provide the evidence to answer the question q_i, but the answer span a_i itself is likely to be noise. Anyway, $x_i \in \mathcal{D}$ is a noisy sample when excerpt evidence e_i or answer span a_i is noisy. We focus on improving the models on weakly supervised MRC training data.

2.2 Empirical Explorations

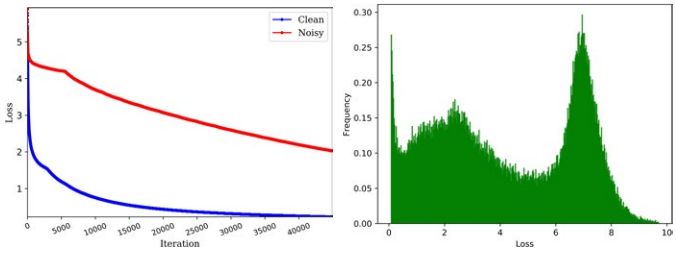
Typical MRC models usually learn the model parameters θ by minimizing the following loss function:

$$\mathcal{L} = -\sum_{i=1}^{N} log(P_{s_i}^1) + log(P_{e_i}^2) = -\sum_{i=1}^{N} y_i^T \log(P(a_i|e_i, q_i)) = -\sum_{i=1}^{N} y_i^T \log(h_\theta(x_i)) \quad (1)$$

where s_i and e_i of answer a_i are the start and end positions in excerpt e_i for sample x_i. $P_{s_i}^1$ and $P_{e_i}^2$ are the probabilities of the starting and ending position, respectively. y_i defines the label of the start and end indices. $h_\theta(x)$ defines the softmax probability produced by the model.

Taking Eq. (1) as the loss function, we train MRC models on weakly supervised datasets and record the entire loss convergence process, and collect all samples' losses computed by a trained model instance, as shown in Figure 1. From Figure 1(a), we can find that in the early stages of training, noise samples usually lead to significantly larger losses than clean samples. And from Figure 1(b), the losses of the entire dataset can be roughly divided into two clusters, we argue that the cluster with larger mean loss value corresponds to noisy samples, and conversely the other corresponds to clean samples.

These observations intuitively suggest that we can use a 2-component mixture model to unsupervisedly fit the overall loss distribution, where two independent components correspond to the loss distributions caused by noise and clean data, respectively. During training, we can reasonably correct the loss distribution before the loss back propagation by using the mixture model to infer whether the losses come from noise or clean data, thereby reducing disturbance from noise and pushing the model to pay more attention to the supervision signals from clean data. It is worth noting that the entire process does not use any additional supervision signals, but it gives the model much additional important information.



(a) (b)

Figure 1: Analysis of loss characteristics. (a): Comparison of loss convergence processes when training on original SQuAD data and noisy SQuAD data with 80% noise; (b): Frequency distribution histogram of losses obtained by infering all samples of distantly supervised TriviaQA data using a model instance.

3 Methodology

We propose hierarchical loss correction strategy to avoid fitting noise and enhance supervision signals from clean samples. The overall framework of the proposed methods is shown in Figure 2. We first *model loss* by fitting a GMM, then perform *loss correction* operation before back propagation.

3.1 Modeling Loss

Based on observations in Section 2.2, we can effectively infer whether a sample is more likely to be clean or noisy by fitting a probability distribution model to the losses of all training data. Intuitively, we argue that losses corresponding to clean and noisy samples obey two independent probability distributions, respectively. Therefore, losses of all training samples obey a mixture probability distribution composed of the above two distributions. We employ the widely used unsupervised GMM to fit the losses, since

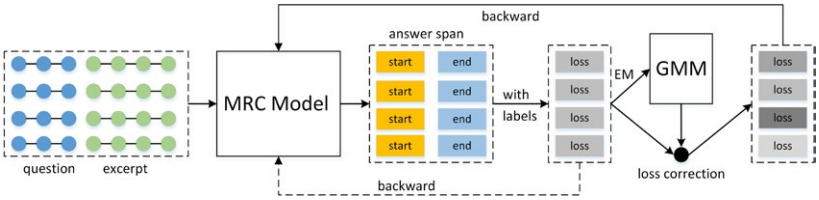


Figure 2: The framework of the proposed methods. Original losses are computed by aligning the predictions with the ground truth. A GMM is fitted to them to give the posterior probabilities to compute the corrected loss distribution for actual back propagation. original losses are used during pretraining.

loss histogram in Figure 3 shows Gaussian distribution is suitable, which has good mathematical properties. Specifically, we use 2-component GMM to fit the loss distributions of clean and noisy samples, respectively. Next, we introduce how to fit GMM to losses unsupervisedly and use it to model noise.

We assume that the observed losses $\boldsymbol{l} = \{l_i\}_{i=1}^{N}$ can be generated by a GMM θ_G:

$$P(\boldsymbol{l}|\theta_G) = \sum_{i=1}^{K} \alpha_k \phi(\boldsymbol{l}|\theta_k) \tag{2}$$

where $\theta_G = (\alpha_1, \alpha_2, \ldots, \alpha_K; \theta_1, \theta_2, \ldots, \theta_K)$, θ_k are parameters of the k-th Gaussian component and α_k are mixing coefficients for the convex combination of each individual probability density function (PDF) $p(\boldsymbol{l}|\theta_k)$. We employ the Expectation Maximization (EM) algorithm to fit GMM to the observed losses.

Specifically, we define the latent variables $\hat{\gamma}_{jk}$ to be the posterior probability of the point l_j having been generated by mixture component θ_k, where $j = 1, 2, \ldots, N, k = 1, 2, \ldots, K$. In the E-step we fix the parameters α_k, θ_k and update the latent variables using Bayes rule:

$$\hat{\gamma}_{jk} = E(\gamma_{jk}|\boldsymbol{l}, \theta_G) = P(\gamma_{jk} = 1|\boldsymbol{l}, \theta_G) = \frac{\alpha_k \phi(l_j|\theta_k)}{\sum_{k=1}^{K} \alpha_k \phi(l_j|\theta_k)} \tag{3}$$

And given fixed $\hat{\gamma}_{jk}$, the M-step estimates parameters $\hat{\mu}_k$, $\hat{\delta}_k$ of the Gaussian distribution, and $\hat{\alpha}_k$ as:

$$\hat{\mu}_k = \frac{\sum_{j=1}^{N} \hat{\gamma}_{jk} l_j}{\sum_{j=1}^{N} \hat{\gamma}_{jk}}; \hat{\sigma}_k = \frac{\sum_{j=1}^{N} \hat{\gamma}_{jk} (l_j - \mu_k)^2}{\sum_{j=1}^{N} \hat{\gamma}_{jk}}; \hat{\alpha}_k = \frac{\sum_{j=1}^{N} \hat{\gamma}_{jk}}{N} \tag{4}$$

Repeat the above calculation until convergence or the iterations exceeds the maximum limitation.

Given a fitted GMM, we can effectively model the losses. Specifically, we calculate the probability of a sample being clean or noisy through the posterior probability as follows:

$$p(\theta_k|l_i) = \frac{p(\theta_k) p(l_i|\theta_k)}{p(l_i)} \tag{5}$$

We use the component θ_k with the smallest mean μ_k to represent the loss distribution of clean samples.

3.2 Hierarchical Loss Correction

We further consider correcting losses to avoid fitting noise. The correction process includes hierarchical operations, fine-grained loss function correction and high-level loss distribution correction.

Since standard cross-entropy (CE) in loss Eq. (1) is ill-suited to deal with noisy samples because the model will exploit wrong knowledge from noisy samples (Zhang et al., 2017), it is replaced with the hard bootstrapping loss (Reed et al., 2015) to correct the training objective and alleviate the disturbance of noise, which deals with noisy samples by adding a perception term to CE loss:

$$\mathcal{L}_{hard} = -\sum_{i=1}^{N} (\beta y_i + (1-\beta) z_i)^T) log(h_i) \tag{6}$$

where $z_i := \mathbb{1}[k = \arg\max h_j, j = 1, \ldots, N]$, β weights the model prediction z_i in the loss function. Following Reed et al. (2015), we set $\beta = 0.8, \forall i$.

We further propose to correct the loss distribution based on the posterior probability by GMM. Generally, neural MRC models are trained by stochastic gradient descend (SGD) approach, in which losses directly affect the calculation of gradients, which in turn affect the optimization process, so that samples with larger losses have more influence. Traditional models trained on clean data try to fit all losses with the intuition that the under-fitting leads to large losses. But when training on noisy data, we argue large losses are more likely to be caused by noise and need to be corrected. We correct entire loss distribution by using GMM to infer the possibilities that samples are clean, and adopting a softmax operation to assign larger weight factors to the samples with higher probabilities and lower ones to others. The loss distribution correction operation with weight factors is given as:

$$\mathcal{L}_{correct} = \sum_{i=1}^{N} \frac{1}{Z} e^{\frac{p(k=k_c|l_i^{hard})}{T}} l_i^{hard} \tag{7}$$

where $Z = \sum_{j=1}^{N} e^{\frac{p(k=k_c|l_j^{hard})}{T}}$ is the normalization factor, $k_c = \arg\min(\theta_G.means)$ is the Gaussian component with the smallest value of mean parameter in GMM model θ_G, indicating that it is clean component fitted to the clean data, and T is the temperature parameter.

Algorithm 1: Loss correction process for reading comprehension question answering.

Input: Training epoch number K; training data size N; train triple samples $\{x_i\}_{i=1}^{N}$; GMM refitting frequency f; the size of mini-batch b.

initialize MRC model θ;
Pretrain θ with original losses by standard cross entropy;
for $k \leftarrow 1$ *to* K **do**
  if $k \ \% \ f == 0$ **then**
    Compute all losses $\boldsymbol{l}$ of all samples $\{x_i\}_{i=1}^{N}$ by Eq. (6);
    Fit GMM θ_G to all losses $\boldsymbol{l}$ using EM algorithm as Eq. (3) and Eq. (4);
    $k_c \leftarrow \arg\min(\theta_G.means)$ // choosing the Gaussian component with the smallest mean value to represent the distribution of clean data;
  for *mini-batch in batches of epoch* **do**
    Compute batch losses $\boldsymbol{l}_{hard}$ of the mini-batch samples $\{x_i\}_{i}^{b}$ by Eq. (6);
    Compute posterior probabilities $\{p(k = k_c|l_i)\}_{i=1}^{b}$ for $\boldsymbol{l}_{hard}$;
    Compute corrected batch losses $\boldsymbol{l}_{hard}^{c}$ by Eq. (7);
    Loss back propagation from $\boldsymbol{l}_{hard}^{c}$ and updata θ;

3.3 Overviews

In summary, the framework of the proposed methods is shown in Figure 2, and we train the improved moels according to Algorithm 1. In practice, we first pretrain the original model using standard CE. Then, we compute the bootstrapping losses, and fit a 2-component GMM to these losses using EM algorithm and record the clean Gaussian component with minimum mean value. In each training step, we compute batch losses of the batch samples and the probabilities of these samples being clean, then employ a softmax operation to compute the weight factors to further calculate the corrected losses. At the end of the step, we do back propagation based on the corrected losses.

4 Experimental Setup

4.1 Datasets

SQuAD. SQuAD (Rajpurkar et al., 2016) is a standard and high-quality MRC dataset. The annotators were asked to write more than 100,000 questions and select a span of arbitrary length from the given Wikipedia paragraph to answer the question. In practice, we use the SQuAD v1.1, and randomly select a certain percentage of samples to add noise to them. For each noisy sample, we randomly select a

continuous sequence of tokens from the evidence paragraph to replace the original label. Note that in this scenario, the answer is noisy. In order to fully explore the influence of noise, we generate 4 noisy training data, and their noise ratios are 0.2, 0.4, 0.6 and 0.8, respectively.

TriviaQA. TriviaQA (Joshi et al., 2017) is a collection of trivia question-answer pairs that were scraped from the web. We use their distantly supervised MRC dataset whose excerpt evidences are scraped from Wikipedia. We convert TriviaQA into a weakly supervised data format that conforms to the definition in the section 2.1. Note that, in this scenario, the evidence file is noisy. However, unlike the randomly created noise in squad, noise in TriviaQA is real in natural scenes.

4.2 Setup

Baselines. We use two widely used models (Cui et al., 2019; Lee et al., 2019b), and a shrunken model as the baselines. **BERT**: We modify a pre-trained uncased BERT (Devlin et al., 2018) model on a masked language task to MRC task by mapping the features extracted by BERT into the inferencing position logits to predict answer spans through a dense layer. **BiDAF**: Seo et al. (2016) proposed a multistage hierarchical process, which represents context at different levels of granularity, and uses a two-way attention flow mechanism to obtain query-aware context representation, we follow the implementation setting of original BiDAF. **BiDAF**$_m$: To explore the impact of model capacity on the proposed methods, we build a mini version of BiDAF, denoted BiDAFm, by reducing the amount of parameters; specifically, we set word dimension to 50 (original 100), char channel size to 20 (original 100), hidden size of LSTM to 35 (original 100), char channel width to 2 (original 5) and char dimension to 3 (original 8).

Evaluation Metrics. Following Chen et al. (2017) and Lee et al. (2019b), we use these two official evaluation metrics to evaluate our models, namely ExactMatch (EM) and F1 score. Among them, EM evaluates the percentage of prediction answers that exactly match one of the ground truth ones and F1 score can measure the average overlap between the prediction and ground truth answer. And we directly use the official evaluation script provided by SQuAD v1.1 for evaluation.

Settings. We implement the proposed methods by employing the loss correction strategies based on the above three baselines, including using a mixture probability distribution model to fit to losses of models, which in turn helps correct the loss distribution, and replacing the cross-entropy loss in Eq. (6) to the hard bootstrap loss which is more suitable for processing noisy data. Based on these settings, we retrain these new models in the same experimental environment. In practice, for mixture models, we use 2-component GMM, and its max iteration number is set to 100. We use Glove pretrained embeddings to initialize word embedding in BiDAF. We set β in hard bootstrapping loss to 0.8, set learning rate in BERT and BiDAF to 0.0005 and 0.001, respectively, and set temperature T to 1.0. We bounding the loss observations in $[\epsilon, 1-\epsilon]$ instead of $[0, 1]$ ($\epsilon = e-4$ in practice) to sidesteps this issue that EM algorithm will become numerically unstable when the observations are very near 0 and 1.

5 Results and Analysis

5.1 Experimental Results

Table 1 shows the evaluation results of the baselines and the improved models using the proposed methods on EM and F1 metrics. We can find that our methods make the original well-performed models achieve a further significant performance improvement on the real distantly supervised TriviaQA dataset. Among them, the improved model based on BERT improves by 13.9% and 10.0% on the EM and F1 respectively, and the improved model based on BiDAF$_m$ improves by 17.4% and 13.2%, respectively. It shows that the proposed methods can effectively improve the models training on noisy data. On noisy SQuADs with different ratios of noise, our methods can still significantly improve models. Taking SQuAD with 60% noise as an example, the improved model based on BiDAF has improved 10.42 percentage points (29.4%) and 9.50 points (21.1%) on EM and F1, respectively. The improved model based on BERT has improved 8.07 percentage points (20.6%) and 8.20 points (16.6%), respectively. It shows that the proposed methods can indeed help reduce the disturbance of noise on the model, and this ability can be clearly reflected on different data sets.

Model		SQuAD										TriviaQA	
		clean		noise-0.2		noise-0.4		noise-0.6		noise-0.8			
		EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
BiDAF_m	OR	60.19	71.87	58.13	69.53	54.08	65.99	38.92	50.79	5.07	7.38	17.41	22.24
	HB	-	-	58.50	70.23	54.47	66.21	42.81	52.05	6.40	8.92	19.22	23.32
	DCE	-	-	**59.09**	70.28	55.93	66.94	**47.23**	**57.89**	8.25	10.84	19.86	24.45
	DHB	-	-	58.61	**70.47**	**56.25**	**67.54**	43.71	52.95	**8.52**	**11.09**	**20.44**	**25.18**
BiDAF	OR	64.18	74.70	60.02	70.62	53.42	63.85	35.36	44.95	10.35	15.35	22.92	27.23
	HB	-	-	61.94	72.01	57.35	67.58	34.91	44.89	10.63	15.98	23.00	27.17
	DCE	-	-	63.25	73.15	57.75	68.46	**45.78**	**54.45**	11.14	15.97	**23.17**	**27.41**
	DHB	-	-	**63.36**	**73.89**	**59.13**	**70.38**	43.91	53.01	**12.16**	**17.12**	23.14	27.28
BERT	OR	69.56	79.08	61.36	72.48	53.13	64.10	39.08	49.44	15.49	24.60	25.65	30.95
	HB	-	-	62.37	73.16	54.22	64.82	43.74	54.03	17.75	25.84	26.24	31.70
	DCE	-	-	63.06	73.73	54.99	66.09	43.73	53.48	17.36	**26.62**	28.28	33.34
	DHB	-	-	**64.12**	**74.09**	**56.94**	**67.34**	**47.15**	**57.64**	**18.43**	26.09	**29.21**	**34.02**

Table 1: Evaluation results of different models under different loss correction strategies on two category of weakly supervised training sets. Among them, *OR* represents the original methods with cross entropy, *HB* represents methods using hard bootstrapping loss only, and *DCE* and *DHB* represent strategies of using loss distribution correction based on cross entropy and hard bootstrap loss, respectively.

Ablation Study. For each group of experiments, we report the experimental results of different models using original cross entropy loss and hard bootstrapping loss, and using high-level loss distribution correction with the two loss functions, respectively. From Table 1, we can find that: (1) Compared with using original cross entropy, the strategy of only correcting the loss function with hard bootstrapping loss can also improve models to a certain extent. (2) Both loss correction combination strategies have significant impacts on models' promotions. (3) The models using the loss distribution correction based on standard cross-entropy strategy has been effectively improved compared to the baselines, and some models using this strategy perform best in some scenarios, such as the BiDAF-based improved model trained on SQuAD with 60% noise. (4) But overall, the improved models using loss distribution correction based on hard bootstrap loss strategy will perform better, because the strategy attempts to provide cleaner loss signals by correcting both the loss values of the samples themselves and the loss distribution. Since there is no guarantee that adopting the combination strategy based on hard bootstrap loss will be better, we recommend to try both combination strategies if conditions permit, and choose the one that performs better, in the practice of applying the proposed methods.

5.2 How does GMM work?

We further analyze GMM's ability to distinguish between noisy and clean samples based on loss distribution unsupervisedly. First, independent of the noisy SQuAD sets for training, we randomly regenerate a series of test sets from original training set to evaluate GMM, which contain corresponding proportions noise and labels used to mark whether the samples are noise. Specifically, we regularly use the model in normal training process to output the loss corresponding to each sample in the corresponding test set, and use a new GMM instance to fit this loss distribution. Then use the fitted GMM to infer whether the sample is clean or noise. Along with training process, we record the best evaluation results of GMM.

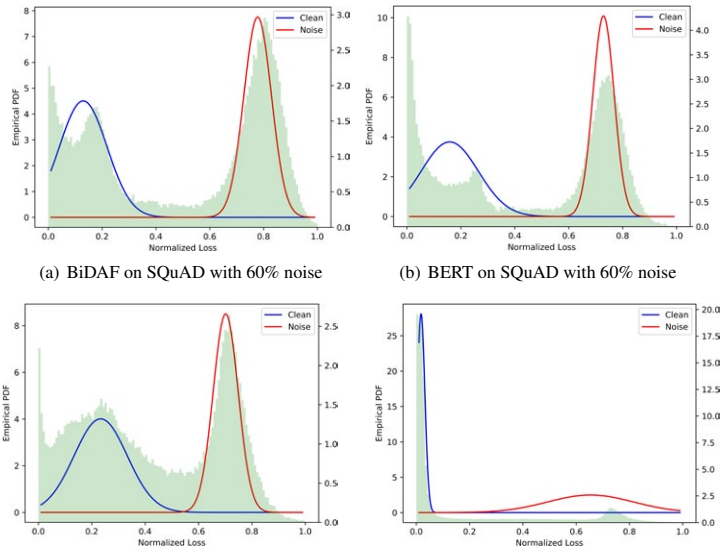
From Table 2, we can find that GMM can very effectively identify noise. On data sets with a noise ratio of 60% or less, BERT-based and BiDAF-based improved models can correctly identify more than 97% and 80% of noisy samples, respectively. And on the noisy data a noise ratio of 80%, the noise recognition rate still reaches 74%. This means that based on the observations in Section 2.2, GMM can provide so much extra useful information out of nothing to help improve the models. Specifically, the posterior probability given by GMM help to correct the loss distribution, thereby reducing the disturbance of noise, and push the model pay more attention to the supervision signals from clean data. We also note that the recognition rates of noise and clean data is a trade-off. Noise recognition and clean recognition are difficult to perform both well at the same time. However, in general, the recognition results of GMM are very effective in correcting the loss distribution, because as long as the attentions to clean samples are increased or the to noisy samples are reduced, the model can be optimized in a more correct direction.

Model		noise-0.2			noise-0.4			noise-0.6			noise-0.8		
		all	noise	clean	all	noise	clean	all	noise	clean	all	noise	clean
BiDAF$_{rn}$	OR	74.30	99.62	67.95	87.24	79.59	92.30	56.57	81.96	18.21	32.26	17.29	92.11
	DHB	54.54	99.77	43.21	87.50	79.23	92.96	72.52	80.20	60.91	33.62	17.20	99.31
BiDAF	OR	78.64	99.61	73.38	81.27	98.61	69.81	77.94	81.93	71.91	72.33	81.81	34.45
	DHB	60.47	99.82	50.60	87.76	80.12	92.81	78.17	81.21	73.58	30.31	17.91	79.90
BERT	OR	72.00	99.35	65.14	67.20	98.97	46.24	68.74	97.02	26.08	71.22	74.16	59.45
	DHB	76.71	99.62	70.96	72.78	98.86	55.58	76.23	98.27	42.95	68.07	74.74	41.36

Table 2: Accuracy of unsupervisedly identifying the noise in the training data of different noisy SQuAD with different noise rates by GMM obtained by fitting to the loss observations. Among them, *all* represents the overall accuracy, *noise*, and *clean* respectively are the proportion of noise samples and clean samples that are correctly identified.

5.3 Fit to Loss Distribution

In addition, we intuitively show how GMM fits the loss distribution, as shown in Figure 3. From Figure 3, we can find that the loss distribution of different models trained on different noisy data sets can be indeed roughly divided into two clusters, indicates that it makes sense to use a two-component mixture probability model to fit the loss distribution. Moreover, the Gaussian distribution is very universal, because it can basically fit loss clusters in various situations. Of course, the operators can explore or design a special distribution to replace the Gaussian distribution for specific scenarios in practice. Note that we focus more on the generalization ability of the Gaussian distribution in this paper.



(a) BiDAF on SQuAD with 60% noise

(b) BERT on SQuAD with 60% noise

(c) BiDAF on TriviaQA

(d) BERT on TriviaQA

Figure 3: Analysis of fitting a 2-component Gaussian mixture model to the losses computed by models (BiDAF and BERT) on different noisy data sets, where two clusters in the histogram correspond to two Gaussian components depicted by the red and blue curves, respectively.

5.4 Explore Other Mixture Model

From observations in Section 2.2, we can know that as long as a probability model can well fit the loss distribution, it can be used to participate in the construction of the mixture model. In addition to GMM, we also explore the Beta Mixture Model (BMM), which performs well in noisy image classification

Model	PDM	SQuAD								TriviaQA	
		noise-0.2		noise-0.4		noise-0.6		noise-0.8			
		EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
BiDAF$_m$	BMM	58.18	69.59	53.93	66.63	**47.39**	57.31	6.42	9.52	19.20	24.77
	GMM	**58.61**	**70.47**	**56.25**	**67.54**	47.23	**57.89**	**8.52**	**11.09**	**20.44**	**25.18**
BERT	BMM	62.89	73.75	55.19	65.85	43.27	51.84	**18.97**	**28.93**	27.38	32.41
	GMM	**64.12**	**74.09**	**56.94**	**67.34**	**47.15**	**57.64**	18.43	26.09	**29.21**	**34.02**

Table 3: Comparison results of employing different mixture models to improve BiDAF$_m$ and BERT on different noisy data sets.

tasks (Arazo et al., 2019). The beta distribution over a normalized loss $l \in [0, 1]$ is defined to have PDF: $p(l|\alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} l^{\alpha-1}(1-l)^{\beta-1}$, where $\Gamma(\cdot)$ is the Gamma function, and $\alpha, \beta > 0$ are parameters. Similarly, the mixture PDF is given by substituting the above into Eq. (5). Based on BiDAF$_m$ and BERT, we conduct comparison experiments on all noisy datasets. The experimental results are shown in Table 3. From Table 3, we can find that: (1) loss correction based on BMM can also bring a significant performance improvement, compared with the results in Table 1; (2) in most scenarios, GMM can help to achieve more significant improvements than BMM, indicating that GMM has obvious advantages in MRC task, and is very suitable for this task. It enlightens us that when there is no better choice, the Gaussian mixture model is a good solution, or serves it as a baseline to explore better models.

6 Related Work

Machine Reading Comprehension. Machine reading comprehension (MRC) (Rajpurkar et al., 2016) has received increasing attention recently, which requires a model to extract an answer span to a question from reference documents (Yu et al., 2018; Devlin et al., 2019; Liu et al., 2020; Zheng et al., 2020; Yuan et al., 2020). Owing to the rise of pre-training models (Devlin et al., 2018), a machine is able to achieve highly competitive results on classic datasets (*e.g.* SQuad (Rajpurkar et al., 2016)), even close to human performance. However, there is still a huge gap between high performance on the leaderboard and poor practical user experience, due to the noisy dataset, high-cost annotation and low resource languages. Recently, the more challenging distantly supervised MRC task, TriviaQA (Joshi et al., 2017) was proposed, in which the provided evidences are noisy and collected based on the distant supervision. (Yuan et al., 2020) proposed a multilingual MRC task to facilitate the study on low resource languages. (Lee et al., 2019b) focused on annotating the unlabeled data with heuristic method and refine the labels by an extra *Refinery* model for multilingual MRC task.

Learning with Noisy Labels. Recently, the great progress has been made on learning with noisy labels in image classification and question answering (QA) domains. Reed et al. (2015) and Ma et al. (2018) proposed a bootstrapping method to reconstruct loss function for noisy data combined with model predictions. Jiang et al. (2018) and Arazo et al. (2019) put forward an empirical assumption that samples with lower losses are clean, then separate the clean and noisy samples based on the loss distribution. For QA task, Lin et al. (2018) and Lee et al. (2019a) utilized an extra paragraph selector to filter noise by calculating confidences of paragraphs. Niu et al. (2020) further proposed a complementary model to generate labels to the paragraphs for training selectors supervisedly.

7 Conclusion

In this paper, we explore natural characteristics of noise from perspective of loss, and find in early stages of training, noisy samples usually result in significantly larger losses than clean samples. Based on the observation, we propose a hierarchical loss correction strategy to avoid fitting noise and strengthen supervision signals from clean samples by incorporating an unsupervisedly fitted GMM and modifying original loss funciton to hard bootstrapping loss. We conducted ample experiments on multiple weakly supervised MRC datasets. Experimental results show that the proposed methods can effectively help models to achieve significant improvements.

References

Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. 2019. Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning (ICML)*, pages 312–321, June.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada, July. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2019. Cross-lingual machine reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1586–1595, Hong Kong, China, November. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Hongyu Gong, Yelong Shen, Dian Yu, Jianshu Chen, and Dong Yu. 2020. Recurrent chunking mechanisms for long-text machine reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6751–6761, Online, July. Association for Computational Linguistics.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2304–2313, Stockholmsmässan, Stockholm Sweden, 10–15 Jul. PMLR.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019a. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy, July. Association for Computational Linguistics.

Kyungjae Lee, Sunghyun Park, Hojae Han, Jinyoung Yeo, Seung-won Hwang, and Juho Lee. 2019b. Learning with limited data for multilingual reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2840–2850, Hong Kong, China, November. Association for Computational Linguistics.

Yankai Lin, Haozhe Ji, Zhiyuan Liu, and Maosong Sun. 2018. Denoising distantly supervised open-domain question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1736–1745, Melbourne, Australia, July. Association for Computational Linguistics.

Dayiheng Liu, Yeyun Gong, Jie Fu, Yu Yan, Jiusheng Chen, Daxin Jiang, Jiancheng Lv, and Nan Duan. 2020. RikiNet: Reading Wikipedia pages for natural question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6762–6771, Online, July. Association for Computational Linguistics.

Xingjun Ma, Yisen Wang, Michael E. Houle, Shuo Zhou, Sarah M. Erfani, Shu-Tao Xia, Sudanthi N. R. Wijewickrema, and James Bailey. 2018. Dimensionality-driven learning with noisy labels. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 3361–3370. PMLR.

Yilin Niu, Fangkai Jiao, Mantong Zhou, Ting Yao, jingfang xu, and Minlie Huang. 2020. A self-training method for machine reading comprehension with soft evidence extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3916–3927, Online, July. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November. Association for Computational Linguistics.

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2015. Training deep neural networks on noisy labels with bootstrapping. In *International Conference on Learning Representations (ICLR)*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. 2018. Fast and accurate reading comprehension by combining self-attention and convolution. In *International Conference on Learning Representations*.

Fei Yuan, Linjun Shou, Xuanyu Bai, Ming Gong, Yaobo Liang, Nan Duan, Yan Fu, and Daxin Jiang. 2020. Enhancing answer boundary detection for multilingual machine reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 925–934, Online, July. Association for Computational Linguistics.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*.

Bo Zheng, Haoyang Wen, Yaobo Liang, Nan Duan, Wanxiang Che, Daxin Jiang, Ming Zhou, and Ting Liu. 2020. Document modeling with graph attention networks for multi-grained machine reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6708–6718, Online, July. Association for Computational Linguistics.

Syntactic Graph Convolutional Network for Spoken Language Understanding

Keqing He[1]*, **Shuyu Lei**[2], **Yushu Yang**[2], **Huixing Jiang**[2], **Zhongyuan Wang**[2]
[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]Meituan Group
{kqin,leishuyu}@bupt.edu.cn
{yangyushu,jianghuixing,wangzhongyuan02}@meituan.com

Abstract

Slot filling and intent detection are two major tasks for spoken language understanding. In most existing work, these two tasks are built as joint models with multi-task learning with no consideration of prior linguistic knowledge. In this paper, we propose a novel joint model that applies a graph convolutional network over dependency trees to integrate the syntactic structure for learning slot filling and intent detection jointly. Experimental results show that our proposed model achieves state-of-the-art performance on two public benchmark datasets and outperforms existing work. At last, we apply the BERT model to further improve the performance on both slot filling and intent detection.

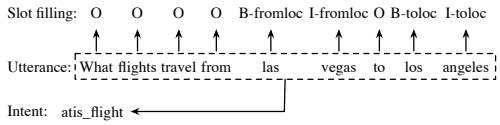
1 Introduction

Spoken Language Understanding (SLU) plays a vital role in a task-oriented dialogue system. Slot filling and intent detection (Tur and De Mori, 2011) are two major tasks for SLU as shown in Figure 1(a). Slot filling aims to obtain the semantic structure for the utterance. Meanwhile, intent detection annotates the categorical intent of the utterance.

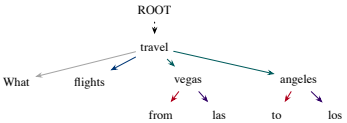
In typical pipeline methods, slot filling and intent detection are built separately. Slot filling is implemented as a standard sequence labeling task (Yao et al., 2014) and intent detection is built as a classification task (Lai et al., 2015), respectively. Essentially, slot filling and intent detection impact mutually. Therefore, more prior work (Hakkani-Tür et al., 2016; Liu and Lane, 2016; Goo et al., 2018; Li et al., 2018; Wang et al., 2018; Zhang et al., 2018a; E et al., 2019; Qin et al., 2019) implement two aforementioned tasks jointly as multi-task learning and achieve more promising results than those pipeline methods. However, most prior work utilize sequential model, such as recurrent neural network, to accumulate the contextual representation for each word to implement SLU with no consideration of prior linguistic knowledge. Intuitively, slot filling and intent detection rely on indicative contextual words for disambiguation and suffer from the degradation on wide contexts. Syntactic dependency parse tree as shown in Figure 1(b), which provides linguistic dependency relation among words, has been shown generally beneficial in various NLP tasks such as machine reading comprehension (Zhang et al., 2019), neural machine translation (Chen et al., 2018) and relation extraction (Zhang et al., 2018b). The major reasons are that the dependency parse tree can capture long-range relations between words and contain implicit clues for disambiguation.

To access a better SLU, we emphasize that SLU model should utilize the dependency representation as prior linguistic knowledge. In this paper, we propose a joint SLU model that applies a Graph Convolutional Network (GCN) over dependency trees to integrate the syntactic structure for joint learning slot filling and intent detection, where the GCN can pool information over arbitrary dependency structures efficiently, which has been proven in (Zhang et al., 2018b). Concretely, our proposed model encode the utterance and output a contextual word representation via a bi-directional LSTM (Hochreiter and Schmidhuber, 1997), then a GCN over dependency tree take contextual word representation as input to

*The work was done when the first author was an intern at Meituan Group. The first two authors contribute equally.



(a) An example of slot filling (BIO format) and intent detection for utterance "What flights travel from las vegas to los angeles".



(b) An example of dependency tree over the utterance "What flights travel from las vegas to los angeles".

Figure 1: An example of utterance in ATIS dataset.

obtain the syntactic structure representation for utterance. In addition, it is worth noting that existing dependency parsers are impossible to parse all the sentences exactly. Thus, a multi-head attention is utilized to fuse the syntactic representation with original contextual word representation against the errors caused by incomplete dependency parser, where the multi-head attention can supplement the syntactic representation with contextual word representation as a self-adaption manner. At last, the fused representation is applied to implement slot filling and intent detection jointly.

The experiments are conducted on two benchmarks SLU datasets: ATIS (Hemphill et al., 1990) and Snips (Coucke et al., 2018). The experimental results demonstrate that our proposed model outperforms the existing state-of-the-art approaches. At last, BERT model (Devlin et al., 2019), as a pre-trained model, is used to our proposed framework. The experimental results also show that our proposed model incorporated with BERT model can further improve the performance on both slot filling and intent detection.

The main contributions of this work are therefore include as follows: 1)We introduce a model that utilizes a GCN to integrate the syntactic structure for joint learning slot filling and intent detection, which to the best of our knowledge is the first work that syntactic structure and GCN are used to implement above two tasks jointly. 2) We utilize a multi-head attention to fuse the syntactic representation with contextual word representation against the errors caused by incomplete dependency parser. 3) We conduct our experiments on two public datasets, and our proposed model achieves new the state-of-the-art performance in overall accuracy metric.

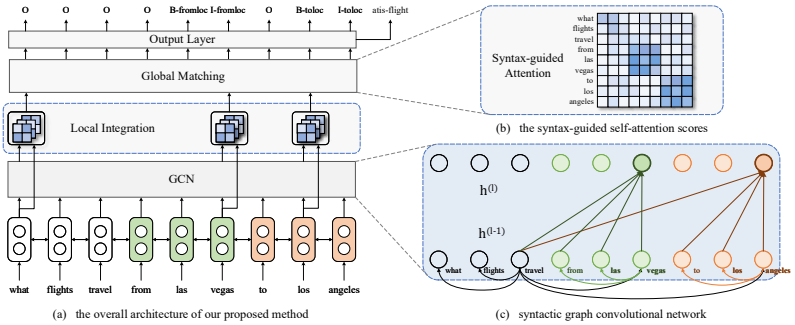
2 Methodology

In this section, we will describe our syntactic graph convolutional network for SLU tasks. The overall architecture of our model is demonstrated in Fig 2. We first use a BiLSTM encoder to obtain the contextual representation of an utterance. Then we perform multi-hop GCN propagation over the dependency tree initialized by the hidden states of the BiLSTM encoder to capture syntactic representation. Subsequently, we integrate the syntactic representation and the contextual hidden states via a feature aggregation layer. Finally, we pass the fused representation to the output layer for final predictions. Both slot filling and intent detection are optimized simultaneously via a joint learning scheme.

2.1 Notations

We now formally define the task of slot filling and intent detection. Let $\mathbf{X} = [x_1, \ldots, x_n]$ denotes a sentence, where n denotes the sequence length. We first use a syntactic parser to generate a dependency tree where each word represents a node. After obtaining a tree with n nodes, we can represent the graph structure with an $n * n$ adjacency matrix $\mathbf{A}$ where $A_{ij} = 1$ if there is an edge going from word x_i to word x_j. [1] Given the input sequence and the corresponding dependency tree, our goal is to predict the slot labels $\mathbf{o}^S = \left(o_1^S, \ldots, o_n^S\right)$ and the intent label o^I.

[1] We treat the dependency tree as an undirected graph, i.e.$\forall i, j, A_{ij} = A_{ji}$. We hypothesize that modeling edge directions and types does not offer additional discriminative power to the network because the GCN can capture adequately informative syntactic patterns for SLU. Besides, models with high complexity are prone to overfitting.



(a) the overall architecture of our proposed method

(b) the syntax-guided self-attention scores

(c) syntactic graph convolutional network

Figure 2: Spoken language understanding with a syntactic graph convolutional network. Fig (a) shows the overall architecture of our proposed method. Fig (b) displays the syntax-guided self-attention scores. Fig (c) shows one-layer detailed graph convolution computation for the word *"vegas"* and *"angeles"* for clarity. For local integration, we only show the computation of three timesteps. As we describe in the introduction, the syntactic structure lets a word focus more on its dependency words, such as $from \leftarrow vegas$ and $to \leftarrow angeles$. These syntactic constraints could enhance contextual representations to distinguish the departure city from the arrival city.

2.2 Syntactic Graph Convolutional Networks over Dependency Trees

The graph convolutional network(GCN) (Kipf and Welling, 2017) has been proved useful for encoding structural information in graphs. GCNs provide flexibility to represent diverse syntactic and semantic relationships between words. Essentially, GCN operates on a graph structure and compute representations for the nodes of the graph by looking at the neighborhood of the node. We can stack L layers of GCNs to account for neighbors that are L-hops away from the current node. Formally, in an L-layer GCN as Fig 2(c) shows, we denote the input vector as $h_i^{(l-1)}$ and output vector as $h_i^{(l)}$ where i represents the i-th node and l represents the l-th layer. Hence, the one-hop graph convolution operation can be written as $h_i^{(l)} = \sigma\left(\sum_{j=1}^{n} A_{ij} W^{(l)} h_j^{(l-1)} + b^{(l)}\right)$, where $W^{(l)}$ is a linear transformation, $b^{(l)}$ a bias term, and σ a nonlinear function (e.g., ReLU).

To initialize the first layer input vector $h^{(0)}$, we first feed the input word vectors into a BiLSTM network to generate contextualized representations. Note that our method is not limited to cooperate with BiLSTM, but any contextual encoder like ELMo(Peters et al., 2018), BERT(Devlin et al., 2019). We also conduct BERT based experiments for comparison. We will show empirically in Section 4.7 that both encoders substantially improve the performance over the original baselines.

Then, we perform the aforementioned graph convolution operation on dependency trees by converting each tree into its corresponding adjacency matrix $\mathbf{A}$, where $A_{ij} = 1$ if there is a dependency edge between words i and j. Adopted from (Zhang et al., 2018b), we also normalize the activations in the graph convolution before feeding it through the nonlinearity and adding self-loops to each node in the graph as $h_i^{(l)} = \sigma\left(\sum_{j=1}^{n} \tilde{A}_{ij} W^{(l)} h_j^{(l-1)} / d_i + b^{(l)}\right)$, where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ with $\mathbf{I}$ being the $n * n$ identity matrix and $d_i = \sum_{j=1}^{n} \tilde{A}_{ij}$ is the degree of token i in the resulting graph.

2.3 Feature Aggregation Mechanism: From Local To Global

After applying L-layer GCNs over dependency trees, we obtain the syntactic knowledge vector $h_i^{(L)}$ of each token x_i. To fuse syntactic representation and contextual representation, we propose the feature aggregation mechanism comprising of the local integration layer and global matching layer. The former builds strong interactions between syntactic vector and contextual vector of one word while the latter models connections at the overall utterance-level. The aggregation mechanism aims to integrate syntactic graph information and contextual representation and enable our model more robust to potential noise from the dependency parser.

Local Integration Given the syntactic representation $h_i^{(L)}$ and contextual representation $h_i^{(0)}$ of word

x_i, local integration intends to control how much information in $h_i^{(L)}$ and $h_i^{(0)}$ should be passed down. We employ a multi-head attention (Vaswani et al., 2017) to capture the relation between syntax and semantics. For the i-th word, we project $\mathbf{H}_i^{local} = \{h_i^{(L)}, h_i^{(0)}\}$ into the distinct key, value, and query representations, denoted K_j, Q_j and V_j for each head j. Then we perform the scaled dot product attention as $\text{Attention}\,(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$.

Then the outputs of all heads are concatenated and passed through a feed-forward layer followed by GeLU activations (Hendrycks and Gimpel, 2017) and a layer normalization. And we perform average pooling on the final outputs of local integration for each word, denoted as $\mathbf{H}' = \{h'_1, \ldots, h'_n\}$, where the syntactic representation and original contextual word representation are fused. Note that all the parameters of the local integration layer are shared for all the words. In Fig 2(a), we only show three network blocks for clarity.

Global Matching After local integration captures the relationship between syntax and semantics of each word, we propose a global matching layer to model connections between fused representations at the overall utterance-level. Similar to the local integration, we use another multi-head attention layer where we project $\mathbf{H}^{global} = \mathbf{H}' = \{h'_1, \ldots, h'_n\}$ into the the distinct key, value, and query representations. The syntactic information from GCN will guide a word to other words of syntactic importance in a sentence. To reduce the overall parameters and computational cost, we do not employ a stack of multiple identical blocks but only one multi-head attention layer for both local integration and global matching. The outputs of global matching will be forwarded into the final output layer.

2.4 Joint Optimization

Finally, we use the output hidden vectors of global matching to predict the slot types and intent. For slot types, we directly perform softmax operation over the hidden vector at each timestep. For intent prediction, we perform max pooling over all the words to obtain a fixed-length vector. Then the final vector is fed to the multi-layer perceptron (MLP) classifier, with one hidden layer, tanh activation, and softmax output layer. The entire model is trained by minimizing the sum of two cross-entropy losses in an end-to-end manner. The overall objective is formulated as $p\left(y^S, y^I|\mathbf{X}\right) = p\left(y^I|\mathbf{X}\right)\prod_{t=1}^{n} p\left(y_t^S|\mathbf{X}\right)$, where y^S, y^I are the softmax output probability of slots and intent respectively.

3 Experiments

3.1 Settings

To evaluate the proposed model, we conduct experiments on two public benchmark datasets, ATIS (Hemphill et al., 1990) and Snips (Coucke et al., 2018). ATIS contains audio recordings of flight reservations, and Snips is collected from the Snips personal voice assistant. For the syntactic parser, we adopt the Stanford parser from (Chen and Manning, 2014). The parser is not updated with our SLU model. For the SLU model, we use the glove embedding with the dimension of 300 and set the dropout rate as 0.1. L2 regularization is used with a rate of 1×10^{-3}. We use the Adam optimizer (Kingma and Ba, 2014) with the learning rate of 0.001. For the GCN model, we set the propagation layer num as 2 and treat the dependency tree as an undirected graph. All the results are reported on the test set after early stopping on the dev set.

3.2 Baselines

We compare our model with the existing baselines including: Joint Seq(Hakkani-Tür et al., 2016) proposes an RNN-based multi-task modeling approach for jointly modeling domain detection, intent detection, and slot filling. Attention BiRNN(Liu and Lane, 2016) leverages the attention mechanism to learn the relationship between slot and intent. Slot-Gated Atten(Goo et al., 2018) proposes the slot-gate to model the correlation of slot filling and intent detection. Self-Attentive Model(Li et al., 2018) proposes a novel self-attentive model with the intent augmented gate mechanism to utilize the semantic correlation between slot and intent. Bi-Model(Wang et al., 2018) proposes the Bi-model to consider the intent and slot filling cross-impact to each other. CAPSULE-NLU(Zhang et al., 2018a) proposes a capsule-based

Model	SNIPS			ATIS		
	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)
Joint Seq (Hakkani-Tür et al., 2016)	87.3	96.9	73.2	94.3	92.6	80.7
Attention BiRNN (Liu and Lane, 2016)	87.8	96.7	74.1	94.2	91.1	78.9
Slot-Gated Full Atten (Goo et al., 2018)	88.8	97.0	75.5	94.8	93.6	82.2
Slot-Gated Intent Atten (Goo et al., 2018)	88.3	96.8	74.6	95.2	94.1	82.6
Self-Attentive Model (Li et al., 2018)	90.0	97.5	81.0	95.1	96.8	82.2
Bi-Model (Wang et al., 2018)	93.5	97.2	83.8	95.5	96.4	85.7
CAPSULE-NLU (Zhang et al., 2018a)	91.8	97.3	80.9	95.2	95.0	83.4
SF-ID Network (E et al., 2019)	90.5	97.0	78.4	95.6	96.6	86.0
Stack-Propagation (Qin et al., 2019)	94.2	98.0	86.9	**95.9**	96.9	86.5
Our model	**94.8***	**98.2***	**87.6***	95.7	**97.2***	**86.9***

Table 1: Slot filling and intent detection results on two datasets. The numbers with * indicate that the improvement of our model over all baselines is statistically significant with $p < 0.05$ under t-test.

model with a dynamic routing-by-agreement schema to accomplish slot filling and intent detection. SF-ID Network(E et al., 2019) introduces an SF-ID network to establish multiple direct connections for the slot filling and intent detection to help them promote each other mutually. Stack-Propagation(Qin et al., 2019) proposes a Stack-Propagation framework that can directly use the intent information as input for slot filling, thus to capture the intent semantic knowledge. We report the experiment results of these models adopted from (Qin et al., 2019).

3.3 Overall Results

We evaluate the SLU performance about slot filling using F1 score, intent prediction using accuracy, and sentence-level semantic frame parsing using overall frame accuracy. We take the overall accuracy as the main evaluation metric since the metric considers the joint performance of both slot filling task and intent detection task. Table 1 displays the performance of our proposed model and baseline models on ATIS and Snips dataset. As shown in Table 1, we observe that our model outperforms all the baselines obviously on Overall (Acc). Specially, compared with the best prior joint work Stack-Propagation, we achieve 0.7% improvement on Overall (Acc) in the Snips dataset. Meanwhile, we achieve 0.4% improvement on Overall (Acc) in the ATIS dataset. In Snips dataset, our model outperforms the baseline models on all the evaluation metric. Although our model achieve 0.2% lower performance than Stack-Propagation on Slot (F1) in ATIS dataset, our model still outperforms Stack-Propagation on main evaluation metric, i.e. Overall (Acc), obviously. Above results indicate that our proposed model can improve the SLU performance significantly by integrating the syntactic structure with contextual information.

4 Qualitative Analysis

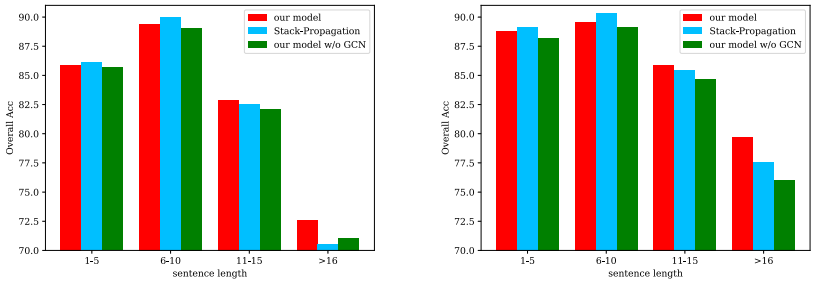
In this section, we present a detailed qualitative analysis on each component of our proposed model and provide certain typical cases to show the effectiveness of incorporating syntactic information. We first perform an ablation study to validate the effect of different modules of our model. Then we explore more methods of feature aggregation and demonstrate the superiority of our proposed local-to-global multi-head attention mechanism. Next, we give some typical cases of our model and baseline to show the effect of syntactic structure. Finally, we conduct experiment with BERT to verify that our model is more effective with pre-trained model.

4.1 Effect of Syntactic GCN

To verify the effectiveness of syntactic information delivered by the dependency tree, we conduct comparison experiments with the same architecture except for the way of constructing the adjacency matrix $\mathbf{A}$. As Table 2.1 shows, we experiment with two distinct adjacency matrices of all 0s and all 1s. The *Adj=0* model which fills the adjacency matrix with all 0s aims to disentangle GCN from our proposed model as a baseline. And the *Adj=1* model which fills the adjacency matrix with all 1s demonstrates the effect of the dependency tree.

Model	SNIPS			ATIS		
	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)
Adj=0*	93.3	97.7	84.6	94.9	96.5	84.9
Adj=1**	92.9	97.7	83.5	90.8	95.5	79.8
Syntax GCN	**94.8**	**98.2**	**87.6**	**95.7**	**97.2**	**86.9**

Table 2: Effect of Syntactic GCN. * indicates that the *Adj=0* model fills the adjacency matrix with all 0s. By contrast, ** indicates that the *Adj=1* model fills the adjacency matrix with all 1s. *Syntax GCN* represents our proposed syntactic GCN model where the adjacency matrix is filled with the dependency tree as described in Section 2.1.



(a) Split by sentence length in the ATIS test dataset

(b) Split by sentence length in the Snips test dataset

Figure 3: Test set performance with regard to sentence length for our proposed model, Stack-Propagation (Qin et al., 2019) and our model w/o GCN which fills the adjacency matrix with all 0s.

In the Snips dataset, compared to the *Adj=0* model, *Adj=1* gets a drop of 1.1% on Overall (Acc) while our model get 3.0% improvements. The similar results are also shown in the ATIS dataset. We hypothesize that this is because filling the adjacency matrix with all 1s essentially builds a fully-connected graph, which induces tremendous noise to the model. By contrast, integrating the syntactic information makes a word focus more on its dependency words as constraints. The experiment results confirm that incorporating syntactic dependency benefits the understanding of natural language.

4.2 Effect of Sentence Length

To understand what our syntax GCN model captures and how it differs from the previous baselines such as Stack-Propagation, we compare their performance over examples with different ranges of sentence length in the Fig 3. Specifically, for each model, we train it on the same training set and report their overall accuracy on examples with different sentence lengths of the test set.

Fig 3 shows that our proposed syntax GCN model outperforms Stack-Propagation with notable improvements at handling long sentences. We believe our model can better resolve issues of long-term dependencies via the explicit syntactic information. Besides, compared to the model w/o GCN, our model consistently achieves superior performance, which demonstrates the effectiveness of the feature aggregation layer.

4.3 Effect of Feature Aggregation

We further explore the benefits of our feature aggregation mechanism in our model. We conduct comparison experiments with the same architecture except for the feature aggregation layer. Table 3 shows the overall results of different feature aggregation methods, including Add, Concat, Gate, Full and our local-to-global multi-head attention mechanism. Given the syntactic representation $h_i^{(L)}$ and contextual representation $h_i^{(0)}$ of the i-th word, we define the Gate as $o_i = \alpha * h_i^{(L)} + (1-\alpha) * h_i^{(0)}$where $\alpha = W_1 h_i^{(L)} + W_2 h_i^{(0)}$, and the Full as $o_i = Concat([h_i^{(L)}; h_i^{(0)}; |h_i^{(L)} - h_i^{(0)}|; h_i^{(L)} * h_i^{(0)}])$.

Compared to the base RNN, all the aggregation methods, Add, Concat, Gate, Full, achieve $1\% \sim 2\%$

Model	SNIPS			ATIS		
	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)
RNN	90.7	96.9	80.7	94.3	95.3	82.5
Add	91.7	97.4	82.3	94.8	95.6	84.4
Concat	91.5	98.1	82.1	94.9	94.4	83.5
Gate	92.0	97.7	82.6	94.9	95.3	83.9
Full	91.5	97.9	81.6	94.9	94.2	83.5
Local Integration	93.4	97.9	85.9	95.2	96.1	85.1
Global Matching	94.1	98.0	86.7	95.3	97.1	86.2
Our model	**94.8**	**98.2**	**87.6**	**95.7**	**97.2**	**86.9**

Table 3: Effect of Feature Aggregation.

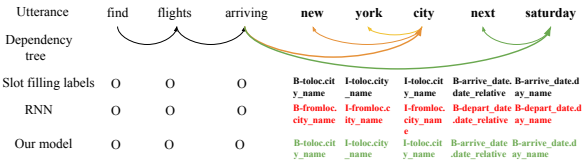


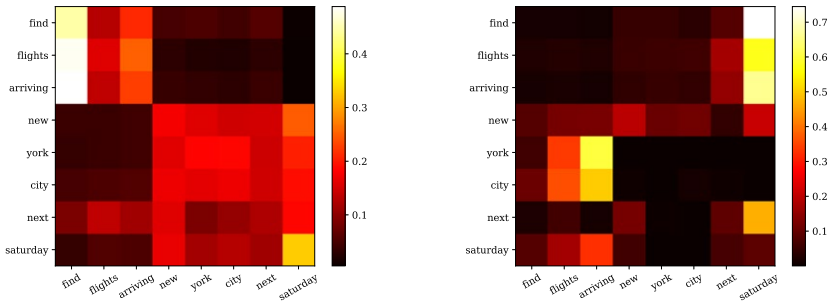
Figure 4: Case study of RNN and our model. The [GREEN] ([RED]) highlight indicates a correct (incorrect) tag.

improvements in both datasets, which demonstrates the effectiveness of incorporating syntactic structure via GCN. Further, our proposed local-to-global aggregation layer outperforms these methods with a statistically significant margin. The results confirm feature aggregation mechanism plays a vital role in the integration of contextual representation and syntactic information.

4.4 Case Study

We display two samples from basic RNN and our model in Fig 4. Given the same input *"find flights arriving new york city next saturday"*, RNN can not distinguish the *from_loc* from *to_loc* because it can not explicitly model the relationships between *new york city* and *arriving*. By contrast, our model leverages the syntactic structure to make *new york city* focus more on its dependency word *arriving*. This example illustrates the syntactic structure could enhance the contextual representation to facilitate the SLU tasks by modeling direct relations between words.

4.5 Visualization Analysis



(a) The model without syntactic GCN (b) Our proposed model

Figure 5: Visualization of attention distributions of the self-attention layer of global matching in our syntactic GCN model(right) and the variant without GCN(left).

To have a quick grasp of how syntactic information works, we perform visualization analysis of at-

tention distributions of the self-attention layer of global matching in our syntactic GCN model and the variant without GCN, as shown in Fig 5. Weights of attention are selected from the first head of the self-attention layer. After integrating syntactic knowledge, the word "city" focuses more on its dependency word "arriving", which convincingly indicates that "new york city" is an entity of arrival city but departure city. The visualization confirms that syntactic knowledge makes a word attentively select the relevant words and enhance contextual representations to distinguish subtle differences. Compared to (Zhang et al., 2019) which restrains the scope of attention only between word and all of its ancestor head words, we incorporate the syntactic dependency tree as a soft mask. We believe this soft mask can alleviate errors caused by incomplete dependency parser.

4.6 Ablation Study

Model	SNIPS			ATIS		
	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)
RNN	90.7	96.9	80.7	94.3	95.3	82.5
GCN*	83.6	97.4	66.8	81.9	95.1	54.5
RNN+GCN**	91.7	97.4	82.3	94.8	95.6	84.4
RNN+GCN+Local Integration	93.4	97.9	85.9	95.2	96.1	85.1
RNN+GCN+Global Matching	94.1	98.0	86.7	95.3	97.1	86.2
Our model	**94.8**	**98.2**	**87.6**	**95.7**	**97.2**	**86.9**

Table 4: Performance of different model variants. * indicates that the *GCN* model initializes the first GCN layer inputs $h^{(0)}$ with word embeddings. ** indicates that the *RNN+GCN* model simply sums contextual embeddings and GCN outputs.

To study the effect of each component of our method, we conduct ablation analysis (Table 4). In the ATIS and Snips dataset, the basic RNN model achieves 82.5 and 80.7 on overall accuracy respectively, which are much higher performance than vanilla GCN, 54.6 and 66.8. These results indicate that SLU tasks need contextual word representation, especially for slot filling task, while the syntactic structure could enhance the RNN model as supplementary knowledge. We can see that the simple RNN+GCN achieves 1.9% improvements in the ATIS dataset and 1.6% improvements in the Snips dataset.

On the other hand, although simply inducing syntactic structure(RNN+GCN) helps improve the basic RNN model, both Local Integration and Global Matching further improve the whole performance. We can see that Local Integration and Global Matching achieve 0.7% and 1.8% improvements in the ATIS dataset, 2.4% and 4.4% improvements in the Snips dataset, compared to the RNN+GCN. The full local-to-global aggregation further achieves 2.5% and 5.3% improvements respectively. The results demonstrate the effectiveness of the feature aggregation mechanism since the syntactic representation and contextual word representation can complement each other. Hence, our proposed local-to-global multi-head attention achieves the best performance.

4.7 Effect of BERT

Model	SNIPS			ATIS		
	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)
Our model	94.8	98.2	87.6	95.7	97.2	86.9
Intent detection (BERT)	-	97.8	-	-	96.5	-
Slot filling (BERT)	95.8	-	-	95.6	-	-
BERT SLU (Chen et al., 2019)	97.0	98.6	92.8	96.1	97.5	88.2
Stack-Propagation + BERT (Qin et al., 2019)	97.0	**99.0**	92.9	96.1	97.5	88.6
Our model + BERT	**97.1**	**99.0**	**93.0**	**96.2**	**97.8**	**88.7**

Table 5: The SLU performance on BERT-based model on two datasets.

Considering the performance with the fine-tuning approach, we also conduct experiments that we replace the contextualized BiLSTM by the BERT (Devlin et al., 2019) in our framework and keep the same architecture in rest of our model. The results of BERT model on ATIS and SNIPS datasets are shown in Table 5. From the Table 5, we can observe our model utilizing BERT achieves a new state-of-the-art performance. These results indicate a strong pre-trained model can further improve the performance for our model on SLU tasks. Our model + BERT outperforms Stack-Propagation + BERT which indicate that our framework is more effective with BERT than baseline models.

5 Related work

Slot filling and intent detection are two major tasks for SLU. Recently, the typical pipeline methods build slot filling and intent detection separately, where slot filling is implemented as a standard sequence labeling task (Yao et al., 2014) and intent detection is built as a classification task (Lai et al., 2015), respectively. More recent work (Hakkani-Tür et al., 2016; Liu and Lane, 2016; Goo et al., 2018; Li et al., 2018; Wang et al., 2018; Zhang et al., 2018a; E et al., 2019; Qin et al., 2019) implement slot filling and intent detection as a joint model to eliminate the error propagation without any linguistic knowledge. Instead, our work apply the linguistic knowledge (i.e. dependency tree) as a prior to guide the learning slot filling and intent detection jointly.

Dependency tree, as a important linguistic knowledge, is applied to recent natural language processing tasks. In relation extraction and machine translation, many studies (Xu et al., 2015; Liu et al., 2015; Miwa and Bansal, 2016; Chen et al., 2018) have show that the dependency trees can capture long-distance relations effectively. In machine reading comprehension, (Zhang et al., 2019) use syntax to guide the text modeling by incorporating explicit syntactic constraints into attention mechanism for better linguistically motivated word representations and achieve promising results on both SQuAD 2.0 and RACE datasets. Inspired by (Zhang et al., 2019), our work utilize the dependency tree to guide the joint model for slot filling and intent detection. Different from (Zhang et al., 2019), our work apply the representation over dependency tree as a inner feature instead of syntactic constraints into attention mechanism.

Graph Convolution Network (GCN) also have been utilized for many natural language processing tasks. (Vashishth et al., 2019) propose a flexible graph convolution based method for learning word embeddings. (Marcheggiani and Titov, 2017) apply a GCN as sentence encoders to produce latent feature representations of words in a sentence for semantic role labeling task. (Bastings et al., 2017) present a simple and effective approach to incorporate syntactic structure by the way of GCN into the encoder-decoder model for machine translation. (Yao et al., 2019) build a single text graph for a corpus based on word co-occurrence and document word relations, then learn a text GCN for the corpus to classify the text. Different from above work, our work propose a model that applies a GCN over dependency trees to integrate the syntactic structure for joint learning slot filling and intent detection.

6 Conclusion

In this paper, we propose a novel joint model that applies a graph convolution network over dependency trees to integrate the syntactic structure for learning slot filling and intent detection jointly. In addition, we utilize multi-head attention to fuse syntactic representation with contextual word representation to access complementary representation for SLU task. Experimental results show that our proposed model outperforms strong baseline models and achieves state-of-the-art performance on both ATIS and Snips datasets in overall accuracy metric. Finally, we apply the BERT model to our framework and experiments demonstrate that our proposed model integrating BERT model can improve the performance on both slot filling and intent detection more obviously.

Acknowledgments

The work was done when the first author was an intern at Meituan Dialogue Group. We thank Xiaojie Wang, Jiangnan Xia and Hengtong Lu for the discussion. We thank all anonymous reviewers for their constructive feedback.

References

Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Simaan. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *EMNLP*, pages 1957–1967.

Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *EMNLP*.

Kehai Chen, Rui Wang, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2018. Syntax-directed attention for neural machine translation. In *AAAI*.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

Haihong E, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *ACL*, pages 5467–5471.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *NAACL*, pages 753–757.

Dilek Hakkani-Tür, Gökhan Tür, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Interspeech*, pages 715–719.

Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language systems pilot corpus. In *Speech and Natural Language*.

Dan Hendrycks and Kevin Gimpel. 2017. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *ArXiv*, abs/1606.08415.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *AAAI*.

Changliang Li, Liang Li, and Ji Qi. 2018. A self-attentive model with gate mechanism for spoken language understanding. In *EMNLP*, pages 3824–3833.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *Interspeech 2016*, pages 685–689.

Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and WANG Houfeng. 2015. A dependency-based neural network for relation classification. In *ACL-IJCNLP*, pages 285–290.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *EMNLP*, pages 1506–1515.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *ACL*, pages 1105–1116.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. In *EMNLP-IJCNLP*, pages 2078–2087.

Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

Shikhar Vashishth, Manik Bhandari, Prateek Yadav, Piyush Rai, Chiranjib Bhattacharyya, and Partha Talukdar. 2019. Incorporating syntactic and semantic information in word embeddings using graph convolutional networks. In *ACL*, pages 3308–3318.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Yu Wang, Yilin Shen, and Hongxia Jin. 2018. A bi-model based rnn semantic frame parsing model for intent detection and slot filling. In *NAACL*, pages 309–314.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *EMNLP*, pages 1785–1794.

Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi. 2014. Spoken language understanding using long short-term memory neural networks. In *SLT*, pages 189–194. IEEE.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *AAAI*, volume 33, pages 7370–7377.

Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip S Yu. 2018a. Joint slot filling and intent detection via capsule neural networks. *arXiv preprint arXiv:1812.09471*.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018b. Graph convolution over pruned dependency trees improves relation extraction. In *EMNLP*.

Zhuosheng Zhang, Yuwei Wu, Junru Zhou, Sufeng Duan, and Hai Zhao. 2019. Sg-net: Syntax-guided machine reading comprehension. *arXiv preprint arXiv:1908.05147*.

Table Fact Verification with Structure-Aware Transformer*

Hongzhi Zhang†, Yingyao Wang◇, Sirui Wang†, Xuezhi Cao†, Fuzheng Zhang†, Zhongyuan Wang†
† Meituan Dianping Group, Beijing, China
◇Harbin Institute of Technology, China
{zhanghongzhi03,wangsirui,caoxuezhi,zhangfuzheng}@meituan.com
yywang@hit-mtlab.net, wzhy@outlook.com

Abstract

Verifying fact on semi-structured evidence like tables requires the ability to encode structural information and perform symbolic reasoning. Pre-trained language models trained on natural language could not be directly applied to encode tables, because simply linearizing tables into sequences will lose the cell alignment information. To better utilize pre-trained transformers for table representation, we propose a **S**tructure-**A**ware **T**ransformer (**SAT**), which injects the table structural information into the mask of the self-attention layer. A method to combine symbolic and linguistic reasoning is also explored for this task. Our method outperforms baseline with 4.93% on TabFact, a large scale table verification dataset.

1 Introduction

Table fact verification aims at classifying whether a textual hypothesis is entailed or refuted by the given table. It could benefit downstream tasks such as fake news detection, misinformation detection, etc. Compared to fact verification over textual evidence (Dagan et al., 2006; Bowman et al., 2015), verification on semi-structured data further requires 1) the ability to encode and understand structural information of tables, and 2) the ability to perform symbolic reasoning over structured data, such as counting, comparing, and numerical calculation. Although large-scale pre-trained language models (Devlin et al., 2019; Yang et al., 2019) achieved dominant results on textual entailment datasets (Wang et al., 2019), they could not be directly used to encode semi-structured data as they are pre-trained on unstructured natural language.

Wenhu et al. (2020) eliminate the discrepancy by serializing tables into word sequences, and then table fact verification could be processed as a natural language inference task. The most straightforward method for table serialization is linearizing the table contents via horizontal scan. However, this would destroy structural information within tables, i.e. the alignments between table cells. In Figure 1, the value "533" and "733" is meaningless digits without the column name "core clock", and it is hard for the model to recover the alignments from the flattened word sequence. Therefore, Table-BERT (Wenhu et al., 2020) includes the column name into cell representation using natural language templates during the linearization. However, comparing or counting column contents of different rows over the flattened word sequence remains a hard task, and simply duplicating the column name multiple times does not achieve satisfying results.

To better utilize the transformer architecture for table representation, we propose to inject the table's structural information into the mask of the self-attention layer. Figure 2 illustrates the pattern commonly adopted when human read or write a table. Usually, each table row describes a record, and cell $c_{1,2}$ describes a record property with the attribute name clarified in the corresponding column name $c_{0,2}$. Besides, values of the same column are usually compared or aggregated for analysis. So, the colored row and column are most crucial to the representation of cell $c_{1,2}$. In the long flattened sequence obtained by horizontal/vertical scan, the alignments between table cells would be disturbed by other unimportant words. To tackle this problem, we have the representation of cell $c_{1,2}$ only depend on the colored cells in Figure 2 by zeroing the attention weights to other ones. Figure 3 illustrates the representation of cell $c_{1,2}$ utilizing transformer. Through masking, only two pseudo sentences, i.e. the corresponding row and column, that share some common words are considered in the representation of each cell. That is, the flattened word sequence is implicitly decomposed into a series of small readable sentences so as to unleashes the power of large pre-trained language model.

*The first two authors contribute equally to this work.

Comparison of intel graphics processing units

cpu	market	core clock (mhz)	execution units	memory bandwidth
celeron g1101 pentiumg69xx	desktop	533	12	17 gb/s
core i3 - 5x0 core i5 - 655k	desktop	733	12	21.3 gb/s
core i7 - 620le core i7 - 6x0lm	mobile	266-566	12	17.1 gb/s

Entailed Statement

1. each cpu have 12 execution unit
2. core i3 – 5x0 have faster core clock than core i7 -620le

Refuted Statement

1. core i7 - 620le is designed for mobile market and its memory bandwidth is 21.3 gb/s.
2. There are three series of cpu designed for desktop market.

Figure 1: Examples of table fact verification, the right boxes provide entailed and refuted statements respectively.

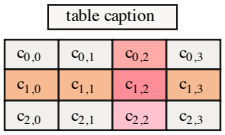


Figure 2: Illustration of table understanding. The colored row and column are crucial to understanding cell $c_{1,2}$.

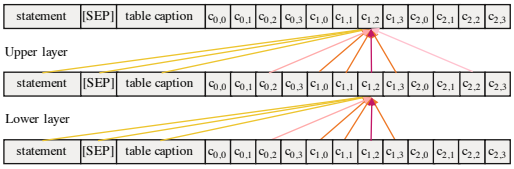


Figure 3: Illustration of masked self-attention for representation of cell $c_{1,2}$. Attentions among cells of the same column are enabled in upper layers to support cross-row reasoning, e.g. $c_{1,2} \sim c_{2,2}$.

Pre-trained transformers are good at semantic-level understanding, i.e. capturing the identical meaning between different expressions. However, one limitation is that they are not doing perfectly in symbolic reasoning (Asai and Hajishirzi, 2020). To tackle this, we perform first-order aggregation over each column and append the result as a special row into the table. An improvement of 1% is achieved, indicating that the ability of hard symbolic reasoning requires further studying.

Our contributions are summarized as follows:

- A **S**tructure-**A**ware **T**ransformer (**SAT**) is devised to better represent semi-structured tables, which injects structural information into attention mask of pre-trained transformers.
- For statements that require symbolic reasoning, we explore a method to combine symbolic reasoning and semantic matching.
- Experimental results show that our method outperforms the state-of-the-art method by 4.93%. Our code is available at https://github.com/zhhongzhi/sat.

2 Methodology

As the examples shown in Figure 1, given a statement S, table fact verification aims to classify whether the statement is entailed or refuted by the evidence table T. The table T consists of a caption t and cells $\{c_{i,j}\}$ of $m \times n$, where m and n are the numbers of rows and columns. Since pre-trained transformer could only take word sequences as input, we feed it with a concatenation of the statement S, the [SEP] token, the table caption t, and the flattened table T_f. The table could be serialized by the horizontal or vertical scan. Figure 3 shows an example of horizontal scanning.

The representation of the word sequence follows the general encoding procedure of the pre-trained transformers (Devlin et al., 2019), so we only describe the self-attention layer in which an attention mask is introduced for table representation. As illustrated in Figure 2, understanding the table requires both horizontal and vertical views. That is, if the table is flattened by a horizontal scan, the vertical alignment information will be lost, and vise versa. For example, the column name $c_{0,2}$ is crucial to the representation of $c_{1,2}$, but its signal could be perturbed by other cells in grey, since all $c_{0,*}$ and $c_{2,*}$ cells are far from $c_{1,2}$ in the flattened sequence and are processed equally.

Therefore, we propose to recover the alignment information by masking signals of unimportant cells during self-attention. The attention mask $\boldsymbol{M} \in \mathbf{R}^{L \times L}$ is defined as:

$$M_{i,j} = \begin{cases} 0 & w_i \sim w_j \\ -\infty & w_i \not\sim w_j \end{cases} \quad (1)$$

where L is the sequence length, and $w_i \sim w_j$ denotes that w_j is attended to when generating representation of w_i, while $w_j \not\sim w_i$ means the opposite. Denote the input of l-th self-attention layer as $\boldsymbol{H}^l \in \mathbf{R}^{L \times d}$, where d is the hidden size. The

attention mask is then applied to the self-attention layer as follows:

$$\boldsymbol{Q}^l, \boldsymbol{K}^l, \boldsymbol{V}^l = \boldsymbol{H}^l\boldsymbol{W}_q, \boldsymbol{H}^l\boldsymbol{W}_k, \boldsymbol{H}^l\boldsymbol{W}_v$$
$$\boldsymbol{A}^l = \text{softmax}(\frac{\boldsymbol{Q}^l\boldsymbol{K}^{l^T} + \boldsymbol{M}}{\sqrt{d_k}}) \quad (2)$$

where $\boldsymbol{W}_* \in \mathbf{R}^{d\times d_k}$ are trainable parameters. The output of self-attention layer is then calculated as:

$$\boldsymbol{H}^{l+1} = \boldsymbol{A}^l\boldsymbol{V}^l \quad (3)$$

It could be observed that if $w_j \not\sim w_i$, then $A_{i,j}$ is reset to zero and $\boldsymbol{H}_j^l$ will not contribute to the representation of w_i, i.e. $\boldsymbol{H}_i^{l+1}$.

Figure 3 sketches the representation learning of tokens in cell $c_{1,2}$ leveraging the masked self-attention. In the lower layers, the token representation of each cell considers information from four aspects: a) tokens of the same row that describe the same entry, b) its column title that clarifies the attribute name, c) the table caption which provides global background, and d) the statement for verification. In the upper layers, cross row attention among cells of the same column is further enabled. In this manner, lower layers focus on capturing low-level lexical information and upper layers are capable of simple cross-row reasoning. Note that tokens of the statement S and the table caption receive information from all cells.

Another preferred ability of SAT is to perform symbolic reasoning such as counting, comparing, and numerical calculation. Pre-trained models like BERT are good at semantic-level understanding, but not symbolic reasoning (Geva et al., 2020; Asai and Hajishirzi, 2020). We explore to enhance the performance of counting verification by converting the counting problem into a semantic matching problem. Specifically, for every column, the frequency of duplicate cell contents is counted as a summary cell, leading to a summary row which is then appended to the table. For example, the summary cell of the second column in Figure 1 is "count desktop:2", so the second refuted statement could be verified via semantic matching.

3 Experiments

3.1 Dataset

Experiments are carried out using TabFact[1] (Wenhu et al., 2020), a large scale table fact verification dataset. The basic statistics of TabFact are listed in Table 1. The dataset contains both simple and complex statements. Simple statements only involve a single row/record, while the complex ones require higher-order semantics (argmax, count, etc.), and the statements are rephrased so more ability on linguistic reasoning is required.

Split	#Statement	#Table	Simple/Complex
Train	92,238	13,182	–
Val	12,792	1,696	–
Test	12,779	1,695	4,230/8,609

Table 1: Basic statistics of TabFact.

3.2 Experimental Settings

Model weights are initialized using BERT-base model trained on English corpus. The first 6 layers are regarded as lower layers, and the other 6 layers are taken as upper layers. We finetune the model with a batch size of 10 and a learning rate of 2e-5. It usually takes 15-18 epochs until convergence.

The flatten sequence is usually longer than the sequence limit of BERT, which requires more memory and training time. Hence, we only retain the top 5 table rows according to the number of words shared with the statement. During experiments, the maximum sequence length is set to 256.

3.3 Results and Ablation Study

The experimental results on TabFact are listed in Table 2. Our method achieves an accuracy of 73.23% on the test set and outperforms Table-BERT by 4.93%. The improvement on complex statements is even larger, which achieves 5.75%.

Effect of Attention Mask Without the attention mask, test accuracy is 67.67% and 64.27% for horizontal and vertical scans respectively, namely a decrease of 5.15% and 8.96% compared to the complete SAT. An interesting finding is that the horizontal scan outperforms the vertical scan when removing the mask, which is consistent with our intuition that each row describes an entry and thus horizontal alignment information is more important. With the cell alignment information recovered by the attention mask, the gap is rather small when using SAT, demonstrating its robustness towards different scan directions.

The last two rows of Table 2 present two variants of the masks, where we adopt an identical mask matrix for all transformer layers instead of using different ones for low/high layers. Results indicate

[1] https://github.com/wenhuchen/Table-Fact-Checking

Model	Val	Test	Test(simple)	Test(complex)
LPA(Wenhu et al., 2020)†	65.1	65.3	78.7	58.5
Table-BERT(Wenhu et al., 2020)†	66.1	65.1	79.1	58.2
Table-BERT tuned*	68.38	68.30	82.35	61.48
BERT with cell position encoding	59.31	59.44	63.24	57.58
SAT with Horizontal scan	72.96	72.82	85.44	66.62
- w/o visible matrix	68.41	67.67	75.93	63.61
- w/o summary row	72.00	72.09	85.53	65.49
- w/o visible matrix w/o summary row	66.84	66.01	74.37	61.90
SAT with Vertical scan	**73.31**	**73.23**	**85.46**	**67.23**
- w/o visible matrix	64.21	64.27	68.77	62.06
- w/o summary row	71.71	71.59	84.70	65.15
- w/o summary row and w/o visible matrix	63.03	62.34	66.71	60.19
- all layers w/o cross row attention	72.83	72.26	84.61	66.11
- all layers w cross row attention	72.02	71.82	83.45	66.10

Table 2: The accuracy (%) of different models. The results annotated with † are cited from literature, and *Table-BERT tuned** denotes results obtained by changing the leaning rate from 5e-5 to 1e-5.

that designing different mask matrix for low/high layers, with the intention to model low-level lexical information and high-level cross-row reasoning, has indeed achieved better performance.

Essentially, by masking signals of unimportant cells, SAT implicitly segments the unnatural long sequence into a series of meaningful subsequences. Such sub-sequences are more friendly to pre-trained language models, so the power of large pre-trained transformer can be unleashed.

The Summary Row Appending a summary row to the table brings a stable improvement of 1%, which mainly contributes to the complex test set. This indicates that although pre-trained transformer is dominant on semantic understanding, its ability on symbolic reasoning is limited. With the counting problem in scope, experimental results show that it is promising to combine both symbolic reasoning and semantic understanding abilities by feeding symbolic reasoning results into SAT.

SAT vs Table Position Embeddings Experiments are further carried out to identify whether the table position encoding method introduced in TaPaS(Herzig et al., 2020) is better than the proposed SAT on table encoding. Row and column positional embeddings are added to the original positional embeddings of BERT to identify the table alignment information. The experimental results are listed in the fourth row of Table 2. An accuracy of 59.8% is observed while the accuracy of the BERT baseline is 68.30%. The results show that BERT is perturbed by the additional table positional embeddings and the model did not converge well. Though the table position information is appended to the inputs, the following transformer layers are not ready to accept and propagate the signal without pre-training. It is demonstrated that simply providing positional information without pre-training is not sufficient for Transformer to encode tables.

3.4 Case study

We analyzed samples that are fixed by SAT compared to baselines. It is observed that a large portion (43/80) of them are statements involve multiple facts/table cells that do not requires logic reasoning. Besides, several problems (9/80) that requires simple count and comparison are fixed. The model both fixed (the other 38) and failed on some samples that require complex symbolic logical reasoning, such as argument sort, conditional aggregation and then comparison. The behavior is most likely random guess for both SAT and baselines. The results show that SAT mainly contributes to the general table representation and enhance the linguistic reasoning, and the summary row appended helps to solve some count problems.

4 Related Work

To encourage the study on table fact verification, Wenhu et al. (2020) construct a large scale table fact checking dataset and study two promising approaches, Table-BERT and Latent Program Algorithm (LPA) respectively. Table-BERT transforms the problem into a natural language inference task to leverage the power of the pre-trained language models. LPA formulates the task as a program

synthesis problem and it is good at symbolic reasoning. Our work aligns with the direction of Table-BERT. Inspired by existing work Weijie et al. (2020); Nguyen et al. (2020); Dong et al. (2019); Yang et al. (2019) that manipulates self-attention masks, we devise a structure-aware transformer to attain better table representation.

There are several recent works that table fact verification could benefit from. Geva et al. (2020) and Asai and Hajishirzi (2020) study to improve the pre-trained model in numerical reasoning and logical comparisons. The enhanced pre-trained model could be directly used in our approach. Herzig et al. (2020) extend BERT's architecture to encode tables for the table question answering task (Iyyer et al., 2017), where additional embeddings identifying the row and column number are added. The proposed architecture is potentially applicable to table fact checking but requires expensive pre-training.

5 Conclusion

We propose SAT to enhance the pre-trained transformer's ability on table representation by injecting structural information into the mask of self-attention layers. Significant improvements on TabFact demonstrate its effectiveness. We further enhance SAT by appending a summary row to the table, the results show that it is promising to solve the fact verification that requires both symbolic reasoning and semantic understanding by feeding symbolic reasoning results into SAT. Overall, an improvement of 4.93% is achieved compared to the state-of-the-art method. The proposed method can further contribute to other semi-structured data (table, graph, etc.) related tasks, e.g. WikiTableQuestions (Pasupat and Liang, 2015) and CommonsenseQA (Talmor et al., 2019). There still exists plenty of potentials that require future studies in this direction.

References

Akari Asai and Hannaneh Hajishirzi. 2020. Logic-guided data augmentation and regularization for consistent question answering.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 13063–13075. Curran Associates, Inc.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models.

Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. 2020. Tapas: Weakly supervised table parsing via pre-training. *arXiv preprint arXiv:2004.02349*.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831, Vancouver, Canada. Association for Computational Linguistics.

Xuan-Phi Nguyen, Shafiq Joty, Steven Hoi, and Richard Socher. 2020. Tree-structured attention with hierarchical accumulation. In *International Conference on Learning Representations*.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 3266–3280. Curran Associates, Inc.

Liu Weijie, Zhou Peng, and Qi Ju Haotang Deng Ping Wang Zhe Zhao, Zhiruo Wang. 2020. K-BERT: Enabling language representation with knowledge graph. In *Proceedings of AAAI 2020*.

Chen Wenhu, Wang Hongmin, Hong Wang Shiyang Li Xiyou Zhou Jianshu Chen, Yunkai Zhang, and William Yang Wang. 2020. Tabfact : A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc.

An Effective Approach for Citation Intent Recognition Based on Bert and LightGBM

Weilong Chen*
University of Electronic Science and Technology of China
chenweilong921@gmail.com

Shuaipeng Liu*
Meituan-Dianping Group
liushuaipeng@meituan.com

Wei Bao*
Southeast University
willinseu@gmail.com

Huixing Jiang†
Meituan-Dianping Group
jianghuixing@meituan.com

ABSTRACT

In the development of science and technology, the public scientific theses have played an important role and greatly promoted the development of society. The vast majority scientific progress was announced in the form of papers in past centuries, and impactful contributions were often recognized by the research community with a great number of citations. However, inappropriate citation of papers still occurs from time to time and hinders the progress of human civilization. In this paper, we proposed an effective framework to address the citation intent recognition challenge in ACM WSDM Cup 2020[1]. Our team name is *ferryman* and in our solution, we regarded this problem as the Information Retrieve (IR) task and proposed a framework with two stages of recall and ranking and finally our team won *the 1st place* with a Mean Average Precision @ 3 (MAP@3) score of 0.42583 on the final leaderboard[2].

KEYWORDS

Citation Intent Recognition, Information Retrieve, Nature Language Processing

1 INTRODUCTION

WSDM Cup is a competition-style event co-located with the leading WSDM conference. This paper describes our solution for Citation Intent Recognition, one of WSDM Cup 2020 tasks, and we won the 1st place in the final leaderboard. Science has emerged as a dominant engine of innovation for modern society. Moreover, its rich published traces allow us to understand, predict and guide its advance and utility like never before. Research papers are the dominant media for state-of-art knowledge. Therefore, if we can develop models that understand research papers, we can greatly enhance the ability of computers to understand knowledge.

The competition provided a large paper dataset, which contains roughly 800K papers, along with paragraphs or sentences which describe the research papers. These pieces of description are mainly from paper text which introduces citations. The participants are required to recognize the paper cited in the describe texts. This competition uses Mean Average Precision @3 (MAP@3) as the evaluation metric which is described by the following function:

$$MAP@3 = \frac{1}{|U|}\sum_{u=1}^{|U|}\sum_{k=1}^{min(3,n))} P(k)) \quad (1)$$

Where |U| is the number of press_id in the test set, P(k) is the precision at cutoff k, n is the number of predicted papers.

After analyzing the challenge, we regard it as an Information Retrieve (IR) task[11], The IR focuses on the problem of finding the most matched Top N documents with a query from a massive number of candidate documents. In this challenge, the description text is the query and the candidate papers are the documents to be retrieved. To handle this challenge, we made a plan with two stages including recall and ranking. In recall stage, several unsupervised methods like Axiomatic F1EXP[5], DFI Similarity[7], Okapi BM25[14] are built to reduce the scope of candidates, then we draw learning to rank models such as BERT[4][10] and lightGBM[6] to ranking the candidate papers which is selected in the recalling stage.

The rest of the paper is organized as follows: Section 2 describes our solution which contains the model details. In Section 3, we show the experiments and results of our model. Finally, we conclude our analysis of the challenge, as well as some additional discussions of the future directions in Section 4.

2 METHODOLOGY

In this section, we introduce our framework for Citation Intent Recognition. Firstly, we introduce the recall strategy. Secondly, we introduce the rank strategy based BERT and lightGBM, Finally We introduce how to integrate the models.An overall framework and processing pipeline of our solution is showed in Figure 1. Our trained models and source code are publicly available on GitHub[3].

2.1 Recall Strategy

In the recall stage, candidate papers and descriptions were represented as a vector using vector space model and bag-of-N-gram model, in practice, the max N is set to 2 owing to the huge computational space. Then we use several similarity measurement to reduce the retrieve scope, including TFIDF, BM25, LM Dirichlet, Axiomatic F3EXP, DFI Similarity, Axiomatic F1EXP, Axiomatic F2EXP, Axiomatic F1LOG, Axiomatic F2LOG, Axiomatic F3LOG, Boolean Similarity, LM Jelinek Mercer Similarity, DFR Similarity, IB Similarity and so on[2][11]. And we apply the structure introduced above

*Both authors contributed equally to this research.
†All the corresponding to Huixing Jiang.
[1]http://www.wsdm-conference.org/2020/wsdm-cup-2020.php
[2]https://biendata.com/competition/wsdm2020/final-leaderboard/
[3]https://github.com/myeclipse/wsdm_cup_2020_solution

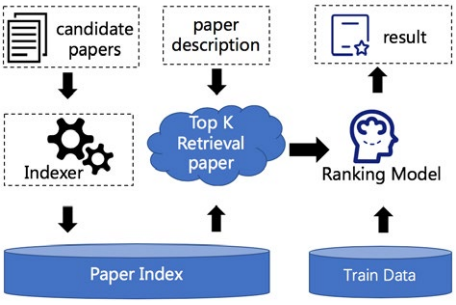


Figure 1: An overall framework and pipeline of our solution for citation intent recognition

on different scales of a paper, such as title, abstract, keywords and full text. In our practice, the F1EXP has the highest recall score and BM25 get the highest MAP score. The recall results is not only used to reduce the retrieve scope but all as a part of features used in the LGB ranking stage.

2.2 BERT Model

The BERT[4][10] model architecture is based on a multilayer bidirectional Transformer[15] As Fig. 2. Instead of the traditional left-to-right language modeling objective, BERT is trained on two tasks: predicting randomly masked tokens and predicting whether two sentences follow each other. BERT model gets a lot of state of the arts in many tasks, and we also use the BERT model in our strategy. There are two types of BERT models following the same architecture as BERT but instead pre-trained on the different scientific texts: SciBERT[1] and BioBERT[9]. Also, we trained the pre-trained model in two ways: The Point-Wise model and the Pair-Wise model.

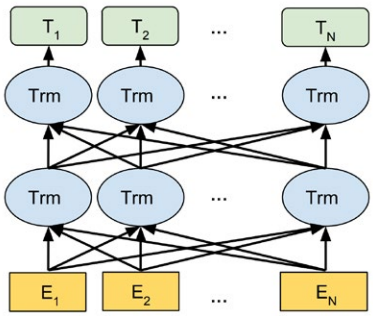


Figure 2: Bidirectional transformer architectures of BERT

2.2.1 Data Preprocessing. The better preprocessing of the input can get better performance. Firstly, we removed the excess white-space and some stop words, and we did some word segmentation and did part-of-speech tagging. Secondly, we normalized the word form for the different tags of the sentence and lowercased all letters. We compared the input without preprocessing and the input with preprocessing, finding that the input with preprocessing is better than another one.

2.2.2 Bert with Point-Wise. We trained the BERT with Point-Wise way which means we defined the task as the binary classification. We preprocessed the two sentences (the description sentence and the paper-described sentence). We joined them in one sentence with [SEP] token and put them into the BERT model. We trained the token of the sentence with binary cross-entropy loss to dig the difference between description sentence and paper-described sentence As Figure 3. The probability can measure how well the two sentences match. However, too much negative samples can destroy the performance of the BERT model and the Point-Wise way didn't take into account the internal dependencies between the documents corresponding to the same query. On the one hand, the samples in the input space are not independently identically distribution. On the other hand, the structure between these samples was not fully utilized. When different queries correspond to different numbers of documents, the overall loss will be dominated by the query group with a large number of documents. Each group of queries should be equivalent. We need to have another way to get better performance of the model. We tried the Pair-Wise model.

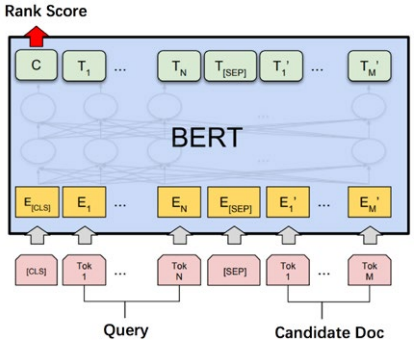


Figure 3: Ranking with BERT

2.2.3 Bert with Pair-Wise. Learning2Rank applies machine learning technology to the ranking problem and trains the ranking model. Usually, the discriminant supervised machine learning algorithm is applied. Learning2Rank task seeks ranking results and does not require precise scoring, as long as there is a relative scoring. Learning2Rank framework has the following characteristics:

- The samples in the input space are two feature vectors (corresponding to the same query) composed of two documents (and corresponding query).
- The samples in the output space are pairwise preference.
- The samples in the space are two-variable functions and the loss function evaluates the difference between the predicted preference and the true preference of the document pair.

We did the same preprocessing to the input sentence as the way described in the above. We used the margin ranking loss as our loss function and trained several triplet samples with the same description text and different paper-described sentences. It not only helped to get a better ranking of similarity but also compared the differences between each description text. We got a higher score than the BERT model with Point-Wise.

2.3 Lightgbm Model

In order to increase the diversity of the model, in addition to Bert, we choose LightGBM for modeling, and for simplicity, it is called lgb here. lgb model is a gradient boosting framework that uses tree based learning algorithms. LightGBM builds the tree in a leaf-wise way, as shown in Figure 4, which makes the model converge faster.LightGBM is not sensitive to outliers and can achieve high accuracy, which is widely used in industry. And in this work, compared with Bert, the effect of LightGBM is better, the LightGBM single-model can reach 0.413 in the leaderboard. Total number of features is 1684, this contains of semantic features, statistical features and so on, which will be explained later.

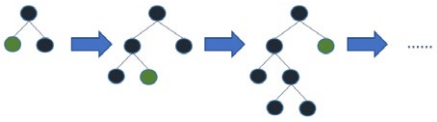


Figure 4: LightGBM's leaf growth strategy

In this work, the training method of LightGBM is lambdarank(pairwise strategy), which is about 0.5% higher than the traditional binary classification model(pointwise strategy). The following will be carried out from two aspects of feature engineering and model construction.

2.3.1 feature engineering. Our feature engineering mainly consists of the following 3 aspects:

- *Semantic feature.* Semantic features include various pre-trained word vector models such as fasttext[3], glove[13], word2vec[12], doc2vec[8] etc. And we retrain them to calculate the similarity between description and abstract. Specifically, we represent the vector of a sentence as the average of the word vectors of each word in it. Then we use the cosine distance formula and the Manhattan distance formula to measure the correlation between the two sentences, and the correlation value is used as our semantic feature.
- *Statistical features and word frequency features.* In this section, we use various word frequency-based methods to capture similarities, such as bm25, tfidf, f1exp and various length and proportion features. Among these word frequency features, we find that the similarity obtained through the bm25 method is very important. At the same time, compared with the semantic features, the word frequency features bring greater benefits to the model as a whole. We believe this is due to the large number of specialized terms in the corpus.
- *Rank features.* In order to make our model easier to "know" the essential purpose of ranking, we sort the various similarity values according to description_id (or paper_id), and divide the ranking value by the number of description_id (or paper_id) to get the relative ranking ratio. This part of can bring a 3% boosting. In detail, suppose we have m correlation features. Then through our grouping and sorting operation, since we can group according to description_id or paper_id, we can get another 2m new sorting features, and divide by the corresponding number in the group, we can also get another 2m new sorting scale feature.

2.3.2 Modeling Methodology. Since the same description can recall multiple paper abstracts, from the perspective of a classification problem, this is an imbalance of positive and negative samples, so the number of samples cannot be too large. However, in the composition of the training set, we found that the positive sample coverage ratio of the recall samples is also very important, so we chose a higher number of recall samples. At the same time, in the training set, because some descriptions cannot recall the positive samples through our recall strategy, we artificially added the positive samples to the training set in order to ensure the coverage of the positive samples. Through the above data preprocessing steps, the amount of training data for lgb model is about 5 million.

Learning to Rank is one of the most commonly used algorithms to implement ranking through machine learning. It mainly includes three types of single document method (pointwise), document pair method (pairwise) and document list (listwise). The pointwise single-document method means it will judge the relevance of each document to this query, and converting the documents ranking problem into a classification (such as related, irrelevant) or a regression problem. However, the pointwise method does not learn other document as features when modeling, so it cannot consider the order relationship between different documents. The purpose of rank learning is mainly to sort the documents in the search results according to the magnitude of relevance, so pointwise is bound to have some defects.

Aiming at the problem of pointwise, the pairwise document method does not care about the specific value of the correlation between a document and a query, but converts the ranking problem into any two different documents related to the relative order of the current query. In order to be relevant and irrelevant, the two categories are recorded as +1, 0, and then transformed into classification problems. Listwise treats all related documents corresponding to a query as a single training sample.

In total, Our Lgb model is trained using a 5-fold cross-validation method. The training target is lambdarank, and the offline verification indicators are MAP @ 3 and MAP @ 5. The model score can reach 0.413.

2.4 Ensemble Methodology

In the model ensemble stage, we adopted a simple and efficient way and get 1.2% boosting. We group the model prediction results of LightGBM and BERT by description id, and then add the the ranking values with weighting operation, the weights of which are 6 and 4, respectively. The details are shown in Figure 5.

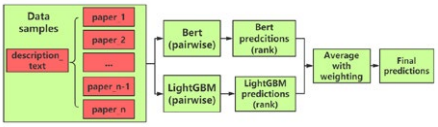


Figure 5: Ensemble strategy based on rank blending with weighting operation

3 EXPERIMENT

3.1 Experimental Settings

In this experiment, our training set has a total of about 63,000 paper description documents, and its number on the test set is about 34,000. At the same time, our candidate paper dataset has a total of about 840,000 papers.For each piece of description, we need to choose 3 best-matching papers in candidate paper dataset.

Table 1: Online map@3 score with different models

Model	Online MAP@3 LB score
Bert(pointwise)	0.397
Bert(pairwise)	0.402
LightGBM(pointwise)	0.405
LightGBM(pairwise)	0.413
Ensemble	0.425

3.2 Model Comparison

Here we compare the performance of our method with different settings. The results are shown in Table 1.From the table, we can see that no matter in Bert or LightGBM, the result of pairwise training method is better than pointwise. At the same time, the LightGBM model based on detailed feature engineering is very effective. Our best single mode is LightGBM trained using pairwise methods, which is reflected in the algorithm settings as lambdarank.

At the same time, our highest score is the ensemble model of the Bert model and LightGBM model. We noticed that the improvement based on ensemble between LightGBM models is very limited, but the Bert model and LightGBM model can bring a huge improvement of 1.2%, which we believe is due to the huge difference between the two models.

4 CONCLUSION

In this paper, we propose a method based on Bert and LightGBM for recognition of paper citations, in which both Bert and LightGBM are trained using pairwise methods. At the same time, we won the first place in the Citation Intent Recognition competition (WSDM Cup 2020 track1).

ACKNOWLEDGEMENTS

We thank everyone associated with organizing and sponsoring the WSDM Cup 2020. Dataset was provided by Microsoft Research. Challenge was sponsored and managed by the 13th ACM International Conference on Web Search and Data Mining (WSDM 2020). Competition platform was hosted by Biendata. we are very grateful to WSDM Cup Chairs Kyumin Lee and Neil Shah for their great efforts during the challenge.

REFERENCES

[1] Iz Beltagy, Arman Cohan, and Kyle Lo. 2019. Scibert: Pretrained contextualized embeddings for scientific text. *arXiv preprint arXiv:1903.10676* (2019).
[2] Andrzej Białecki, Robert Muir, Grant Ingersoll, and Lucid Imagination. 2012. Apache lucene 4. In *SIGIR 2012 workshop on open source information retrieval*. 17.
[3] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146.
[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[5] Hui Fang and ChengXiang Zhai. 2005. An exploration of axiomatic approaches to information retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. 480–487.
[6] Guolin Ke, Qi Meng, Thomas William Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tieyan Liu. 2017. LightGBM: a highly efficient gradient boosting decision tree. (2017), 3149–3157.
[7] İlker Kocabaş, Bekir Taner Dinçer, and Bahar Karaoğlan. 2014. A nonparametric term weighting method for information retrieval based on measuring the divergence from independence. *Information retrieval* 17, 2 (2014), 153–176.
[8] Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. arXiv:cs.CL/1405.4053
[9] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv:1901.08746* (2019).
[10] Shuaipeng Liu, Shuo Liu, and Lei Ren. 2019. Trust or Suspect? An Empirical Ensemble Framework for Fake News Classification. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining, Melbourne, Australia*. 11–15.
[11] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge university press.
[12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. arXiv:cs.CL/1301.3781
[13] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1532–1543. https://doi.org/10.3115/v1/D14-1162
[14] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *Nist Special Publication Sp* 109 (1995), 109.
[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.


CODE A BETTER LIFE
一行代码 亿万生活
美团
技术团队
长按二维码关注我们