# DSCF: Dual-Source Counterfactual Fusion for High-Dimensional Combinatorial Interventions

**Anonymous submission**


## Abstract

Estimating counterfactual outcomes from observational data is critical for informed decision-making in domains such as personalized marketing, healthcare, and online platforms. In these contexts, decision processes frequently involve high-dimensional combinatorial interventions, including bundled channel allocation or product set recommendations. For such scenarios, both causal assessment of historical strategies and optimization of novel interventions necessitate models capable of extrapolating to intervention combinations that are underrepresented or entirely absent in observational data. Specifically, in digital marketing, companies often need to evaluate new combinations of channels or target emerging user segments that have not been previously exposed. Furthermore, inherent biases in observational datasets, stemming from prior allocation policies and targeting mechanisms, further aggravate coverage sparsity and compromise off-support counterfactual inference. In this work, we propose Dual-Source Counterfactual Fusion (DSCF), a scalable framework that enables accurate counterfactual prediction under high-dimensional combinatorial interventions, with improved robustness to confounding bias. DSCF jointly models observational data and proxy counterfactual samples through a dual-head mixture-of-experts architecture and domain-guided fusion. This design effectively balances bias reduction and information diversity while enabling adaptive generalization to counterfactual inputs. Extensive experiments on both synthetic and semi-synthetic datasets demonstrate the effectiveness and robustness of DSCF across diverse scenarios.


## Introduction

Understanding the joint effects of multiple interdependent interventions is increasingly critical to decision-making in domains such as online platforms and digital marketing (Yao et al. 2022). Figure 1 illustrates a typical digital marketing scenario: advertisers determine exposure strategies for each user based on individual characteristics (e.g., demographics and behavioral signals), which in turn influence the set of marketing channels a user is exposed to. These combinations, together with user intent, influence downstream business outcomes such as conversion rate and long-term retention. To enable fine-grained business analysis and future strategic optimization, a core counterfactual question is: how would these outcomes change if we assigned the user a different combination of channels? Modeling each channel in isolation fails to capture the synergistic or antagonistic effects that arise from their co-occurrence, necessitating a shift from single-intervention analysis to combinatorial counterfactual modeling.

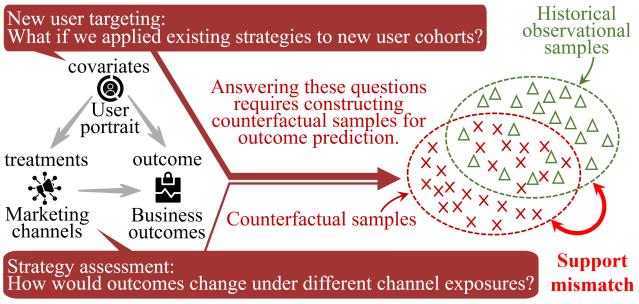



Figure 1: Illustration of key counterfactual questions and support mismatch in high-dimensional intervention settings.

However, historical observational data only documents outcomes for a limited and biased subset of intervention combinations, specifically those deployed under prior allocation policies that systematically prioritize safe, high-performing strategies aligned with historically targeted user segments. This policy-driven selection mechanism results in systematic selection bias and coverage sparsity. Furthermore, the inherent low-rank structure of user populations exacerbates this challenge: demographically or behaviorally similar users tend to receive homogenized treatments, leaving extensive regions of the combinatorial intervention space underexplored. Under such conditions, direct empirical learning from observational data is insufficient to support forward-looking strategy development (e.g., targeting novel user cohorts, reconfiguring campaign bundles, or simulating budget reallocations). Without the capacity for out-of-support generalization, such models are inherently incapable of providing actionable insights for future decision-making.

This need for off-support counterfactual generalization under high-dimensional, biased conditions presents a critical yet underexplored challenge, particularly in internet applications such as high-value action discovery (M-Squared 2025), multi-touch attribution (Ren et al. 2018; Arava et al. 2018; Yao et al. 2022), and ad optimization (Shi et al. 2024). Existing methods (Wang et al. 2024) are effective only for low-dimensional combinatorial interventions and fall short

in real-world applications, either due to poor off-support generalization or restrictive assumptions.

To mitigate these limitations, we propose **Dual-Source Counterfactual Fusion (DSCF)**, a scalable framework for accurate and robust counterfactual prediction under support-sparse, high-dimensional combinatorial interventions. DSCF seeks to combine the low-bias nature of proxy counterfactual samples (obtained via matching) with the greater informational richness of observational data (compared to proxy data). To this end, it jointly learns from both domains through a dual-head Multi-gate Mixture-of-Experts (MMoE) (Ma et al. 2018) architecture, incorporating an independently trained domain classifier to enable dynamic, input-dependent fusion for improved counterfactual adaptation. Throughout the pipeline, DSCF imposes minimal assumptions on data distribution and modality, making it well-suited for industrial-scale applications in real-world settings.

We evaluate DSCF on both synthetic and semi-synthetic benchmarks. On synthetic data, it consistently achieves substantial improvements across diverse experimental configurations. On semi-synthetic datasets constructed from real-world user logs, it reduces RMSE and MAE by 32.1% and 48.3%, respectively, compared to the state-of-the-art methods.

## Related Work

**Classical ITE extensions.** Traditional ITE methods, such as sample reweighting (Arbour, Dimmery, and Sondhi 2021; Chesnaye et al. 2022), matching (Stuart 2010; Schwab, Linhardt, and Karlen 2018; Wu et al. 2023), and representation learning (Shalit, Johansson, and Sontag 2017; Shi, Blei, and Veitch 2019), aim to adjust for confounding by aligning covariate distributions across treatment groups. Some efforts extend these methods to combinatorial settings by treating covariates and interventions as a joint feature space and applying ITE-style adjustment to the joint observational distribution. However, the exponentially large treatment space leads to severe support sparsity and renders direct adjustment ineffective. Reweighting-based methods (Zou et al. 2020) merely rescale samples within the observed support and cannot extrapolate beyond it (Cortes, Mansour, and Mohri 2010), while matching tends to over sample a small subset of observational samples, reducing diversity and increasing variance. Representation learning methods (Tanimoto et al. 2021) often assume invariant treatment effects across domains, which is an unrealistic premise in tasks like CTR or LTV prediction where effects are highly context-dependent. Moreover, under sparse observational coverage, the learned representations may become non-invertible (Johansson, Sontag, and Ranganath 2019), resulting in irreversible information loss and degraded estimation quality.

**Advanced modeling paradigms.** Recent methods tailored to combinatorial interventions include counterfactual data augmentation (Qian, Curth, and van der Schaar 2021), low-rank modeling (Agarwal, Agarwal, and Vijaykumar 2023), and meta-learning (Chauhan et al. 2025). While effective in constrained settings, these approaches often rely on strong structural assumptions or incur significant computational overhead, limiting their scalability to high-dimensional, support-sparse regimes. Data augmentation methods require expanding the training set by a factor of the intervention dimension and fitting a separate predictor for each intervention, incurring substantial computational costs and storage overhead. Low-rank models capture only coarse structures and fail to represent high-order interactions prevalent in complex systems, limiting their expressiveness. While meta-learning approaches offer adaptability, their reliance on nested optimization and heavily parameterized architectures hinders scalability and applicability in real-world high-dimensional settings.

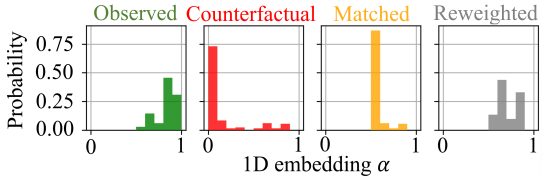



Figure 2: Empirical distributions of observed, counterfactual, matched, and reweighted sample features over a latent one-dimensional space.

## Problem Statement

In combinatorial counterfactual prediction, the goal is to predict outcomes under different combinations of interventions and contexts, based on observational data. The observational data is denoted as $D_{obs} = \{(\mathbf{x}_i, \mathbf{t}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^d$ represents covariates (e.g., user demographics), $\mathbf{t}_i \in \{0,1\}^p$ represents treatment assignments, and $y_i \in \mathbb{R}$ is the observed outcome (e.g., conversion rate). Each element of $\mathbf{t}_i$, referred to as a *cause*, indicates the presence or absence of a specific intervention (such as whether a particular marketing channel was accessed). The full vector $\mathbf{t}_i$, consisting of all causes, represents the *treatment*, the joint assignment of all binary interventions applied to a given unit. We aim to learn a hypothesis $f_\theta : \mathbb{X} \times \mathbb{T} \to \mathbb{R}$ which predicts the outcome $y$ based on both covariate $\mathbf{x}$ and treatment $\mathbf{t}$. We use binary treatments for clarity, although the method could be applied to more general intervention types, including dense or categorical inputs.

Although the exact form of the counterfactual distribution may vary across applications, eliminating confounding between covariates and causes remains a universal objective. As a practical approximation, we adopt a factorized form that assumes independence between covariates and causes. Specifically, we aim to minimize the expected loss under a factorized counterfactual distribution $P(\mathbf{X})\prod_{i=1}^p P(T^i)$: $\mathbb{E}_{P(\mathbf{X})\prod_{i=1}^p P(T^i)}\left[\mathcal{L}(f_\theta(\mathbf{X},\mathbf{T}), y(\mathbf{X},\mathbf{T}))\right]$, where $\mathcal{L}(\cdot,\cdot)$ is the error function and $y(\cdot,\cdot)$ denotes the true outcome (Zou et al. 2020). Given the combinatorial explosion of the intervention space, the inherent bias in observational data, and practical requirements for generalizing to off-support counterfactual scenarios, we do not assume the positivity condition (Rosenbaum and Rubin 1983; Pearl 2010), which requires every treatment to have a non-zero probability of being observed. Instead, we characterize distribution shift using sample-level distances.

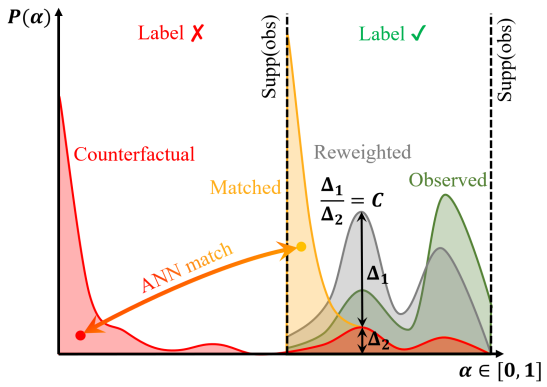



Figure 3: Schematic illustration of how reweighting and matching respond to support mismatch over the same space.

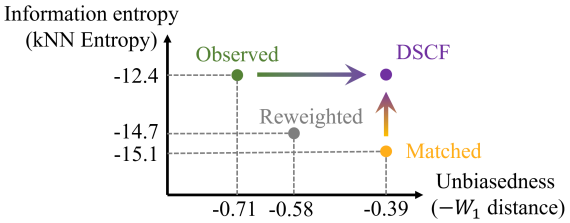



Figure 4: Illustration of the trade-off between unbiasedness and information richness across training distributions.

## Method

In this section, we first illustrate the distributional heterogeneity among observational, reweighted, and matched data through a motivating example. This insight motivates the design of DSCF, which jointly learns from observational and matched data to harness their complementary strengths. We then introduce its core components and deployment details.

### Limitations under Support Constraints

While matching and reweighting are widely used for counterfactual prediction under combinatorial interventions, they exhibit notable limitations when the positivity assumption fails. We begin with an illustrative example in Figure 2. Here, high-dimensional feature vectors $(\mathbf{x}, \mathbf{t})$ from real-world user logs are projected onto a one-dimensional latent axis $\alpha \in [0, 1]$, partitioned into 10 equal-width bins. The y-axis reflects the sample proportion in each bin. Observational samples (drawn from the logged dataset directly) and counterfactual samples (constructed by independently permuting the treatment columns of the observational data) exhibit clear support mismatch: the former are concentrated in $[0.5, 1.0]$, while the latter are primarily located in $[0.0, 0.5]$.

We next apply reweighting and matching to adjust the observational training distribution toward the target counterfactual distribution. As shown in Figure 3, reweighting rescales in-support observational samples based on their counterfactual likelihood but cannot, by design, extrapolate to out-of-support regions. Matching, by contrast, more closely approximates the counterfactual distribution (see Appendix A.5), but often over-concentrates in narrow regions, such as collapsing all probability mass from $[0.0, 0.5]$ onto the single point $\alpha = 0.5$.

To further assess these trade-offs, we evaluate observational, matched, and reweighted training distributions along two axes: **unbiasedness** (measured by the negative Wasserstein-1 distance to the counterfactual distribution) and **information richness** (measured by kNN entropy). As shown in Figure 4, matching achieves the lowest bias but suffers from substantial entropy loss due to oversampling a narrow subset of observational units, which is a consequence of positivity violations. In contrast, observational data, while biased, retains higher entropy, indicating greater information diversity. Given that the potential outcome mapping $(\mathbf{x}, \mathbf{t}) \mapsto y$ is assumed invariant across domains, our proposed method, DSCF, seeks to balance this trade-off by jointly learning from both sources. It combines the bias reduction offered by matching with the informational diversity of observational data to enable robust and accurate counterfactual prediction.

### Dual-Source Counterfactual Fusion

DSCF implements this dual-source strategy via a dual-head Multi-gate Mixture-of-Experts (MMoE) (Ma et al. 2018) architecture, with a domain classifier adaptively fusing the two heads at inference time. An overview is shown in Figure5.

**Proxy counterfactual dataset construction.** To obtain a training subset that is minimally biased with respect to the target counterfactual distribution, we construct a proxy counterfactual dataset via matching, aiming to approximate the joint distribution over $\mathbb{X} \times \mathbb{T}$ as closely as possible. We first independently shuffle each column of the observational treatment matrix and concatenate the result with the original covariates, yielding feature vectors $\{(\bar{\mathbf{x}}_i, \bar{\mathbf{t}}_i)\}_{i=1}^n \sim P(\mathbf{X}) \prod_{i=1}^p P(T^i)$. We then perform approximate nearest neighbor (ANN) search, using the synthetic inputs $(\bar{\mathbf{x}}, \bar{\mathbf{t}})$ to query the observational dataset $D_{\text{obs}}$ in the joint feature space $\mathbb{X} \times \mathbb{T}$. For each query, we include the entire matched observational sample $(\mathbf{x}', \mathbf{t}', y')$ in the proxy dataset $D_{\text{pcf}} = \{(\tilde{\mathbf{x}}_i, \tilde{\mathbf{t}}_i, \tilde{y}_i)\}_{i=1}^n$, so that the mapping $(\mathbf{x}, \mathbf{t}) \mapsto y$ remains faithful to the original data-generating process. We employ industrial-grade ANN engines such as FAISS (Johnson, Douze, and Jégou 2019) for scalable retrieval.

**Dual-head joint learning with MMoE.** We adopt MMoE as our main prediction module, consisting of a shared set of $K$ expert networks $\{E_k(\cdot)\}_{k=1}^K$ and two task-specific gating networks, $G^{\text{obs}}$ and $G^{\text{pcf}}$. The former facilitates cross-domain knowledge sharing and supervision, while the latter assigns each input to a soft combination of experts to enable adaptive specialization. Let $\mathbf{z}_{\text{obs}} = [\mathbf{x}; \mathbf{t}]$ and $\mathbf{z}_{\text{pcf}} = [\tilde{\mathbf{x}}; \tilde{\mathbf{t}}]$ denote the inputs for observational and proxy data, respectively. Each gating network produces softmax-normalized expert weights:

$$\mathbf{g}^{\text{obs}} = \sigma(G^{\text{obs}}(\mathbf{z}_{\text{obs}})), \quad \mathbf{g}^{\text{pcf}} = \sigma(G^{\text{pcf}}(\mathbf{z}_{\text{pcf}})),$$

where $\sigma(\cdot)$ denotes the softmax activation. These weights generate task-specific representations as weighted combinations of expert outputs:

$$\mathbf{h}^{\text{obs}} = \sum_{k=1}^K g_k^{\text{obs}} \cdot E_k(\mathbf{z}_{\text{obs}}), \quad \mathbf{h}^{\text{pcf}} = \sum_{k=1}^K g_k^{\text{pcf}} \cdot E_k(\mathbf{z}_{\text{pcf}}).$$

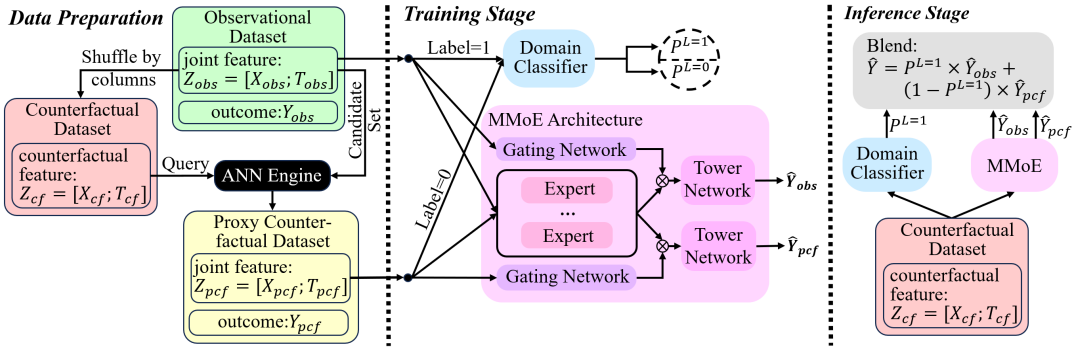



Figure 5: DSCF framework overview: data preparation, training, and inference. During inference, new samples drawn from the same counterfactual distribution as in the data preparation stage are fed into the model for prediction.

Final predictions are produced by task-specific output towers:

$$\hat{y}^{\text{obs}} = o_{\text{obs}}(\mathbf{h}^{\text{obs}}), \quad \hat{y}^{\text{pcf}} = o_{\text{pcf}}(\mathbf{h}^{\text{pcf}}).$$

During training, $\hat{y}^{\text{obs}}$ and $\hat{y}^{\text{pcf}}$ are supervised using samples from $D_{\text{obs}}$ and $D_{\text{pcf}}$, respectively.

**Domain-guided prediction fusion.** Given that observational and proxy counterfactual data originate from different regions of the true counterfactual distribution, we introduce a domain classifier to fuse the supervision signals provided by both prediction heads. Specifically, the domain classifier $g_{\text{cls}}(\cdot)$ takes the combined input $\mathbf{z} = [\mathbf{x}; \mathbf{t}]$ and outputs a confidence score: $\alpha = \sigma(g_{\text{cls}}(\mathbf{z}))$, where $\sigma(\cdot)$ denotes the sigmoid function and $\alpha \in [0, 1]$ represents the probability that the input comes from the observational domain.

We assign domain labels $L \in \{0, 1\}$, with $L = 1$ for observational samples and $L = 0$ for proxy counterfactuals. The classifier is trained independently using balanced pairs $(\mathbf{z}_{\text{obs}}, 1)$ and $(\mathbf{z}_{\text{pcf}}, 0)$, ensuring no prior bias. We deliberately decouple domain classification from the MMoE framework to prevent interference from the inductive biases of the prediction heads through gradient propagation, and to improve training stability. Importantly, to ensure semantic alignment with the prediction heads, the domain classifier is trained to distinguish $D_{\text{obs}}$ from $D_{\text{pcf}}$, rather than from synthetic counterfactuals with shuffled treatments. This design yields coherent fusion behavior and is empirically validated in ablation studies.

**Training and inference.** The MMoE-based prediction module is optimized to minimize the expected supervised loss over both data sources:

$$\mathcal{L}_{\text{reg}} = \mathbb{E}_{(\mathbf{x},\mathbf{t},y)\sim D_{\text{obs}}} \left[ \mathcal{L}(\hat{f}_{\text{obs}}(\mathbf{x}, \mathbf{t}), y) \right] + \mathbb{E}_{(\tilde{\mathbf{x}},\tilde{\mathbf{t}},\tilde{y})\sim D_{\text{pcf}}} \left[ \mathcal{L}(\hat{f}_{\text{pcf}}(\tilde{\mathbf{x}}, \tilde{\mathbf{t}}), \tilde{y}) \right],$$

where $\hat{f}_{\text{obs}}$ and $\hat{f}_{\text{pcf}}$ denote the two prediction routes within the MMoE:

$$\hat{f}_{\text{obs}}(\mathbf{x}, \mathbf{t}) := o_{\text{obs}} \left( \sum_{k=1}^{K} g_k^{\text{obs}} \cdot E_k([\mathbf{x}; \mathbf{t}]) \right),$$

$$\hat{f}_{\text{pcf}}(\tilde{\mathbf{x}}, \tilde{\mathbf{t}}) := o_{\text{pcf}} \left( \sum_{k=1}^{K} g_k^{\text{pcf}} \cdot E_k([\tilde{\mathbf{x}}; \tilde{\mathbf{t}}]) \right),$$

with $E_k$ denoting the shared experts and $g_k^{\text{obs}}, g_k^{\text{pcf}}$ the task-specific gating weights. The domain classifier is trained independently using binary cross-entropy:

$$\mathcal{L}_{\text{cls}} = \mathbb{E}_{(\mathbf{z},L)} \left[ \mathcal{L}_{\text{CE}}(\sigma(g_{\text{cls}}(\mathbf{z})), L) \right],$$

where $\mathcal{L}_{\text{CE}}$ denotes the binary cross-entropy loss, $\mathbf{z} = [\mathbf{x}; \mathbf{t}]$, and $L \in \{0, 1\}$ indicates the domain label.

At inference, final predictions are fused using the domain affinity:

$$\hat{f}_{\text{DSCF}}(\mathbf{x}, \mathbf{t}) = \alpha(\mathbf{x}, \mathbf{t}) \cdot \hat{f}_{\text{obs}}(\mathbf{x}, \mathbf{t}) + (1 - \alpha(\mathbf{x}, \mathbf{t})) \cdot \hat{f}_{\text{pcf}}(\mathbf{x}, \mathbf{t}),$$

where $\alpha(\mathbf{x}, \mathbf{t}) := \sigma(g_{\text{cls}}([\mathbf{x}; \mathbf{t}]))$ denotes the learned domain affinity.

## Theoretical Justification

We provide a theoretical justification for the DSCF framework by analyzing its expected risk under the true counterfactual distribution $P_{\text{cf}}$. Let $\hat{f}_{\text{DSCF}}$ be the final fused prediction and $\mathcal{L} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_{\geq 0}$ a pointwise loss function.

**Assumptions.** We assume the loss function $\mathcal{L}(y, \hat{y})$ is $L_\ell$-Lipschitz in $\hat{y}$ and bounded by $B_{\mathcal{L}}$. The true outcome function $y(\mathbf{x}, \mathbf{t})$ is $L_y$-Lipschitz, and both prediction heads $\hat{f}_{\text{obs}}$, $\hat{f}_{\text{pcf}}$ are $L_f$-Lipschitz and bounded in output by $B$. The domain classifier $\alpha(\mathbf{x}, \mathbf{t}) := \sigma(g_{\text{cls}}([\mathbf{x}; \mathbf{t}]))$ has classification error at most $\varepsilon_{\text{cls}}$ over a balanced mixture of $D_{\text{obs}}$ and $D_{\text{pcf}}$.

**Theorem 1** (Counterfactual Risk Bound for DSCF)**.** *Let $P_{\text{cf}}$ denote the true counterfactual distribution and $P_{\text{pcf}}$ the proxy counterfactual distribution constructed via approximate matching. Let $\mathcal{L}_{\text{obs}}(\mathbf{x}, \mathbf{t}) := \mathcal{L}(\hat{f}_{\text{obs}}(\mathbf{x}, \mathbf{t}), y(\mathbf{x}, \mathbf{t}))$ and $\mathcal{L}_{\text{pcf}}(\mathbf{x}, \mathbf{t}) := \mathcal{L}(\hat{f}_{\text{pcf}}(\mathbf{x}, \mathbf{t}), y(\mathbf{x}, \mathbf{t}))$. Then the expected counterfactual risk satisfies:*

$$\mathbb{E}_{(\mathbf{x},\mathbf{t})\sim P_{\text{cf}}} \left[ \mathcal{L}\big(\hat{f}_{\text{DSCF}}(\mathbf{x}, \mathbf{t}), y(\mathbf{x}, \mathbf{t})\big) \right] \leq \underbrace{\mathbb{E}_{(\mathbf{x},\mathbf{t})\sim P_{\text{pcf}}} \left[ \min \left\{ \mathcal{L}_{\text{obs}}(\mathbf{x}, \mathbf{t}),\ \mathcal{L}_{\text{pcf}}(\mathbf{x}, \mathbf{t}) \right\} \right]}_{\textit{oracle prediction}} + \underbrace{\varepsilon_{\text{proxy}}}_{\textit{proxy bias}} + \underbrace{B_{\mathcal{L}} \cdot \varepsilon_{\text{cls}}}_{\textit{fusion penalty}}.$$

*where $\varepsilon_{\text{proxy}} := L_\ell(L_y + L_f) \cdot \varepsilon_{\text{ANN}}$, with $\varepsilon_{\text{ANN}}$ denoting the maximal distance between a target counterfactual input and its matched proxy neighbor. The fusion penalty term is bounded by $B_{\mathcal{L}} \cdot \varepsilon_{\text{cls}}$, where $B_{\mathcal{L}} = 2L_\ell B$.*

A complete proof is given in Appendix A. Each term in the bound reflects a key design component of DSCF:

- *Oracle prediction:* captures one key benefit of joint training—by selecting the better head per input, DSCF minimizes per-sample risk; additional gains from cross-domain knowledge sharing are demonstrated empirically.
- *Proxy bias:* measures the discrepancy between $P_{\text{cf}}$ and $P_{\text{pcf}}$, bounded when ANN matching yields geometrically close neighbors, even under positivity violations.
- *Fusion penalty:* accounts for domain classification error; as $\varepsilon_{\text{cls}} \to 0$, DSCF recovers oracle performance.

**Remark 1** (Distributional Advantage over Reweighting)**.** *Proxy matching achieves a strictly smaller 1-Wasserstein distance to the true counterfactual distribution than permutation weighting under typical positivity violations (Appendix A.5). This advantage stems from proxy matching's ability to directly minimize transport cost by approximating off-support mass through nearest neighbors. In contrast, reweighting normalizes within-support density without access to unseen regions, leading to larger distributional discrepancy. Consequently, proxy matching induces a tighter risk bound under standard Lipschitz conditions.*

# Experiments

We evaluate the proposed DSCF framework on both synthetic and semi-synthetic datasets to assess its effectiveness in counterfactual prediction under high-dimensional combinatorial interventions. We compare DSCF against representative baselines and analyze the impact of its core components through detailed result analysis and ablation studies.

## Experiment Setup

**Baselines.** We compare against the following baselines. **kNN** is a non-parametric retrieval method. **S-Learner** and **$\text{NN}_{\text{pcf}}$** are supervised models trained on observational and proxy counterfactual data, respectively. **PW** (Arbour, Dimmery, and Sondhi 2021) and **VSR** (Zou et al. 2020) reweight observational samples to match the counterfactual joint distribution over covariates and interventions. **RMNet** (Tanimoto et al. 2021) learns domain-invariant representations. **H-Learner** (Chauhan et al. 2025) is a meta-learning approach for multi-intervention, multi-outcome settings, adapted here to single-outcome prediction. **DSCF-Sep** is a variant of our model that disables joint training and directly fuses S-Learner and $\text{NN}_{\text{pcf}}$ using a domain classifier. We exclude **SCP** (Qian, Curth, and van der Schaar 2021), which requires $p$ separate models and $p$-fold data augmentation, making it impractical for high-dimensional combinatorial interventions. We also exclude **Synthetic Combinations (SC)** (Agarwal, Agarwal, and Vijaykumar 2023), as its idealized setting does not align with our problem context, resulting in poor performance. See Appendix B for details.

**Implementation details.** For fair comparison, the predictive models used in S-Learner, $\text{NN}_{\text{pcf}}$, PW, VSR, and RMNet all adopt a 4-layer MLP with 128 hidden units per layer. In DSCF, each expert shares the first two layers of this architecture, with the number of experts set to 5. For H-Learner, the hypernetwork width is set to $4p$ to ensure sufficient capacity. All models are trained with the same number of epochs, learning rate, optimizer, and batch size. See Appendix B for additional details.

**Evaluation metrics.** We evaluate all models on a held-out test set drawn from the factorized counterfactual distribution $P(\mathbf{X}) \prod_{i=1}^{p} P(T^i)$. Metrics include Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), averaged over 5 random seeds.

## Synthetic Experiment on Fully Controlled Data

**Dataset.** We construct a synthetic dataset to evaluate counterfactual prediction under high-dimensional combinatorial interventions with fully controlled ground truth and realistic data characteristics. Each sample consists of a covariate vector $\mathbf{x} \in \mathbb{R}^d$, a binary treatment vector $\mathbf{t} \in \{0,1\}^p$, and a real-valued outcome $y \in \mathbb{R}$.

*Covariates.* To mimic the structure of real-world user data (e.g., long-tailed, low-rank, and nonlinear), we first sample a latent vector $\mathbf{z} \in \mathbb{R}^r$ from a 50-component Gaussian Mixture Model (GMM), where $\mathbf{z} \sim \sum_{k=1}^{50} \pi_k \cdot \mathcal{N}(\mu_k, \Sigma_k)$ and $\pi_k \propto 1/k^\alpha$ with $\alpha = 1.5$, such that the cluster weights follow a Zipf distribution. The latent vector is mapped to the covariate space via a linear projection and nonlinear transformation: $\mathbf{x} = \mathbf{W}_{\text{up}}\mathbf{z} + \epsilon$, with $\epsilon \sim \mathcal{N}(0, 0.01^2\mathbf{I})$. For nonlinearity, the first third of features undergo $\tanh$, and the second third use exponential transformation.

*Treatments.* Each treatment dimension is assigned independently using a confounded logistic model: $P(T^j = 1 \mid \mathbf{x}) = \sigma(\gamma \cdot \mathbf{x}^\top \beta^{(j)} + \eta_j)$, where $\eta_j \sim \mathcal{N}(0, 0.1^2)$ and $\gamma$ controls the confounding strength.

*Outcomes.* The outcome combines additive effects, interaction terms, and nonlinearities: $y = \mathbf{x}^\top \beta_x + \mathbf{t}^\top \beta_t + \sum_{(i,j)\in\mathcal{I}} \left(2 \cdot t_i t_j \cdot \alpha_{ij} + 0.3 \cdot t_i \cdot \mathbf{x}^\top \gamma_{ij}\right) + 0.2 \cdot \phi(\mathbf{x}, \mathbf{t}) + \epsilon$, with $\epsilon \sim \mathcal{N}(0, 0.1^2)$ and $\phi(\mathbf{x}, \mathbf{t}) = \sin(\mathbf{x}^\top w_x) + \exp(-|\mathbf{t}^\top w_t|)$. The interaction set $\mathcal{I}$ includes $\lceil c \cdot p/2 \rceil$ random treatment pairs, with $c = 5$ controlling outcome complexity.

We vary the number of treatments $p \in \{10, 20, 30\}$ and the confounding strength $\gamma \in \{0.1, 0.3, 0.5, 1.0\}$. Each training set contains $p \times 10{,}000$ samples, and the corresponding test set is drawn from the factorized counterfactual distribution with the same sample size.

**Results.** Table 1 reports RMSE and MAE of all methods under varying numbers of intervention components and confounding strengths. Overall, DSCF achieves the best performance on 21 out of 24 metrics, and ranks second on the remaining 3, consistently outperforming all baselines.

Under low confounding ($\gamma = 0.1$), H-Learner slightly outperforms DSCF at $p = 30$, but as $\gamma$ increases, it becomes unstable—its RMSE at $p = 20$ under $\gamma = 0.3$ and $\gamma = 1.0$ exceeds 15 and 40 respectively. In contrast, DSCF remains robust and consistently outperforms its variants and all baselines. Notably, $\text{NN}_{\text{pcf}}$ often surpasses more sophisticated alternatives such as PW, VSR, and RMNet. This observation supports our hypothesis that reweighting and representation-based methods become unreliable under

Table 1: Prediction errors (RMSE and MAE) on synthetic datasets. Best results are **bolded**, second-best are *italicized*.

| Method | $\gamma = 0.1$ | | $\gamma = 0.3$ | | $\gamma = 0.5$ | | $\gamma = 1.0$ | |
|---|---|---|---|---|---|---|---|---|
| | RMSE ± $\sigma$ | MAE ± $\sigma$ | RMSE ± $\sigma$ | MAE ± $\sigma$ | RMSE ± $\sigma$ | MAE ± $\sigma$ | RMSE ± $\sigma$ | MAE ± $\sigma$ |
| $p = 10$ | | | | | | | | |
| kNN | $6.393_{\pm 0.000}$ | $2.559_{\pm 0.000}$ | $7.145_{\pm 0.000}$ | $3.038_{\pm 0.000}$ | $7.528_{\pm 0.000}$ | $3.498_{\pm 0.000}$ | $7.924_{\pm 0.000}$ | $4.217_{\pm 0.000}$ |
| S-Learner | $3.952_{\pm 0.160}$ | $1.434_{\pm 0.040}$ | $5.499_{\pm 0.013}$ | $1.834_{\pm 0.008}$ | $6.164_{\pm 0.057}$ | $2.197_{\pm 0.035}$ | $6.265_{\pm 0.042}$ | $2.574_{\pm 0.010}$ |
| $NN_{pcf}$ | $3.776_{\pm 0.172}$ | $1.385_{\pm 0.037}$ | $5.330_{\pm 0.020}$ | $1.632_{\pm 0.032}$ | $5.682_{\pm 0.022}$ | $1.776_{\pm 0.003}$ | $6.130_{\pm 0.055}$ | $2.109_{\pm 0.007}$ |
| PW | $4.339_{\pm 0.098}$ | $1.485_{\pm 0.015}$ | $5.641_{\pm 0.040}$ | $1.823_{\pm 0.017}$ | $5.979_{\pm 0.076}$ | $2.079_{\pm 0.023}$ | $6.359_{\pm 0.130}$ | $2.652_{\pm 0.061}$ |
| VSR | $4.072_{\pm 0.054}$ | $1.455_{\pm 0.004}$ | $5.625_{\pm 0.077}$ | $1.864_{\pm 0.008}$ | $6.009_{\pm 0.092}$ | $2.130_{\pm 0.018}$ | $6.301_{\pm 0.031}$ | $2.662_{\pm 0.009}$ |
| RMNet | $3.869_{\pm 0.077}$ | $1.437_{\pm 0.011}$ | $5.444_{\pm 0.086}$ | $1.827_{\pm 0.037}$ | *5.657*$_{\pm 0.037}$ | $2.133_{\pm 0.027}$ | $6.227_{\pm 0.017}$ | $2.622_{\pm 0.012}$ |
| H-Learner | $3.766_{\pm 0.027}$ | *1.121*$_{\pm 0.012}$ | $5.743_{\pm 0.244}$ | $1.663_{\pm 0.086}$ | $6.532_{\pm 0.023}$ | $1.856_{\pm 0.013}$ | *5.621*$_{\pm 0.015}$ | $2.165_{\pm 0.011}$ |
| DSCF-Sep (ours) | *3.761*$_{\pm 0.131}$ | $1.229_{\pm 0.024}$ | *5.329*$_{\pm 0.014}$ | *1.554*$_{\pm 0.018}$ | $5.790_{\pm 0.017}$ | *1.771*$_{\pm 0.011}$ | $6.082_{\pm 0.036}$ | *2.097*$_{\pm 0.001}$ |
| DSCF (ours) | **3.100**$_{\pm 0.026}$ | **0.832**$_{\pm 0.017}$ | **4.469**$_{\pm 0.152}$ | **1.132**$_{\pm 0.003}$ | **5.112**$_{\pm 0.200}$ | **1.350**$_{\pm 0.048}$ | **5.207**$_{\pm 0.066}$ | **1.640**$_{\pm 0.042}$ |
| $p = 20$ | | | | | | | | |
| kNN | $6.287_{\pm 0.000}$ | $3.776_{\pm 0.000}$ | $7.364_{\pm 0.000}$ | $4.024_{\pm 0.000}$ | $7.011_{\pm 0.000}$ | $4.195_{\pm 0.000}$ | $8.895_{\pm 0.000}$ | $4.614_{\pm 0.000}$ |
| S-Learner | $4.459_{\pm 0.094}$ | $2.081_{\pm 0.037}$ | $6.423_{\pm 0.277}$ | $2.476_{\pm 0.079}$ | $5.588_{\pm 0.067}$ | $2.568_{\pm 0.074}$ | $7.849_{\pm 0.168}$ | $3.494_{\pm 0.008}$ |
| $NN_{pcf}$ | $4.580_{\pm 0.057}$ | $2.036_{\pm 0.031}$ | $6.178_{\pm 0.044}$ | $2.134_{\pm 0.018}$ | $5.642_{\pm 0.101}$ | $2.482_{\pm 0.109}$ | $7.393_{\pm 0.216}$ | $3.083_{\pm 0.061}$ |
| PW | $4.892_{\pm 0.154}$ | $1.998_{\pm 0.003}$ | $6.114_{\pm 0.133}$ | $2.393_{\pm 0.008}$ | $5.541_{\pm 0.356}$ | $2.692_{\pm 0.134}$ | $7.654_{\pm 0.242}$ | $3.302_{\pm 0.037}$ |
| VSR | $4.361_{\pm 0.092}$ | $2.016_{\pm 0.055}$ | $6.286_{\pm 0.248}$ | $2.348_{\pm 0.028}$ | *5.470*$_{\pm 0.020}$ | $2.609_{\pm 0.032}$ | $7.638_{\pm 0.126}$ | $3.250_{\pm 0.043}$ |
| RMNet | $4.827_{\pm 0.132}$ | $2.085_{\pm 0.027}$ | $6.129_{\pm 0.205}$ | $2.235_{\pm 0.015}$ | $5.488_{\pm 0.119}$ | $2.597_{\pm 0.036}$ | $7.742_{\pm 0.361}$ | $3.472_{\pm 0.126}$ |
| H-Learner | **3.469**$_{\pm 0.025}$ | *1.542*$_{\pm 0.016}$ | $15.165_{\pm 2.051}$ | *1.995*$_{\pm 0.003}$ | $6.756_{\pm 0.108}$ | *2.237*$_{\pm 0.027}$ | $41.361_{\pm 4.743}$ | $3.180_{\pm 0.012}$ |
| DSCF-Sep (ours) | $4.280_{\pm 0.061}$ | $1.756_{\pm 0.021}$ | *6.125*$_{\pm 0.125}$ | $2.010_{\pm 0.024}$ | $5.507_{\pm 0.084}$ | $2.383_{\pm 0.088}$ | *7.362*$_{\pm 0.130}$ | *3.066*$_{\pm 0.058}$ |
| DSCF (ours) | *3.725*$_{\pm 0.153}$ | **1.234**$_{\pm 0.058}$ | **5.462**$_{\pm 0.160}$ | **1.401**$_{\pm 0.118}$ | **4.673**$_{\pm 0.105}$ | **1.674**$_{\pm 0.055}$ | **6.640**$_{\pm 0.185}$ | **2.202**$_{\pm 0.193}$ |
| $p = 30$ | | | | | | | | |
| kNN | $12.178_{\pm 0.000}$ | $8.574_{\pm 0.000}$ | $16.968_{\pm 0.000}$ | $11.545_{\pm 0.000}$ | $18.101_{\pm 0.000}$ | $12.588_{\pm 0.000}$ | $19.428_{\pm 0.000}$ | $13.811_{\pm 0.000}$ |
| S-Learner | $1.742_{\pm 0.019}$ | $1.273_{\pm 0.015}$ | $4.295_{\pm 0.087}$ | $2.713_{\pm 0.041}$ | $9.196_{\pm 0.047}$ | $5.436_{\pm 0.028}$ | $11.751_{\pm 0.117}$ | $7.532_{\pm 0.065}$ |
| $NN_{pcf}$ | $1.712_{\pm 0.013}$ | $1.258_{\pm 0.009}$ | $2.956_{\pm 0.007}$ | $1.985_{\pm 0.020}$ | $6.466_{\pm 0.031}$ | $4.108_{\pm 0.036}$ | $9.385_{\pm 0.014}$ | $6.002_{\pm 0.021}$ |
| PW | $1.841_{\pm 0.012}$ | $1.310_{\pm 0.008}$ | $3.948_{\pm 0.027}$ | $2.473_{\pm 0.027}$ | $7.814_{\pm 0.088}$ | $4.600_{\pm 0.041}$ | $11.226_{\pm 0.032}$ | $7.343_{\pm 0.016}$ |
| VSR | $1.794_{\pm 0.036}$ | $1.300_{\pm 0.028}$ | $4.683_{\pm 0.006}$ | $2.947_{\pm 0.015}$ | $7.256_{\pm 0.046}$ | $4.494_{\pm 0.029}$ | $11.369_{\pm 0.040}$ | $7.326_{\pm 0.023}$ |
| RMNet | $1.887_{\pm 0.054}$ | $1.361_{\pm 0.030}$ | $3.967_{\pm 0.040}$ | $2.568_{\pm 0.023}$ | $8.074_{\pm 0.109}$ | $4.788_{\pm 0.056}$ | $11.227_{\pm 0.047}$ | $7.263_{\pm 0.033}$ |
| H-Learner | **0.820**$_{\pm 0.036}$ | **0.599**$_{\pm 0.025}$ | *2.633*$_{\pm 0.026}$ | *1.701*$_{\pm 0.007}$ | **4.806**$_{\pm 0.060}$ | *3.115*$_{\pm 0.044}$ | *7.268*$_{\pm 0.016}$ | *5.057*$_{\pm 0.013}$ |
| DSCF-Sep (ours) | $1.527_{\pm 0.013}$ | $1.078_{\pm 0.003}$ | $2.931_{\pm 0.007}$ | $1.943_{\pm 0.019}$ | $6.463_{\pm 0.031}$ | $4.098_{\pm 0.036}$ | $9.384_{\pm 0.014}$ | $6.001_{\pm 0.021}$ |
| DSCF (ours) | *0.983*$_{\pm 0.045}$ | *0.705*$_{\pm 0.021}$ | **2.458**$_{\pm 0.265}$ | **1.517**$_{\pm 0.130}$ | *5.070*$_{\pm 0.242}$ | **2.919**$_{\pm 0.068}$ | **6.823**$_{\pm 0.218}$ | **4.389**$_{\pm 0.126}$ |

Table 2: Evaluation on the semi-synthetic dataset at two data scales ($n = 0.1$M / 8M) using RMSE and MAE.

| Method | $n = 0.1$M | | $n = 8$M | |
|---|---|---|---|---|
| | RMSE ± $\sigma$ | MAE ± $\sigma$ | RMSE ± $\sigma$ | MAE ± $\sigma$ |
| kNN | $1.794_{\pm 0.000}$ | $1.201_{\pm 0.000}$ | $1.232_{\pm 0.000}$ | $0.773_{\pm 0.000}$ |
| S-Learner | $0.945_{\pm 0.009}$ | $0.559_{\pm 0.003}$ | $0.604_{\pm 0.003}$ | $0.366_{\pm 0.003}$ |
| $NN_{pcf}$ | $0.870_{\pm 0.001}$ | $0.527_{\pm 0.002}$ | $0.519_{\pm 0.003}$ | $0.288_{\pm 0.002}$ |
| PW | $1.122_{\pm 0.008}$ | $0.586_{\pm 0.002}$ | $0.568_{\pm 0.002}$ | $0.346_{\pm 0.002}$ |
| VSR | $0.874_{\pm 0.003}$ | $0.530_{\pm 0.002}$ | $0.576_{\pm 0.006}$ | $0.355_{\pm 0.004}$ |
| RMNet | $0.936_{\pm 0.005}$ | $0.550_{\pm 0.002}$ | $0.581_{\pm 0.004}$ | $0.353_{\pm 0.004}$ |
| H-Learner | $3.890_{\pm 0.344}$ | $0.577_{\pm 0.021}$ | $0.732_{\pm 0.012}$ | $0.455_{\pm 0.004}$ |
| DSCF-Sep (ours) | *0.839*$_{\pm 0.001}$ | *0.509*$_{\pm 0.001}$ | *0.504*$_{\pm 0.004}$ | *0.279*$_{\pm 0.002}$ |
| DSCF (ours) | **0.668**$_{\pm 0.001}$ | **0.398**$_{\pm 0.001}$ | **0.353**$_{\pm 0.002}$ | **0.149**$_{\pm 0.002}$ |

positivity violations, where observational support is insufficient. In such cases, nearest neighbor retrieval from the observational dataset—guided by permuted counterfactual queries—offers a simple yet effective way to perceive information beyond the observational support. This empirical pattern directly motivates our proxy construction strategy.

Finally, while the non-parametric kNN method is not expected to perform competitively, it serves as a useful reference for quantifying distributional shift. Its large and steadily increasing error with growing $p$ and $\gamma$ reflects the widening gap between the observational and counterfactual distributions.

## Semi-Synthetic Experiment on Real-World Data

**Dataset.** To evaluate DSCF on real-world data, we construct a semi-synthetic dataset based on user logs from a large-scale short-video platform. The covariates include user demographics, while the interventions consist of 50 user-content interaction variables encompassing binary indicators, categorical features, and continuous scores (e.g., play duration, clicks, shares). The prediction target is the change in the user's monthly lifetime.

To fully leverage real-world business data while retaining full control over the outcome generation process, we adopt a two-stage procedure—comprising (1) parameter estimation and (2) sample generation—to construct semi-synthetic data that: (i) preserves the empirical input distribution, (ii) captures realistic outcome variability (e.g., long-tailed behavior), (iii) avoids model-induced bias, and (iv) maintains sufficient functional complexity. Specifically, we first apply a fixed, non-trainable two-layer MLP with Kaiming initialization to the concatenated input $[\mathbf{x}; \mathbf{t}]$, producing latent representations $\mathbf{z} = f([\mathbf{x}; \mathbf{t}]) \in \mathbb{R}^k$. We then estimate a symmetric matrix $\mathbf{M} \in \mathbb{R}^{k \times k}$ by minimizing the squared error between the quadratic form $\mathbf{z}^\top \mathbf{M} \mathbf{z}$ and the observed outcome $y$ across the observational dataset. After fixing the parameters $f(\cdot)$ and $\mathbf{M}$, we compute synthetic outcomes for both original and permuted input pairs as $y = \mathbf{z}^\top \mathbf{M} \mathbf{z} + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 0.1^2)$.

To evaluate model performance under different data regimes, we use the full dataset of 8 million samples for training and testing, and additionally report results on a low-resource subset containing 0.1 million samples.

**Results.** Table 2 reports the performance of all methods on the semi-synthetic dataset. DSCF consistently outperforms all baselines across both evaluation metrics and data scales. The performance gap between DSCF and the second-best method ($NN_{pcf}$) widens with increased training data: under

Table 3: Ablation results across varying confounding strengths $\gamma$, with RMSE and MAE averaged over $p \in \{10, 20, 30\}$.

| id | Reg Data | | Reg Model | Cls Data | | | Output | $\gamma = 0.1$ | | $\gamma = 0.3$ | | $\gamma = 0.5$ | | $\gamma = 1.0$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $D_{obs}$ | $D_{pcf}$ | | $D_{obs}$ | $D_{pcf}$ | $D_{cf}$ | | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE |
| (1) | ✓ | | MLP | | | | | 3.374 | 1.596 | 5.405 | 2.341 | 6.982 | 3.400 | 8.622 | 4.534 |
| (2) | | ✓ | MLP | | | | | 3.356 | 1.560 | 4.821 | 1.917 | 5.930 | 2.789 | 7.636 | 3.731 |
| (3) | ✓ | ✓ | MLP×2 | ✓ | ✓ | | Reweighted | 3.189 | 1.354 | 4.795 | 1.836 | 5.920 | 2.751 | 7.610 | 3.722 |
| (4) | ✓ | ✓ | HardShare | | | | Obs Head | 3.183 | 1.371 | 4.762 | 1.824 | 6.239 | 2.812 | 7.688 | 3.701 |
| (5) | ✓ | ✓ | HardShare | | | | Pcf Head | 3.161 | 1.345 | 4.714 | 1.819 | 6.210 | 2.770 | 7.737 | 3.696 |
| (6) | ✓ | | MMoE | | | | | 2.988 | 1.155 | 4.604 | 1.725 | 5.695 | 2.475 | 7.516 | 3.688 |
| (7) | | ✓ | MMoE | | | | | 2.782 | 1.097 | 4.473 | 1.559 | 5.431 | 2.253 | 6.995 | 3.211 |
| (8) | ✓ | ✓ | MMoE | | | | Obs Head | 2.658 | 0.999 | 4.387 | 1.489 | 5.214 | 2.172 | 6.462 | 2.993 |
| (9) | ✓ | ✓ | MMoE | | | | Pcf Head | 2.610 | 0.968 | 4.155 | 1.372 | 4.973 | 1.994 | 6.290 | 2.751 |
| (10) | ✓ | ✓ | MMoE | | | | avg | **2.602** | 0.924 | 4.197 | 1.372 | 5.005 | 2.012 | 6.248 | 2.779 |
| (11) | ✓ | ✓ | MMoE | ✓ | | ✓ | Reweighted | 2.603 | 0.931 | 4.159 | 1.358 | 4.962 | 1.986 | 6.236 | 2.747 |
| (12) | ✓ | ✓ | MMoE-lite | ✓ | ✓ | | Reweighted | 2.664 | **0.881** | 4.318 | 1.402 | 5.430 | 2.175 | 6.714 | 3.025 |
| (13) | ✓ | ✓ | MMoE | ✓ | ✓ | | Reweighted | 2.603 | 0.924 | **4.130** | **1.350** | **4.952** | **1.981** | **6.223** | **2.743** |

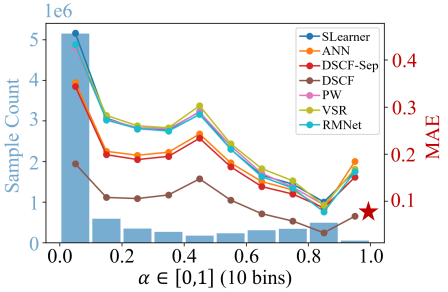



Figure 6: Model performance (MAE) across different values of domain affinity $\alpha$, with sample counts shown in blue bars.

the low-resource setting ($n = 0.1$M), DSCF reduces RMSE and MAE by 23.3% and 24.3%, respectively; under the high-resource setting ($n = 8$M), the improvements grow to 32.1% (RMSE) and 48.3% (MAE). These results highlight DSCF's superior scalability and its enhanced ability to exploit large-scale data.

Figure 6 reports the MAE of different methods across test samples grouped by their domain affinity scores $\alpha$, as predicted by the domain classifier. The vast majority of samples fall into the counterfactual region ($\alpha \in [0.0, 0.1]$), indicating a pronounced distributional shift under high-dimensional combinatorial interventions, where most target configurations are rarely or never observed.

The methods exhibit a clear three-tier performance hierarchy. Traditional approaches (e.g., S-Learner, PW, RMNet, VSR) exhibit nearly indistinguishable performance across bins but degrade sharply in low-$\alpha$ regions, revealing their limitations under severe positivity violations. Proxy-based methods ($\text{NN}_{\text{pcf}}$, DSCF-Sep) improve upon this by leveraging matched samples. They achieve progressively lower MAE compared to first-group methods as test samples move closer to the counterfactual region, demonstrating a stronger ability to capture off-support patterns. At the top tier, DSCF consistently outperforms all baselines by a wide margin across all affinity bins. The near-uniform MAE improvements over second-tier methods across bins demonstrate that joint training and a shared expert pool allow the observational branch to inject high-entropy, domain-specific information into the proxy branch. Moreover, in observational regions (high $\alpha$), DSCF even surpasses the observational-only S-Learner, due to the unbiased supervision from the proxy branch, which in turn regularizes the observational head during joint training.

## Ablation Study

Table 3 presents ablations to isolate the contributions of DSCF's three key components: proxy counterfactual data, multi-source joint training, and domain-guided fusion. Row indices refer to those shown in the table. For more detailed results and explanations, please refer to Appendix B.

*Proxy data.* Rows (1)(2) and (6)(7) reveal that training on $D_{\text{pcf}}$ alone already surpasses $D_{\text{obs}}$ alone across all $\gamma$, confirming the strong signal supplied by our proxy construction.

*Joint training.* Even with a simple hard-shared backbone, combining the two data sources (rows (4)(5)) yields sizeable gains over single-source MLPs. Replacing hard sharing with MMoE (rows (8)(9)) brings further improvement. Crucially, under the same MMoE architecture, joint training (8)(9) outperforms the single-source variants (6)(7), indicating that the gains arise from leveraging complementary training distributions, rather than from structural complexity alone.

*Domain-guided fusion.* Comparing rows (1)(2)(3) and (10)(11)(13) demonstrates that adaptive weighting consistently reduces both RMSE and MAE. The advantage widens when $\alpha$ is small, i.e., when observational samples still occupy a non-negligible fraction of the counterfactual domain. This highlights the fusion module's ability to exploit complementary signals in partially overlapping data.

*Model capacity control.* Row (12) reduces every hidden dimension of the expert, gate, and tower networks in the prediction module by half, yielding a parameter count comparable to the MLP baselines. Despite this, it still achieves superior performance, confirming that our gains stem from architectural design rather than over-parameterization.

## Conclusion

We present DSCF, a principled and scalable framework for counterfactual prediction under high-dimensional combinatorial interventions. By jointly leveraging observational data and proxy counterfactual samples through a dual-head MMoE architecture and domain-guided fusion, DSCF balances bias reduction and information richness without relying on strong structural assumptions. Theoretically, we establish a novel risk bound under the true counterfactual distribution, decomposing the estimation error into oracle prediction, proxy bias, and fusion penalty, each of which is tightly aligned with a corresponding model component. Empirically, DSCF consistently outperforms existing methods across synthetic and semi-synthetic benchmarks, demonstrating superior robustness, scalability, and generalization.

## References


Agarwal, A.; Agarwal, A.; and Vijaykumar, S. 2023. Synthetic combinations: A causal inference framework for combinatorial interventions. *Advances in Neural Information Processing Systems*, 36: 19195–19216.

Arava, S. K.; Dong, C.; Yan, Z.; Pani, A.; et al. 2018. Deep neural net with attention for multi-channel multi-touch attribution. *arXiv preprint arXiv:1809.02230*.

Arbour, D.; Dimmery, D.; and Sondhi, A. 2021. Permutation weighting. In *International Conference on Machine Learning*, 331–341. PMLR.

Chauhan, V. K.; Clifton, L.; Nigam, G.; and Clifton, D. A. 2025. Individualised Treatment Effects Estimation with Composite Treatments and Composite Outcomes. *arXiv preprint arXiv:2502.08282*.

Chesnaye, N. C.; Stel, V. S.; Tripepi, G.; Dekker, F. W.; Fu, E. L.; Zoccali, C.; and Jager, K. J. 2022. An introduction to inverse probability of treatment weighting in observational research. *Clinical kidney journal*, 15(1): 14–20.

Cortes, C.; Mansour, Y.; and Mohri, M. 2010. Learning bounds for importance weighting. *Advances in neural information processing systems*, 23.

Fournier, N.; and Guillin, A. 2015. On the rate of convergence in Wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3): 707–738.

Johansson, F. D.; Sontag, D.; and Ranganath, R. 2019. Support and invertibility in domain-invariant representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 527–536. PMLR.

Johnson, J.; Douze, M.; and Jégou, H. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*.

Ma, J.; Zhao, Z.; Yi, X.; Chen, J.; Hong, L.; and Chi, E. H. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 1930–1939.

M-Squared. 2025. How High Value Actions (HVAs) are Reshaping Marketing Mix Models. https://msquared.club/blogs/attribution-today/. "High Value Actions are meaningful digital behaviors that signal consumer interest, intent, or future purchase likelihood.".

Pearl, J. 2010. Causal inference. *Causality: objectives and assessment*, 39–58.

Qian, Z.; Curth, A.; and van der Schaar, M. 2021. Estimating multi-cause treatment effects via single-cause perturbation. *Advances in Neural Information Processing Systems*, 34: 23754–23767.

Ren, K.; Fang, Y.; Zhang, W.; Liu, S.; Li, J.; Zhang, Y.; Yu, Y.; and Wang, J. 2018. Learning multi-touch conversion attribution with dual-attention mechanisms for online advertising. In *Proceedings of the 27th acm international conference on information and knowledge management*, 1433–1442.

Rosenbaum, P. R.; and Rubin, D. B. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1): 41–55.

Schwab, P.; Linhardt, L.; and Karlen, W. 2018. Perfect match: A simple method for learning representations for counterfactual inference with neural networks. *arXiv preprint arXiv:1810.00656*.

Shalit, U.; Johansson, F. D.; and Sontag, D. 2017. Estimating individual treatment effect: generalization bounds and algorithms. In *International conference on machine learning*, 3076–3085. PMLR.

Shi, C.; Blei, D.; and Veitch, V. 2019. Adapting neural networks for the estimation of treatment effects. *Advances in neural information processing systems*, 32.

Shi, W.; Fu, C.; Xu, Q.; Chen, S.; Zhang, J.; Zhu, Q.; Hua, Z.; and Yang, S. 2024. Ads Supply Personalization via Doubly Robust Learning. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 4874–4881.

Stuart, E. A. 2010. Matching methods for causal inference: A review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 25(1): 1.

Tanimoto, A.; Sakai, T.; Takenouchi, T.; and Kashima, H. 2021. Regret minimization for causal inference on large treatment space. In *International Conference on Artificial Intelligence and Statistics*, 946–954. PMLR.

Wang, Y.; Li, H.; Zhu, M.; Wu, A.; Xiong, R.; Wu, F.; and Kuang, K. 2024. Causal Inference with Complex Treatments: A Survey. *arXiv preprint arXiv:2407.14022*.

Wu, A.; Kuang, K.; Xiong, R.; Li, B.; and Wu, F. 2023. Stable estimation of heterogeneous treatment effects. In *International Conference on Machine Learning*, 37496–37510. PMLR.

Yao, D.; Gong, C.; Zhang, L.; Chen, S.; and Bi, J. 2022. CausalMTA: Eliminating the user confounding bias for causal multi-touch attribution. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 4342–4352.

Zou, H.; Cui, P.; Li, B.; Shen, Z.; Ma, J.; Yang, H.; and He, Y. 2020. Counterfactual prediction for bundle treatment. *Advances in Neural Information Processing Systems*, 33: 19705–19715.

# Appendix

## A Detailed Theoretical Analysis

We provide a detailed proof of Theorem 1, which establishes an upper bound on the expected counterfactual risk of DSCF. The bound decomposes into three interpretable components: an *oracle prediction* term capturing the per-sample minimum loss between the two prediction heads, a *proxy bias* term measuring the approximation error between proxy and true counterfactual distributions, and a *fusion penalty* due to domain classification inaccuracy.

To support this result, we begin by formalizing the learning setup and regularity assumptions. We then prove a proxy risk approximation bound based on nearest-neighbor matching, followed by a detailed derivation of the main risk bound. Finally, we provide a distributional comparison with permutation weighting (PW), showing that proxy matching induces a strictly lower Wasserstein distance to the target counterfactual distribution under positivity violations, which leads to tighter lower bounds on achievable risk.

### A.1 Setup and Definitions

Let $P_{\mathrm{cf}}$ denote the target counterfactual distribution and $P_{\mathrm{pcf}}$ the proxy counterfactual distribution obtained via ANN matching. The learned prediction function is a soft fusion:

$$\hat{f}_{\mathrm{DSCF}}(\mathbf{x}, \mathbf{t}) := \alpha(\mathbf{x}, \mathbf{t}) \cdot \hat{f}_{\mathrm{obs}}(\mathbf{x}, \mathbf{t}) + (1 - \alpha(\mathbf{x}, \mathbf{t})) \cdot \hat{f}_{\mathrm{pcf}}(\mathbf{x}, \mathbf{t}),$$

where $\alpha(\mathbf{x}, \mathbf{t}) := \sigma(g_{\mathrm{cls}}([\mathbf{x}; \mathbf{t}])) \in [0, 1]$ is the learned domain affinity score. Let $\mathcal{L} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_{\geq 0}$ denote a scalar loss function, and define the pointwise losses:

$$\mathcal{L}_{\mathrm{obs}}(\mathbf{x}, \mathbf{t}) := \mathcal{L}(\hat{f}_{\mathrm{obs}}(\mathbf{x}, \mathbf{t}),\ y(\mathbf{x}, \mathbf{t})),$$
$$\mathcal{L}_{\mathrm{pcf}}(\mathbf{x}, \mathbf{t}) := \mathcal{L}(\hat{f}_{\mathrm{pcf}}(\mathbf{x}, \mathbf{t}),\ y(\mathbf{x}, \mathbf{t})),$$
$$\mathcal{L}_{\mathrm{DSCF}}(\mathbf{x}, \mathbf{t}) := \mathcal{L}(\hat{f}_{\mathrm{DSCF}}(\mathbf{x}, \mathbf{t}),\ y(\mathbf{x}, \mathbf{t})).$$

Our goal is to upper bound the population risk under the true counterfactual distribution:

$$\mathcal{R}_{P_{\mathrm{cf}}}(\hat{f}_{\mathrm{DSCF}}) := \mathbb{E}_{(\mathbf{x}, \mathbf{t}) \sim P_{\mathrm{cf}}} \left[ \mathcal{L}_{\mathrm{DSCF}}(\mathbf{x}, \mathbf{t}) \right].$$

### A.2 Assumptions

We assume the following standard conditions:

**Assumption A1 (Lipschitz Regularity).** We assume the following Lipschitz conditions hold:

- The loss function $\mathcal{L}(y, \hat{y})$ is $L_\ell$-Lipschitz in its second argument and bounded above by $B_{\mathcal{L}}$:
$$|\mathcal{L}(y, \hat{y}_1) - \mathcal{L}(y, \hat{y}_2)| \leq L_\ell \cdot |\hat{y}_1 - \hat{y}_2|, \quad 0 \leq \mathcal{L}(y, \hat{y}) \leq B_{\mathcal{L}}.$$
- The true outcome function $y(\mathbf{x}, \mathbf{t})$ is $L_y$-Lipschitz in its inputs:
$$|y(\mathbf{x}, \mathbf{t}) - y(\mathbf{x}', \mathbf{t}')| \leq L_y \cdot \|(\mathbf{x}, \mathbf{t}) - (\mathbf{x}', \mathbf{t}')\|.$$
- The learned prediction function $f(\mathbf{x}, \mathbf{t})$ is $L_f$-Lipschitz in its inputs:
$$|f(\mathbf{x}, \mathbf{t}) - f(\mathbf{x}', \mathbf{t}')| \leq L_f \cdot \|(\mathbf{x}, \mathbf{t}) - (\mathbf{x}', \mathbf{t}')\|.$$

**Assumption A4 (Domain Classification Error).** Define the domain label $L(\mathbf{x}, \mathbf{t}) \in \{0, 1\}$, where $L = 1$ if the sample is from the observational domain and $L = 0$ if from the proxy domain. Let $\hat{L}(\mathbf{x}, \mathbf{t}) := \mathbf{1}[\alpha(\mathbf{x}, \mathbf{t}) > 0.5]$. Then the classification error under a 50-50 mixture distribution $P_{\mathrm{mix}} := \frac{1}{2}(P_{\mathrm{obs}} + P_{\mathrm{pcf}})$ is bounded:

$$\mathbb{P}_{(\mathbf{x}, \mathbf{t}) \sim P_{\mathrm{mix}}} \left[ \hat{L}(\mathbf{x}, \mathbf{t}) \neq L(\mathbf{x}, \mathbf{t}) \right] \leq \varepsilon_{\mathrm{cls}}.$$

### A.3 Proxy Risk Approximation

We now quantify the discrepancy between the proxy and target counterfactual risks under the assumption that proxy samples are matched within a bounded neighborhood. This result provides a formal justification for using proxy samples as a surrogate for unobserved counterfactuals when estimating the target risk.

**Lemma A.1** (Proxy Risk Approximation)**.** *Let $f : \mathcal{X} \times \mathcal{T} \to \mathbb{R}$ be any prediction function. Suppose that for every $(\mathbf{x}, \mathbf{t}) \sim P_{\mathrm{cf}}$, there exists a matched sample $(\mathbf{x}', \mathbf{t}') \sim P_{\mathrm{pcf}}$ such that*

$$\|(\mathbf{x}, \mathbf{t}) - (\mathbf{x}', \mathbf{t}')\| \leq \varepsilon_{\mathrm{ANN}}.$$

*Then under Assumptions A1, the expected risks satisfy*

$$\left| \mathcal{R}_{P_{\mathrm{cf}}}(f) - \mathcal{R}_{P_{\mathrm{pcf}}}(f) \right| \leq L_l \cdot (L_y + L_f) \cdot \varepsilon_{\mathrm{ANN}} := \varepsilon_{\mathrm{proxy}}.$$

*Proof.* Recall the expected risk under each distribution:

$$\mathcal{R}_{P_{\mathrm{cf}}}(f) = \mathbb{E}_{(\mathbf{x},\mathbf{t})\sim P_{\mathrm{cf}}}\left[\mathcal{L}(f(\mathbf{x},\mathbf{t}),\ y(\mathbf{x},\mathbf{t}))\right],$$
$$\mathcal{R}_{P_{\mathrm{pcf}}}(f) = \mathbb{E}_{(\mathbf{x},\mathbf{t})\sim P_{\mathrm{cf}}}\left[\mathcal{L}(f(\mathbf{x}',\mathbf{t}'),\ y(\mathbf{x}',\mathbf{t}'))\right],$$

where $(\mathbf{x},\mathbf{t}) \sim P_{\mathrm{cf}}$ and $(\mathbf{x}',\mathbf{t}')$ is its nearest matched point in $P_{\mathrm{pcf}}$ such that

$$\|(\mathbf{x},\mathbf{t}) - (\mathbf{x}',\mathbf{t}')\| \le \varepsilon_{\mathrm{ANN}}.$$

We compare the pointwise losses:

$$\begin{aligned} & |\mathcal{L}(f(\mathbf{x},\mathbf{t}),\ y(\mathbf{x},\mathbf{t})) - \mathcal{L}(f(\mathbf{x}',\mathbf{t}'),\ y(\mathbf{x}',\mathbf{t}'))| \\ \le\ & |\mathcal{L}(f(\mathbf{x},\mathbf{t}),\ y(\mathbf{x},\mathbf{t})) - \mathcal{L}(f(\mathbf{x},\mathbf{t}),\ y(\mathbf{x}',\mathbf{t}'))| \\ & + |\mathcal{L}(f(\mathbf{x},\mathbf{t}),\ y(\mathbf{x}',\mathbf{t}')) - \mathcal{L}(f(\mathbf{x}',\mathbf{t}'),\ y(\mathbf{x}',\mathbf{t}'))|. \end{aligned}$$

We bound both terms using Assumption A1:

- First term (label shift):
$$|y(\mathbf{x},\mathbf{t}) - y(\mathbf{x}',\mathbf{t}')| \le L_y \cdot \varepsilon_{\mathrm{ANN}} \quad \Rightarrow \quad \text{term} \le L_l \cdot L_y \cdot \varepsilon_{\mathrm{ANN}}.$$
- Second term (prediction shift):
$$|f(\mathbf{x},\mathbf{t}) - f(\mathbf{x}',\mathbf{t}')| \le L_f \cdot \varepsilon_{\mathrm{ANN}} \quad \Rightarrow \quad \text{term} \le L_l \cdot L_f \cdot \varepsilon_{\mathrm{ANN}}.$$

Summing both bounds:

$$|\mathcal{L}(f(\mathbf{x},\mathbf{t}), y(\mathbf{x},\mathbf{t})) - \mathcal{L}(f(\mathbf{x}',\mathbf{t}'), y(\mathbf{x}',\mathbf{t}'))| \le L_l \cdot (L_y + L_f) \cdot \varepsilon_{\mathrm{ANN}}.$$

Taking expectation over $(\mathbf{x},\mathbf{t}) \sim P_{\mathrm{cf}}$, and noting the one-to-one matching with $(\mathbf{x}',\mathbf{t}') \sim P_{\mathrm{pcf}}$, we conclude:

$$\left|\mathcal{R}_{P_{\mathrm{cf}}}(f) - \mathcal{R}_{P_{\mathrm{pcf}}}(f)\right| \le L_l \cdot (L_y + L_f) \cdot \varepsilon_{\mathrm{ANN}} = \varepsilon_{\mathrm{proxy}}.$$ □

### A.4 Proof of Theorem 1

**Theorem A.1** (Risk Bound of DSCF)**.** *Let $P_{\mathrm{cf}}$ denote the true counterfactual distribution and $P_{\mathrm{pcf}}$ the proxy distribution constructed via approximate matching. Let $\mathcal{L}_{\mathrm{obs}}(\mathbf{x},\mathbf{t}) := \mathcal{L}(\hat{f}_{\mathrm{obs}}(\mathbf{x},\mathbf{t}), y(\mathbf{x},\mathbf{t}))$ and $\mathcal{L}_{\mathrm{pcf}}(\mathbf{x},\mathbf{t}) := \mathcal{L}(\hat{f}_{\mathrm{pcf}}(\mathbf{x},\mathbf{t}), y(\mathbf{x},\mathbf{t}))$. Then the expected counterfactual risk satisfies:*

$$\mathbb{E}_{(\mathbf{x},\mathbf{t})\sim P_{\mathrm{cf}}}\left[\mathcal{L}\big(\hat{f}_{\mathrm{DSCF}}(\mathbf{x},\mathbf{t}), y(\mathbf{x},\mathbf{t})\big)\right] \le \underbrace{\mathbb{E}_{(\mathbf{x},\mathbf{t})\sim P_{\mathrm{pcf}}}\left[\min\left\{\mathcal{L}_{\mathrm{obs}}(\mathbf{x},\mathbf{t}),\ \mathcal{L}_{\mathrm{pcf}}(\mathbf{x},\mathbf{t})\right\}\right]}_{\textit{oracle prediction}} + \underbrace{\varepsilon_{\mathrm{proxy}}}_{\textit{proxy bias}} + \underbrace{B_{\mathcal{L}} \cdot \varepsilon_{\mathrm{cls}}}_{\textit{fusion penalty}}.$$

*where $\varepsilon_{\mathrm{proxy}} := L_\ell(L_y + L_f) \cdot \varepsilon_{\mathrm{ANN}}$, with $\varepsilon_{\mathrm{ANN}}$ denoting the maximal distance between a target counterfactual input and its matched proxy neighbor. The fusion penalty term is bounded by $B_{\mathcal{L}} \cdot \varepsilon_{\mathrm{cls}}$, where $B_{\mathcal{L}} = 2L_\ell B$.*

*Proof.* The proof proceeds in three steps.

**Step 1: Reduction to Proxy Risk.** From Lemma A.1, we have:

$$\mathcal{R}_{P_{\mathrm{cf}}}(\hat{f}_{\mathrm{DSCF}}) = \mathcal{R}_{P_{\mathrm{pcf}}}(\hat{f}_{\mathrm{DSCF}}) + \left[\mathcal{R}_{P_{\mathrm{cf}}}(\hat{f}_{\mathrm{DSCF}}) - \mathcal{R}_{P_{\mathrm{pcf}}}(\hat{f}_{\mathrm{DSCF}})\right] \le \mathcal{R}_{P_{\mathrm{pcf}}}(\hat{f}_{\mathrm{DSCF}}) + \varepsilon_{\mathrm{proxy}}.$$

**Step 2: Decomposition under Proxy Risk.** We now analyze $\mathcal{L}_{\mathrm{DSCF}}(\mathbf{x},\mathbf{t})$ pointwise for $(\mathbf{x},\mathbf{t}) \sim P_{\mathrm{pcf}}$. Let:

$$\hat{f}_{\mathrm{DSCF}} := \alpha \cdot \hat{f}_{\mathrm{obs}} + (1-\alpha) \cdot \hat{f}_{\mathrm{pcf}}, \quad f^* := \arg\min\{\mathcal{L}_{\mathrm{obs}},\ \mathcal{L}_{\mathrm{pcf}}\}.$$

Then, using triangle inequality and Lipschitz continuity:

$$\begin{aligned} \mathcal{L}_{\mathrm{DSCF}} &= \mathcal{L}(y, \hat{f}_{\mathrm{DSCF}}) \\ &\le \mathcal{L}(y, f^*) + |\mathcal{L}(y, \hat{f}_{\mathrm{DSCF}}) - \mathcal{L}(y, f^*)| \\ &\le \mathcal{L}(y, f^*) + L_l \cdot |\hat{f}_{\mathrm{DSCF}} - f^*|. \end{aligned}$$

We now bound the prediction deviation $|\hat{f}_{\mathrm{DSCF}} - f^*|$. There are two cases:

- If $f^* = \hat{f}_{\mathrm{pcf}}$, then:
$$\hat{f}_{\mathrm{DSCF}} - f^* = \alpha \cdot (\hat{f}_{\mathrm{obs}} - \hat{f}_{\mathrm{pcf}}).$$

- If $f^* = \hat{f}_{\text{obs}}$, then:

$$\hat{f}_{\text{DSCF}} - f^* = (1-\alpha) \cdot (\hat{f}_{\text{pcf}} - \hat{f}_{\text{obs}}).$$

In both cases, we have:

$$|\hat{f}_{\text{DSCF}} - f^*| \leq |\alpha - L| \cdot |\hat{f}_{\text{obs}} - \hat{f}_{\text{pcf}}|.$$

Assuming both heads are bounded in range (say by $B$), then:

$$|\hat{f}_{\text{obs}} - \hat{f}_{\text{pcf}}| \leq 2B.$$

Hence:

$$\mathcal{L}_{\text{DSCF}} \leq \min\{\mathcal{L}_{\text{obs}}, \mathcal{L}_{\text{pcf}}\} + 2L_l B \cdot |\alpha - L|.$$

Taking expectation over $(\mathbf{x}, \mathbf{t}) \sim P_{\text{pcf}}$ (for which $L = 0$) gives:

$$\mathcal{R}_{P_{\text{pcf}}}(\hat{f}_{\text{DSCF}}) \leq \mathbb{E}_{P_{\text{pcf}}}[\min\{\mathcal{L}_{\text{obs}}, \mathcal{L}_{\text{pcf}}\}] + 2L_l B \cdot \mathbb{E}_{P_{\text{pcf}}}[\alpha].$$

By Assumption A4, the expected error in classifying proxy samples is bounded:

$$\mathbb{E}_{P_{\text{pcf}}}[\alpha(\mathbf{x}, \mathbf{t})] \leq \varepsilon_{\text{cls}}.$$

Letting $B_{\mathcal{L}} := 2L_l B$, we get:

$$\mathcal{R}_{P_{\text{pcf}}}(\hat{f}_{\text{DSCF}}) \leq \mathbb{E}_{P_{\text{pcf}}}[\min\{\mathcal{L}_{\text{obs}}, \mathcal{L}_{\text{pcf}}\}] + B_{\mathcal{L}} \cdot \varepsilon_{\text{cls}}.$$

**Step 3: Final Bound.** Combining the two steps:

$$\mathcal{R}_{P_{\text{cf}}}(\hat{f}_{\text{DSCF}}) \leq \mathcal{R}_{P_{\text{pcf}}}(\hat{f}_{\text{DSCF}}) + \varepsilon_{\text{proxy}} \leq \mathbb{E}_{P_{\text{pcf}}}[\min\{\mathcal{L}_{\text{obs}}, \mathcal{L}_{\text{pcf}}\}] + \varepsilon_{\text{proxy}} + B_{\mathcal{L}} \cdot \varepsilon_{\text{cls}}.$$

□

### A.5 Distributional Advantage of Proxy Matching

This section provides a theoretical justification for the use of matching in the construction of proxy counterfactual training data. Specifically, we show that the proxy counterfactual distribution produced by nearest-neighbor matching is geometrically closer—under the 1-Wasserstein distance—to the true counterfactual distribution than the reweighted distribution obtained via permutation weighting (PW), especially under violations of the positivity assumption. The result provides a distribution-level rationale for preferring matching over reweighting in high-dimensional, support-mismatched settings.

**Definition A.1** (1-Wasserstein Distance)**.** *Let $P, Q$ be probability distributions over $\mathcal{Z} = \mathcal{X} \times \mathcal{T}$. Given a cost function $c : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}_{\geq 0}$ (e.g., Euclidean distance), the 1-Wasserstein distance between $P$ and $Q$ is defined as*

$$W_1(P, Q) := \inf_{\pi \in \Pi(P,Q)} \mathbb{E}_{(z,z') \sim \pi} \left[ c(z, z') \right],$$

*where $\Pi(P, Q)$ denotes the set of couplings with marginals $P$ and $Q$.*

**Theorem A.2** (Matching Induces Lower 1-Wasserstein Distance)**.** *Let $(\mathcal{Z}, d)$ be a separable metric space with bounded metric $d$, which also serves as the cost function in the definition of 1-Wasserstein distance. Let $P_{\text{cf}}$ be the true counterfactual distribution and $P_{\text{obs}}$ the observational distribution. Let $S_{\text{obs}} := \text{supp}(P_{\text{obs}})$, and suppose the positivity assumption is violated:*

$$A := \mathcal{Z} \setminus S_{\text{obs}}, \quad \textit{with} \quad P_{\text{cf}}(A) = 1 - \lambda > 0.$$

*We compare two distributions:*

- *The **permutation weighting (PW)** distribution $P_{\text{PW}}$, constructed by reweighting the observational distribution $P_{\text{obs}}$ using normalized importance weights:*

$$P_{\text{PW}}(\mathrm{d}z) := \frac{w(z)}{\int_{S_{\text{obs}}} w(z') \, \mathrm{d}P_{\text{obs}}(z')} \cdot P_{\text{obs}}(\mathrm{d}z), \quad \textit{where} \quad w(z) := \frac{\mathrm{d}P_{\text{cf}}}{\mathrm{d}P_{\text{obs}}}(z).$$

  *Here, $w(z)$ denotes the Radon-Nikodym derivative of $P_{\text{cf}}$ with respect to $P_{\text{obs}}$, assumed to exist on $S_{\text{obs}}$, and is set to zero outside of* $\text{supp}(P_{\text{obs}})$.
- *The **proxy matching** distribution $P_{\text{pcf}} := m_{\#} P_{\text{cf}}$, defined by mapping each $z \sim P_{\text{cf}}$ to its nearest neighbor $m(z) := \arg\min_{z' \in S_{\text{obs}}} d(z, z')$.*

*Then:*

$$W_1(P_{\text{cf}}, P_{\text{pcf}}) \leq W_1(P_{\text{cf}}, P_{\text{PW}}),$$

*with strict inequality under mild conditions.*

*Proof.* The proof proceeds in four steps.

**Step 1: Define a valid coupling for proxy matching.** We define a joint distribution (coupling) $\pi_{\text{match}} := (\text{id}, m)_{\#} P_{\text{cf}}$, i.e., pushforward of $P_{\text{cf}}$ under the mapping $z \mapsto (z, m(z))$. This coupling pairs each sample $z$ with its nearest neighbor $m(z) \in S_{\text{obs}}$. Since $m(z) \in S_{\text{obs}}$ for all $z$, we know that for $z \in A$, $d(z, m(z))$ measures how far $z$ is from the observed support. For $z \in S_{\text{obs}}$, we have $m(z) = z$, and hence $d(z, m(z)) = 0$.

The transport cost under this coupling is:

$$W_1(P_{\text{cf}}, P_{\text{pcf}}) \leq \int_{\mathcal{Z}} d(z, m(z)) \, \mathrm{d}P_{\text{cf}}(z) = \int_A d(z, m(z)) \, \mathrm{d}P_{\text{cf}}(z) =: \rho.$$

This defines $\rho$ as the expected geometric projection cost from the counterfactual distribution to the observed support. This inequality holds because Wasserstein distance is defined as the *minimum* expected cost over all possible couplings. So any specific coupling (like the one induced by matching) yields an upper bound.

**Step 2: Lower bound for permutation weighting.** Let $\pi^\star$ be any optimal coupling between $P_{\text{PW}}$ and $P_{\text{cf}}$. Since $P_{\text{PW}}$ assigns no mass to $A$, but $P_{\text{cf}}$ assigns mass $1 - \lambda > 0$, the coupling must move this mass into $S_{\text{obs}}$:

$$\pi^\star(A, S_{\text{obs}}) = 1 - \lambda.$$

*Explanation:* The term $\pi^\star(A, S_{\text{obs}})$ denotes the joint probability mass that the coupling $\pi^\star$ assigns to all pairs $(z, z')$ where $z \in A := \mathcal{Z} \setminus S_{\text{obs}}$ and $z' \in S_{\text{obs}}$. In other words,

$$\pi^\star(A, S_{\text{obs}}) := \int_{A \times S_{\text{obs}}} \mathrm{d}\pi^\star(z, z').$$

Since $\pi^\star$ must preserve the marginal $P_{\text{cf}}$ on the first coordinate (by definition of a coupling), and $P_{\text{cf}}(A) = 1 - \lambda$, all this mass must be transported to some point $z' \in S_{\text{obs}}$, because $P_{\text{PW}}$ has no support outside $S_{\text{obs}}$. Thus, the coupling must assign exactly $1 - \lambda$ mass to the set $A \times S_{\text{obs}}$.

For any $z \in A$ and $z' \in S_{\text{obs}}$, we always have:

$$d(z, z') \geq d(z, S_{\text{obs}}) := \inf_{s \in S_{\text{obs}}} d(z, s).$$

Using this inequality, we lower bound the Wasserstein cost:

$$\begin{aligned} W_1(P_{\text{cf}}, P_{\text{PW}}) &= \int_{\mathcal{Z} \times \mathcal{Z}} d(z, z') \, \mathrm{d}\pi^\star(z, z') \\ &\geq \int_{A \times S_{\text{obs}}} d(z, z') \, \mathrm{d}\pi^\star(z, z') \\ &\geq \int_{A \times S_{\text{obs}}} d(z, S_{\text{obs}}) \, \mathrm{d}\pi^\star(z, z') \\ &= \int_A d(z, S_{\text{obs}}) \, \mathrm{d}P_{\text{cf}}(z) = \rho. \end{aligned}$$

*Key point (marginal consistency):* Since $\pi^\star$ is a coupling between $P_{\text{cf}}$ and $P_{\text{PW}}$, its left marginal is $P_{\text{cf}}$. Thus,

$$\int_{A \times S_{\text{obs}}} f(z) \, \mathrm{d}\pi^\star(z, z') = \int_A f(z) \, \mathrm{d}P_{\text{cf}}(z),$$

for any integrable function $f(z)$, including $f(z) = d(z, S_{\text{obs}})$.

**Step 3: Combine inequalities.** We now have:

$$W_1(P_{\text{cf}}, P_{\text{pcf}}) \leq \rho \leq W_1(P_{\text{cf}}, P_{\text{PW}}).$$

**Step 4: When is the inequality strict?** Equality would require that *every* point $y \in S_{\text{obs}}$ receives *exactly* the extra mass $(\lambda^{-1} - 1) P_{\text{cf}}(\mathrm{d}y)$ from the set $\{z \in A : m(z) = y\}$. Unless the geometry of $A$ and the projection map $m$ conspires to make this identity hold $P_{\text{cf}}$–a.s., extra intra-support rearrangement is necessary, adding positive cost and making the inequality strict. Such perfect alignment occurs only in contrived edge cases, so in practice we have $W_1(P_{\text{pcf}}, P_{\text{cf}}) < W_1(P_{\text{PW}}, P_{\text{cf}})$.

□

**Remark A.1** (Generalization Implication)**.** *While 1-Wasserstein proximity does not guarantee a strictly lower test error, it bounds the difference in risk under standard Lipschitz conditions. Specifically, if the loss function $\mathcal{L}$ is $L_\ell$-Lipschitz in its first argument and the model $f$ is $L_f$-Lipschitz, then (Fournier and Guillin 2015)*

$$|\mathcal{R}_{P_{\text{cf}}}(f) - \mathcal{R}_{P_{\text{train}}}(f)| \leq L_\ell L_f \cdot W_1(P_{\text{cf}}, P_{\text{train}}).$$

*Therefore, proxy matching induces a strictly tighter upper bound on the generalization error than permutation weighting, assuming comparable training loss. This highlights the distributional advantage of matching-based proxy construction.*

## B Additional Experimental Details

### B.1 Dataset Visualization

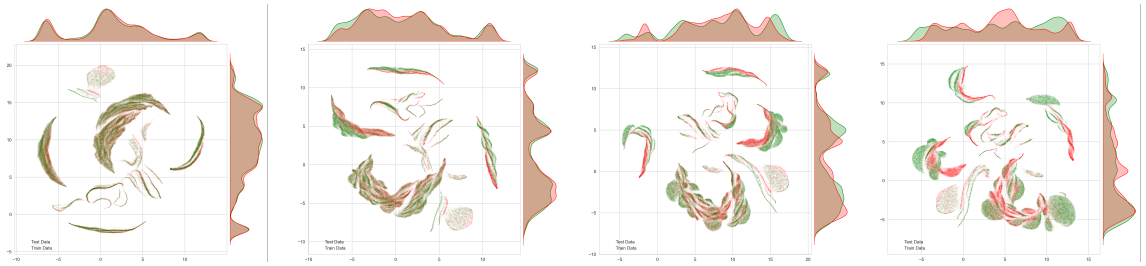
To examine the structure and coverage of our datasets, we visualize training and test sets using UMAP projections over the joint input space $(\mathbf{x}, \mathbf{t})$. As UMAP is a nonlinear projection, local geometric distortion may occur; the plots are intended for qualitative comparison only.

- **Synthetic datasets:** We visualize 12 settings covering all combinations of intervention dimensionality $p \in \{10, 20, 30\}$ and confounding strength $\gamma \in \{0.1, 0.3, 0.5, 1.0\}$. Each subplot shows a 2D UMAP projection, with training samples in green and test samples in red. Representative examples are shown in Figure 7.

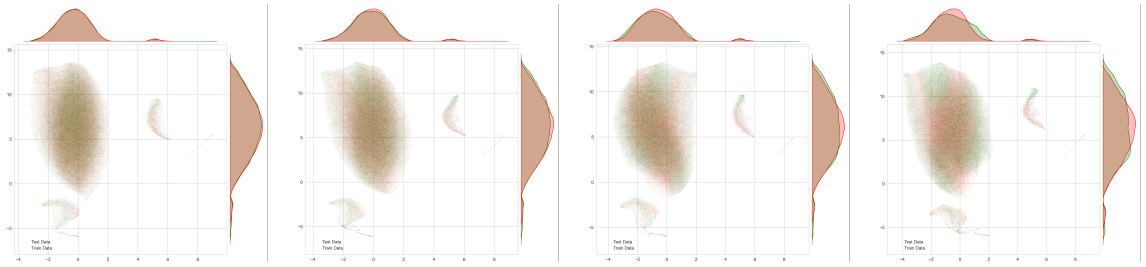
  Although fully synthetic, these datasets are generated via a structured process that induces long-tailed distributions, nonlinear manifolds, and multi-scale clustering. The resulting input space exhibits rich and heterogeneous geometry—resembling real-world user behavior patterns.

  As shown in Figure 7, increasing $\gamma$ leads to stronger covariate-treatment dependencies and more pronounced support mismatch between training and test sets. While the structure also varies across different values of $p$, no consistent trend is observed, likely due to randomness in data generation.

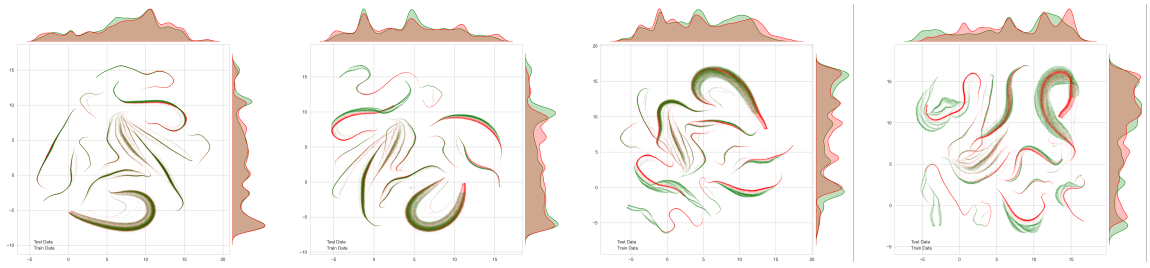
- **Semi-synthetic dataset:** Figure 8 shows the UMAP projection of the semi-synthetic dataset constructed from real-world logs. While the projection reveals some structural heterogeneity, it does not clearly reflect the extent of support mismatch between training and test sets. To quantify this, we compare the number of distinct intervention combinations: the training set contains approximately 200,000 unique combinations, the test set contains around 400,000, but their overlap includes only about 30,000 shared combinations. These statistics highlight the severe sparsity and shift in real-world combinatorial settings, even under semi-synthetic construction.



(a) $p = 10, \gamma = 0.1$ (b) $p = 10, \gamma = 0.3$ (c) $p = 10, \gamma = 0.5$ (d) $p = 10, \gamma = 1.0$



(e) $p = 20, \gamma = 0.1$ (f) $p = 20, \gamma = 0.3$ (g) $p = 20, \gamma = 0.5$ (h) $p = 20, \gamma = 1.0$



(i) $p = 30, \gamma = 0.1$ (j) $p = 30, \gamma = 0.3$ (k) $p = 30, \gamma = 0.5$ (l) $p = 30, \gamma = 1.0$

Figure 7: UMAP projections of 12 synthetic datasets under different treatment dimensionalities $p$ and confounding strengths $\gamma$. Training samples are shown in green and test samples in red.

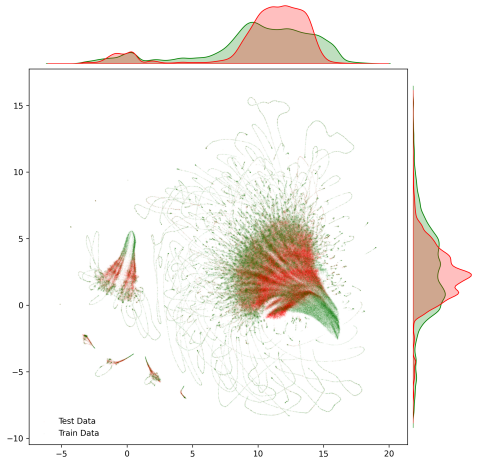



Figure 8: UMAP projection of the semi-synthetic dataset. Training samples are shown in green and test samples in red.

### B.2 Implementation Details for Baselines

**Training Setup.** All baseline models are implemented within a unified PyTorch framework with consistent training configurations unless otherwise specified. We search learning rates over $\{1 \times 10^{-3}, 1 \times 10^{-4}\}$, and select $1 \times 10^{-4}$ for synthetic experiments and $1 \times 10^{-3}$ for semi-synthetic experiments, based on convergence speed and validation performance. The batch size is set to 1024 for synthetic experiments and 2048 for semi-synthetic ones. All models are trained for up to 50 epochs using the Adam optimizer, with early stopping based on validation loss. Unless noted otherwise, all predictors adopt a 4-layer MLP with 128 hidden units per layer and ReLU activations. For the DSCF framework, we search the number of experts in $\{3, 5\}$, and use 5 in the final experiments due to better stability and performance. For baseline methods, key hyperparameters such as the regularization coefficient in VSR and the smoothing factor in PW are selected based on validation performance. All experiments are conducted on GPUs with at least 16 GB of memory. Synthetic experiments are run on a Windows 11 system with an NVIDIA RTX 5080 GPU (16 GB VRAM). Semi-synthetic experiments are conducted on an Ubuntu system with an NVIDIA A100 GPU (80 GB VRAM). Key software libraries include PyTorch, NumPy, and FAISS.

**Method-Specific Configurations.**

- **RMNet** follows the original implementation with an IPM-based regularization loss. The regularization distribution is chosen as the same independent surrogate used in our main model. Since the original paper does not specify the regularization strength, we set it to $1 \times 10^{-3}$, following common practice in CFRNet (Shalit, Johansson, and Sontag 2017).
- **PW** is known to suffer from high variance under positivity violations (Cortes, Mansour, and Mohri 2010). To mitigate instability, we truncate sample weights at the 99th percentile of the estimated importance weights computed over the training set, and normalize them across the full dataset.
- **VSR** uses a VAE to encode interventions into latent representations regularized toward a standard Gaussian prior. In our experiments, we find that when data complexity is low (especially for binary variables), the model tends to produce near-Gaussian latent representations, making it difficult for the domain classifier to distinguish between observational and prior samples, and ultimately preventing effective reweighting. To alleviate this, we reduce the prior regularization strength to 0.1 based on empirical observations.
- **$\text{NN}_{\text{pcf}}$** is trained on proxy counterfactual samples constructed via approximate matching. To prevent over-representation of a narrow subset of samples, we adopt a hybrid retrieval strategy: for 90% of the queries, we select the nearest neighbor deterministically; for the remaining 10%, we sample randomly from the top 0.32% of closest points. This heuristic balances retrieval accuracy and sample diversity (Wu et al. 2023).
- **H-Learner** uses a meta-learning architecture with a two-stage structure. The hypernetwork takes the treatment vector as input and generates the weights of an outcome predictor, which then maps covariates to outcomes. The hidden dimension of the hypernetwork is set to $4p$ to ensure sufficient capacity. This design results in a parameter count roughly $4p$ times larger than standard models, making H-Learner considerably more resource-intensive.
- **Synthetic Combinations (SC)** assumes access to outcomes under multiple interventions for the same unit. This assumption is unrealistic in real-world scenarios where units differ in covariates and IDs, and it also contradicts the definition of counterfactuals. To emulate this setting, we perform $k$-means clustering over the covariate space and treat each cluster as a pseudo-unit. The number of clusters is set to $20p$ to ensure sufficient within-unit intervention diversity, thereby providing enough pseudo-unit to support horizontal regression as required by SC. Despite these adjustments, SC still fails to adapt to our setting, and performs poorly in our synthetic experiments (see Table 4).

Table 4: Prediction errors (RMSE and MAE) of SC across synthetic datasets with varying $p$ and confounding strengths $\gamma$.

| SC | $\gamma = 0.1$ | | $\gamma = 0.3$ | | $\gamma = 0.5$ | | $\gamma = 1.0$ | |
|---|---|---|---|---|---|---|---|---|
| | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE |
| $p = 10$ | 6.104 | 3.501 | 7.636 | 4.236 | 7.439 | 3.928 | 8.035 | 3.884 |
| $p = 20$ | 6.356 | 4.462 | 35.73 | 3.752 | 8.559 | 3.713 | 101.9 | 4.680 |
| $p = 30$ | 8.364 | 5.851 | 12.90 | 8.830 | 18.10 | 12.65 | 21.58 | 15.57 |

Table 5: Ablation configurations in Table 3. Full results for each configuration (without averaging) are provided in Table 6.

| id | Reg Data | | Reg Model | Cls Data | | | Output |
|---|---|---|---|---|---|---|---|
| | $D_{obs}$ | $D_{pcf}$ | | $D_{obs}$ | $D_{pcf}$ | $D_{cf}$ | |
| (1) | ✓ | | MLP | | | | |
| (2) | | ✓ | MLP | | | | |
| (3) | ✓ | ✓ | MLP×2 | ✓ | ✓ | | Reweighted |
| (4) | ✓ | ✓ | HardShare | | | | Obs Head |
| (5) | ✓ | ✓ | HardShare | | | | Pcf Head |
| (6) | ✓ | | MMoE | | | | |
| (7) | | ✓ | MMoE | | | | |
| (8) | ✓ | ✓ | MMoE | | | | Obs Head |
| (9) | ✓ | ✓ | MMoE | | | | Pcf Head |
| (10) | ✓ | ✓ | MMoE | | | | avg |
| (11) | ✓ | ✓ | MMoE | ✓ | | ✓ | Reweighted |
| (12) | ✓ | ✓ | MMoE-lite | ✓ | ✓ | | Reweighted |
| (13) | ✓ | ✓ | MMoE | ✓ | ✓ | | Reweighted |

Table 6: Full ablation results across varying confounding strengths $\gamma$ and intervention dimensions $p$

| id | $\gamma = 0.1$ | | $\gamma = 0.3$ | | $\gamma = 0.5$ | | $\gamma = 1.0$ | |
|---|---|---|---|---|---|---|---|---|
| | RMSE ± $\sigma$ | MAE ± $\sigma$ | RMSE ± $\sigma$ | MAE ± $\sigma$ | RMSE ± $\sigma$ | MAE ± $\sigma$ | RMSE ± $\sigma$ | MAE ± $\sigma$ |
| $p = 10$ | | | | | | | | |
| (1) | $3.952_{\pm 0.160}$ | $1.434_{\pm 0.040}$ | $5.499_{\pm 0.013}$ | $1.834_{\pm 0.008}$ | $6.164_{\pm 0.057}$ | $2.197_{\pm 0.035}$ | $6.265_{\pm 0.042}$ | $2.574_{\pm 0.010}$ |
| (2) | $3.776_{\pm 0.172}$ | $1.385_{\pm 0.037}$ | $5.330_{\pm 0.020}$ | $1.632_{\pm 0.032}$ | $5.682_{\pm 0.022}$ | $1.776_{\pm 0.003}$ | $6.130_{\pm 0.055}$ | $2.109_{\pm 0.007}$ |
| (3) | $3.761_{\pm 0.131}$ | $1.229_{\pm 0.024}$ | $5.329_{\pm 0.014}$ | $1.554_{\pm 0.018}$ | $5.790_{\pm 0.017}$ | $1.771_{\pm 0.011}$ | $6.082_{\pm 0.036}$ | $2.097_{\pm 0.001}$ |
| (4) | $3.575_{\pm 0.067}$ | $1.229_{\pm 0.024}$ | $5.424_{\pm 0.098}$ | $1.554_{\pm 0.030}$ | $5.716_{\pm 0.108}$ | $1.785_{\pm 0.047}$ | $6.011_{\pm 0.076}$ | $2.190_{\pm 0.011}$ |
| (5) | $3.543_{\pm 0.067}$ | $1.211_{\pm 0.017}$ | $5.388_{\pm 0.093}$ | $1.592_{\pm 0.018}$ | $5.665_{\pm 0.112}$ | $1.735_{\pm 0.045}$ | $5.992_{\pm 0.078}$ | $2.132_{\pm 0.006}$ |
| (6) | $3.200_{\pm 0.091}$ | $0.991_{\pm 0.028}$ | $4.615_{\pm 0.022}$ | $1.338_{\pm 0.062}$ | $5.234_{\pm 0.232}$ | $1.577_{\pm 0.010}$ | $5.861_{\pm 0.072}$ | $2.179_{\pm 0.026}$ |
| (7) | $3.309_{\pm 0.164}$ | $1.014_{\pm 0.025}$ | $4.796_{\pm 0.167}$ | $1.253_{\pm 0.044}$ | $5.162_{\pm 0.149}$ | $1.410_{\pm 0.034}$ | $5.314_{\pm 0.031}$ | $1.769_{\pm 0.035}$ |
| (8) | $3.100_{\pm 0.045}$ | $0.904_{\pm 0.047}$ | $4.493_{\pm 0.123}$ | $1.180_{\pm 0.037}$ | $5.192_{\pm 0.140}$ | $1.465_{\pm 0.040}$ | $5.258_{\pm 0.055}$ | $1.871_{\pm 0.085}$ |
| (9) | $3.086_{\pm 0.151}$ | $0.867_{\pm 0.011}$ | $4.495_{\pm 0.240}$ | $1.152_{\pm 0.034}$ | $5.102_{\pm 0.256}$ | $1.363_{\pm 0.056}$ | $5.280_{\pm 0.126}$ | $1.654_{\pm 0.031}$ |
| (10) | $3.107_{\pm 0.021}$ | $0.833_{\pm 0.020}$ | $4.463_{\pm 0.139}$ | $1.131_{\pm 0.003}$ | $5.119_{\pm 0.185}$ | $1.371_{\pm 0.043}$ | $5.221_{\pm 0.077}$ | $1.715_{\pm 0.062}$ |
| (11) | $3.070_{\pm 0.136}$ | $0.831_{\pm 0.010}$ | $4.495_{\pm 0.238}$ | $1.135_{\pm 0.019}$ | $5.106_{\pm 0.251}$ | $1.351_{\pm 0.056}$ | $5.276_{\pm 0.124}$ | $1.647_{\pm 0.033}$ |
| (12) | $3.295_{\pm 0.045}$ | $0.878_{\pm 0.006}$ | $4.705_{\pm 0.202}$ | $1.194_{\pm 0.023}$ | $5.014_{\pm 0.257}$ | $1.319_{\pm 0.061}$ | $5.388_{\pm 0.018}$ | $1.740_{\pm 0.089}$ |
| (13) | $3.100_{\pm 0.026}$ | $0.832_{\pm 0.017}$ | $4.469_{\pm 0.152}$ | $1.132_{\pm 0.003}$ | $5.112_{\pm 0.200}$ | $1.350_{\pm 0.048}$ | $5.207_{\pm 0.066}$ | $1.640_{\pm 0.042}$ |
| $p = 20$ | | | | | | | | |
| (1) | $4.459_{\pm 0.094}$ | $2.081_{\pm 0.037}$ | $6.423_{\pm 0.277}$ | $2.476_{\pm 0.079}$ | $5.588_{\pm 0.067}$ | $2.568_{\pm 0.074}$ | $7.849_{\pm 0.168}$ | $3.494_{\pm 0.008}$ |
| (2) | $4.580_{\pm 0.057}$ | $2.036_{\pm 0.031}$ | $6.178_{\pm 0.044}$ | $2.134_{\pm 0.018}$ | $5.642_{\pm 0.101}$ | $2.482_{\pm 0.109}$ | $7.393_{\pm 0.216}$ | $3.083_{\pm 0.061}$ |
| (3) | $4.280_{\pm 0.061}$ | $1.756_{\pm 0.021}$ | $6.125_{\pm 0.125}$ | $2.010_{\pm 0.024}$ | $5.507_{\pm 0.084}$ | $2.383_{\pm 0.088}$ | $7.362_{\pm 0.130}$ | $3.066_{\pm 0.058}$ |
| (4) | $4.611_{\pm 0.114}$ | $1.876_{\pm 0.166}$ | $6.150_{\pm 0.097}$ | $2.093_{\pm 0.038}$ | $5.652_{\pm 0.059}$ | $2.425_{\pm 0.074}$ | $7.591_{\pm 0.293}$ | $2.912_{\pm 0.049}$ |
| (5) | $4.587_{\pm 0.033}$ | $1.826_{\pm 0.019}$ | $6.142_{\pm 0.108}$ | $2.104_{\pm 0.034}$ | $5.623_{\pm 0.065}$ | $2.389_{\pm 0.057}$ | $7.685_{\pm 0.287}$ | $2.952_{\pm 0.067}$ |
| (6) | $4.480_{\pm 0.334}$ | $1.577_{\pm 0.210}$ | $5.764_{\pm 0.223}$ | $1.782_{\pm 0.065}$ | $4.920_{\pm 0.399}$ | $1.926_{\pm 0.136}$ | $6.895_{\pm 0.141}$ | $2.794_{\pm 0.100}$ |
| (7) | $3.865_{\pm 0.363}$ | $1.429_{\pm 0.030}$ | $5.567_{\pm 0.120}$ | $1.603_{\pm 0.083}$ | $5.212_{\pm 0.138}$ | $1.941_{\pm 0.100}$ | $6.905_{\pm 0.120}$ | $2.452_{\pm 0.051}$ |
| (8) | $3.856_{\pm 0.243}$ | $1.363_{\pm 0.077}$ | $5.497_{\pm 0.277}$ | $1.458_{\pm 0.148}$ | $4.737_{\pm 0.170}$ | $1.799_{\pm 0.067}$ | $6.434_{\pm 0.253}$ | $2.253_{\pm 0.102}$ |
| (9) | $3.752_{\pm 0.034}$ | $1.323_{\pm 0.093}$ | $5.513_{\pm 0.202}$ | $1.447_{\pm 0.106}$ | $4.747_{\pm 0.133}$ | $1.700_{\pm 0.063}$ | $6.768_{\pm 0.113}$ | $2.211_{\pm 0.195}$ |
| (10) | $3.714_{\pm 0.137}$ | $1.233_{\pm 0.056}$ | $5.458_{\pm 0.182}$ | $1.377_{\pm 0.131}$ | $4.649_{\pm 0.074}$ | $1.649_{\pm 0.056}$ | $6.521_{\pm 0.155}$ | $2.132_{\pm 0.151}$ |
| (11) | $3.753_{\pm 0.103}$ | $1.255_{\pm 0.063}$ | $5.523_{\pm 0.202}$ | $1.422_{\pm 0.114}$ | $4.710_{\pm 0.083}$ | $1.687_{\pm 0.059}$ | $6.610_{\pm 0.224}$ | $2.206_{\pm 0.195}$ |
| (12) | $3.730_{\pm 0.129}$ | $1.075_{\pm 0.113}$ | $5.265_{\pm 0.100}$ | $1.328_{\pm 0.073}$ | $4.447_{\pm 0.073}$ | $1.513_{\pm 0.015}$ | $6.302_{\pm 0.088}$ | $2.001_{\pm 0.022}$ |
| (13) | $3.725_{\pm 0.153}$ | $1.234_{\pm 0.058}$ | $5.462_{\pm 0.160}$ | $1.401_{\pm 0.118}$ | $4.673_{\pm 0.105}$ | $1.674_{\pm 0.055}$ | $6.640_{\pm 0.185}$ | $2.202_{\pm 0.193}$ |
| $p = 30$ | | | | | | | | |
| (1) | $1.742_{\pm 0.019}$ | $1.273_{\pm 0.015}$ | $4.295_{\pm 0.087}$ | $2.713_{\pm 0.041}$ | $9.196_{\pm 0.047}$ | $5.436_{\pm 0.028}$ | $11.751_{\pm 0.117}$ | $7.532_{\pm 0.065}$ |
| (2) | $1.712_{\pm 0.013}$ | $1.258_{\pm 0.009}$ | $2.956_{\pm 0.007}$ | $1.985_{\pm 0.020}$ | $6.466_{\pm 0.031}$ | $4.108_{\pm 0.036}$ | $9.385_{\pm 0.014}$ | $6.002_{\pm 0.021}$ |
| (3) | $1.527_{\pm 0.013}$ | $1.078_{\pm 0.003}$ | $2.931_{\pm 0.007}$ | $1.943_{\pm 0.019}$ | $6.463_{\pm 0.031}$ | $4.098_{\pm 0.036}$ | $9.384_{\pm 0.014}$ | $6.001_{\pm 0.021}$ |
| (4) | $1.364_{\pm 0.024}$ | $1.007_{\pm 0.018}$ | $2.711_{\pm 0.036}$ | $1.826_{\pm 0.021}$ | $7.349_{\pm 0.195}$ | $4.228_{\pm 0.123}$ | $9.461_{\pm 0.099}$ | $6.001_{\pm 0.045}$ |
| (5) | $1.355_{\pm 0.016}$ | $0.998_{\pm 0.012}$ | $2.611_{\pm 0.031}$ | $1.762_{\pm 0.019}$ | $7.341_{\pm 0.199}$ | $4.186_{\pm 0.123}$ | $9.534_{\pm 0.085}$ | $6.003_{\pm 0.036}$ |
| (6) | $1.285_{\pm 0.086}$ | $0.898_{\pm 0.047}$ | $3.434_{\pm 0.136}$ | $2.055_{\pm 0.051}$ | $6.931_{\pm 0.235}$ | $3.922_{\pm 0.137}$ | $9.791_{\pm 0.908}$ | $6.090_{\pm 0.321}$ |
| (7) | $1.171_{\pm 0.042}$ | $0.848_{\pm 0.046}$ | $3.056_{\pm 0.293}$ | $1.820_{\pm 0.173}$ | $5.920_{\pm 0.708}$ | $3.409_{\pm 0.316}$ | $8.767_{\pm 1.184}$ | $5.412_{\pm 0.529}$ |
| (8) | $1.019_{\pm 0.053}$ | $0.728_{\pm 0.024}$ | $3.172_{\pm 0.903}$ | $1.828_{\pm 0.379}$ | $5.712_{\pm 0.457}$ | $3.252_{\pm 0.252}$ | $7.695_{\pm 0.363}$ | $4.855_{\pm 0.170}$ |
| (9) | $0.992_{\pm 0.050}$ | $0.713_{\pm 0.026}$ | $2.458_{\pm 0.265}$ | $1.518_{\pm 0.130}$ | $5.070_{\pm 0.242}$ | $2.919_{\pm 0.068}$ | $6.823_{\pm 0.218}$ | $4.389_{\pm 0.126}$ |
| (10) | $0.985_{\pm 0.044}$ | $0.705_{\pm 0.019}$ | $2.669_{\pm 0.429}$ | $1.609_{\pm 0.193}$ | $5.246_{\pm 0.103}$ | $3.018_{\pm 0.082}$ | $7.001_{\pm 0.075}$ | $4.489_{\pm 0.049}$ |
| (11) | $0.986_{\pm 0.047}$ | $0.707_{\pm 0.022}$ | $2.458_{\pm 0.265}$ | $1.517_{\pm 0.130}$ | $5.070_{\pm 0.242}$ | $2.919_{\pm 0.068}$ | $6.823_{\pm 0.218}$ | $4.389_{\pm 0.126}$ |
| (12) | $0.967_{\pm 0.014}$ | $0.692_{\pm 0.001}$ | $2.983_{\pm 0.356}$ | $1.685_{\pm 0.090}$ | $6.831_{\pm 0.331}$ | $3.693_{\pm 0.168}$ | $8.453_{\pm 0.322}$ | $5.334_{\pm 0.236}$ |
| (13) | $0.983_{\pm 0.045}$ | $0.705_{\pm 0.021}$ | $2.458_{\pm 0.265}$ | $1.517_{\pm 0.130}$ | $5.070_{\pm 0.242}$ | $2.919_{\pm 0.068}$ | $6.823_{\pm 0.218}$ | $4.389_{\pm 0.126}$ |

### B.3 Ablation Configuration Details

The configurations used in our ablation study are summarized in Table 5. Corresponding results on synthetic datasets are provided in Table 6. Below, we explain the meaning of each field in the configuration table:

- **Reg Data**: Indicates the data source used to train the regression model.

  - $\texttt{D}_{\texttt{obs}}$: Observational data.
  - $\texttt{D}_{\texttt{pcf}}$: Proxy counterfactual data constructed via approximate matching.
- **Reg Model**: Specifies the regression model architecture.
  - `MLP`: A standard multi-layer perceptron trained on a single data source.
  - `MLP`$\times 2$: Two separate MLPs trained independently on $\texttt{D}_{\texttt{obs}}$ and $\texttt{D}_{\texttt{pcf}}$, respectively.
  - `HardShare`: A shared-bottom architecture jointly trained on both data sources.
  - `MMoE`: A multi-gate mixture-of-experts model with separate gating for each data source.
  - `MMoE-lite`: A reduced-capacity MMoE variant with hidden dimensions halved.
- **Cls Data**: Indicates whether a domain classifier is used to compute soft weights for prediction fusion, and the data sources used to train it.
  - (empty): No domain classifier is used; final predictions do not rely on reweighting.
  - $\texttt{D}_{\texttt{obs}}$: Observational data (same as in Reg Data).
  - $\texttt{D}_{\texttt{pcf}}$: Proxy counterfactual data (same as in Reg Data).
  - $\texttt{D}_{\texttt{cf}}$: Synthetic counterfactual data sampled from the factorized target distribution.
- **Output**: Indicates how the final prediction is computed, either directly or via output fusion.
  - (empty): The final prediction is directly taken from the regression model without any fusion.
  - `Reweighted`: Weighted average of the two heads using soft weights from the domain classifier.
  - `Obs Head`: Output from the observational head only.
  - `Pcf Head`: Output from the proxy counterfactual head only.
  - `Avg`: Equal-weight average of the two prediction heads.