PARAMETER-EFFICIENT DISTRIBUTIONALLY ROBUST ADAPTATION OF TABULAR FOUNDATION MODELS UNDER SUBPOPULATION SHIFT

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite strong mean accuracy, tabular foundation models (TFMs) can perform poorly on underrepresented groups under subpopulation shift, where group proportions change between training and deployment. We propose DR-TFM, a parameter-efficient distributionally robust adaptation framework that requires no true group annotations. DR-TFM adjusts attention to labeled context examples by fine-tuning an existing query scaling network or adding and training one, while keeping all other parameters fixed. We instantiate the framework with two robust objectives using estimated groups or source conditional distributions derived from training data. For TabPFN-3, adaptation updates only 0.016% of the pretrained model's parameters. Across five tabular benchmarks, DR-TFM achieves substantially higher average worst-group accuracy than pretrained TFMs and the compared robust baselines without true group annotations, while maintaining competitive mean group accuracy. DR-TFM also improves average worst-group accuracy on ACS Income and across four additional TFMs.

1 INTRODUCTION

Tabular foundation models (TFMs), such as the TabPFN[1], have emerged as a promising approach to general-purpose prediction on tabular data (Hollmann et al., 2025; Qu et al., 2025; Jiang et al., 2026). They have demonstrated strong predictive performance, while recent advances have further improved their scalability and computational efficiency (Grinsztajn et al., 2026).

However, strong predictive performance does not necessarily imply robustness to distribution shift, where the training and test distributions differ. We focus on subpopulation shift, a practically important form of distribution shift in which the training and test distributions consist of the same underlying subpopulations, which we also refer to as groups, but differ in their mixture proportions (Duchi & Namkoong, 2021; Sagawa et al., 2020; Yang et al., 2023). For example, the relative frequencies of demographic groups may vary across data collection sites or deployment environments. When some groups are underrepresented in the training data, predictive methods optimized for average performance can exhibit substantial variation in performance across groups, with particularly poor performance on underrepresented groups. As the training mixture becomes more imbalanced, these disparities can become more pronounced, while aggregate performance may obscure degradation in the worst-performing group.

Figure 1 illustrates this challenge. It shows TabPFN's mean group and worst-group accuracies as the combined proportion of the majority groups in the training data increases, while test groups remain balanced. When the majority groups together account for 90% of the training data, the worst-group accuracy falls to approximately 31%, despite a mean group accuracy of approximately 66%.

To address this challenge, we propose Distributionally Robust Tabular Foundation Models (DR-TFM), a framework that adapts TFMs to subpopulation shift by updating only a small set of model parameters using distributionally robust optimization (DRO) objectives. We instantiate this framework with two objectives. DR-TFM (GR) uses GroupDRO (Sagawa et al., 2020). For DR-TFM

[1]Unless otherwise specified, TabPFN refers to version 3.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

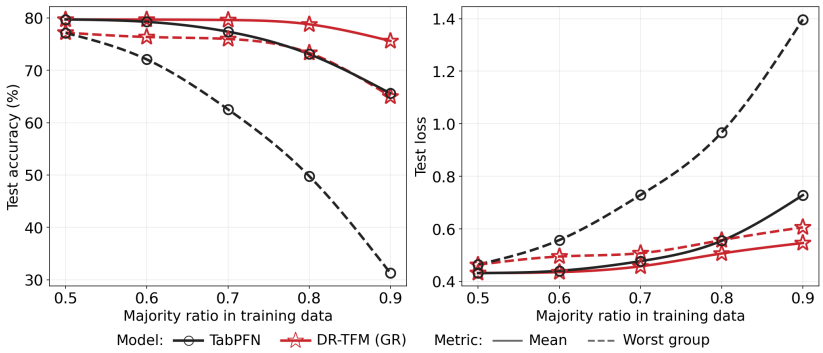

Figure 1: Classification results under subpopulation shift with balanced test groups. TabPFN's worst-group accuracy deteriorates more sharply than its mean group accuracy, whereas DR-TFM (GR) mitigates this degradation without true group labels. Curves show averages over 50 runs.

(MS), we adopt the multi-source DRO objective of Kim et al. (2026), motivated by the empirical gains reported on spurious-correlation benchmarks. Both objectives rely on auxiliary group or source structure, which we infer or construct from the training data. Specifically, we estimate groups by clustering pretrained TFM embeddings and form source subsets through clustering or resampling, enabling adaptation without true group annotations. To the best of our knowledge, DR-TFM is the first adaptation framework for TFMs that explicitly targets subpopulation shift.

The central idea of DR-TFM is to improve robustness by adapting how pretrained representations are used for prediction, rather than modifying the representations themselves. For classification, we implement this idea through query scaling: a lightweight neural network learns coordinate-wise rescalings of query vectors, thereby adjusting how each query attends to labeled context examples. Both DRO instantiations optimize this query scaling network while keeping all remaining model parameters fixed, concentrating robust adaptation on how the context contributes to predictions.

This targeted adaptation offers computational and memory efficiency, scalability, and applicability across TFMs. First, restricting adaptation to the scaling network reduces the computational and memory costs of parameter updates. For example, adapting TabPFN-3 updates only 8,320 parameters, approximately 0.016% of the model. Second, sharing the network across attention heads allows larger encoders and more heads without increasing the trainable parameter count, provided that the head dimension and scaling network architecture remain fixed. Third, the mechanism does not depend on query scaling being part of the pretrained architecture. We fine-tune the existing query scaling network in TabPFN's classification decoder; for TFMs without this component, we add and train one in the final attention layer that attends to the context.

We evaluate DR-TFM through simulations and experiments on five tabular benchmarks and ACS Income. Across these benchmarks, both instantiations achieve substantially higher average worst-group accuracy than pretrained TabPFN while maintaining competitive mean group accuracy. Both also require substantially less time than TabPFN fine-tuning. We further apply DR-TFM (GR) to four additional TFMs and observe consistent improvements in average worst-group accuracy.

Our main contributions can be summarized as follows.

- We propose DR-TFM, a parameter-efficient distributionally robust adaptation framework for TFMs under subpopulation shift, instantiated with two robust objectives without requiring true group annotations.
- We develop an efficient and scalable adaptation strategy that reweights attention through a small, shared query scaling network while keeping pretrained representations fixed.
- Through extensive simulations and real-world experiments, we demonstrate that both instantiations of DR-TFM improve robustness to subpopulation shift, achieving higher average worst-group accuracy than the compared baselines that do not use true group annotations, while maintaining competitive mean group accuracy.
- We demonstrate the applicability of our robust adaptation across multiple TFMs, with consistent improvements in average worst-group accuracy.

2 Related Work

Tabular foundation models and their adaptation. Tabular foundation models (TFMs) transfer knowledge across datasets through pretraining and in-context learning. TabPFN (Hollmann et al., 2025; Grinsztajn et al., 2026) and TabICL (Qu et al., 2025) are pretrained on synthetic datasets, whereas TabDPT (Ma et al., 2025) uses real tabular datasets.

TFM adaptation methods modify the context, model parameters, or prediction outputs. TuneTables (Feuer et al., 2024) compresses the training data into a compact learned context through prompt tuning. Thomas et al. (2024) propose TabPFN-kNN, which constructs query-specific contexts through nearest-neighbor retrieval, and LoCalPFN, which further combines this retrieval with task-specific fine-tuning. MixturePFN (Xu et al., 2025) clusters the training data and fine-tunes cluster-specific experts, while BETA (Liu & Ye, 2025) adapts lightweight input encoders while keeping the pretrained TabPFN transformer fixed. DistPFN (Lee, 2026) targets label shift through test-time posterior adjustment without updating model parameters. We address the distinct challenge of subpopulation shift through parameter-efficient decoder-side adaptation: we keep the pretrained representations fixed and update only a small query-scaling network that modifies how each query attends to labeled context examples.

Beyond generic adaptation, recent work has also examined fairness in TFMs. FairPFN (Robertson et al., 2025) incorporates causal fairness into pretraining using synthetically generated causal data, while Kenfack et al. (2026) study fairness in tabular in-context learning through context selection that either balances groups or ranks examples by the uncertainty in predicting sensitive attributes. These methods target fairness with respect to explicitly observed sensitive attributes, whereas we address robustness to changes in latent subpopulation proportions without access to true group annotations during adaptation or model selection.

Robust learning under subpopulation shift. A broad line of work seeks to improve robustness to subpopulation shifts, where changes in group proportions can disproportionately degrade performance on particular subpopulations (Yang et al., 2023). In settings where group annotations are available, GroupDRO (Sagawa et al., 2020) is a representative approach that minimizes the worst-group loss over predefined groups. The ambiguity set determines which distribution shifts are considered. Subsequent work has expanded these sets. Krueger et al. (2021) allow mixture weights outside the probability simplex, while Jo et al. (2026) introduce hierarchical ambiguity sets that account for both inter- and intra-group uncertainty.

Without group annotations, existing approaches instead rely on proxy signals or latent structure inferred from the data. JTT (Liu et al., 2021) upweights examples misclassified by an initial model. Among latent-group approaches, GEORGE (Sohoni et al., 2020) clusters learned representations within each class and applies GroupDRO to the resulting pseudo-groups. BPA (Seo et al., 2022) and SPARE (Yang et al., 2024) similarly reweight or resample groups inferred from learned representations or early-training outputs. In a related direction, Kim et al. (2026) construct pseudo-sources through subsampling and optimize over mixtures of their estimated conditional label distributions. Much of this literature has focused on vision-based benchmarks, while comparatively less attention has been given to tabular data. A recent tabular-specific approach is LSR (Tong et al., 2025), which reweights samples using score-based density estimates in a learned latent space. Unlike our approach, however, LSR learns its latent representation and predictor from the training data rather than building on a pretrained TFM. Despite growing work on both TFM adaptation and subpopulation robustness, their intersection remains largely unexplored. To our knowledge, no prior method explicitly adapts TabPFN to subpopulation shift without true group annotations.

3 Preliminaries

Setup and notation. Let $X \in \mathcal{X} \subseteq \mathbb{R}^d$ and $Y \in \mathcal{Y}$ denote the input and response variables. Let P^{tr} and P^{te} denote the training and test joint distributions of (X, Y), respectively. For any joint distribution P of (X, Y), let P_X denote its marginal distribution of X, $P_{Y|X}$ its conditional distribution of Y given X, and $\mathbb{E}_P[\cdot]$ the corresponding expectation. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ be the labeled training dataset and $\mathcal{D}'_X = \{x'_j\}_{j=1}^m$ be the unlabeled test dataset. For a parametrized

predictor f_θ, let $\ell(f_\theta(X), Y)$ denote its loss. For a positive integer K, let $[K] = \{1, \ldots, K\}$ and let Δ_{K-1} denote the $(K-1)$-dimensional probability simplex.

Subpopulation shift. We consider training and test distributions that share the same K subpopulation distributions $P_1, \ldots, P_K$ but differ in their mixture proportions. Specifically, $P^{\mathrm{tr}} = \sum_{k=1}^{K} \pi_k^{\mathrm{tr}} P_k$ and $P^{\mathrm{te}} = \sum_{k=1}^{K} \pi_k^{\mathrm{te}} P_k$, where $\pi^{\mathrm{tr}}, \pi^{\mathrm{te}} \in \Delta_{K-1}$ denote the training and test mixture weight vectors, respectively, with $\pi^{\mathrm{tr}} \neq \pi^{\mathrm{te}}$. Standard empirical risk minimization (ERM) does not explicitly account for such changes in mixture proportions and can therefore be vulnerable to subpopulation shift. A detailed theoretical discussion is provided in Appendix A.1.

Distributionally robust optimization. DRO is a promising approach to addressing such distributional shifts (Mohajerin Esfahani & Kuhn, 2018; Duchi & Namkoong, 2021). A standard DRO problem solves

$$\min_{\theta} \sup_{Q \in \mathcal{B}} \mathbb{E}_Q \left[\ell(f_\theta(X), Y) \right], \tag{1}$$

where $\mathcal{B}$ is an ambiguity set specifying the collection of distributions over which robustness is sought, often defined as a suitable neighborhood of the empirical distribution.

Fine-tuning TFM. Given the training dataset $\mathcal{D}$ and a test input x, a pretrained TFM with parameters θ_0 produces a prediction $f_{\theta_0}(x; \mathcal{D})$, where the labeled dataset $\mathcal{D}$ serves as the *context* and x serves as the *query*. A pretrained TFM can be further adapted to $\mathcal{D}$ through fine-tuning (Grinsztajn et al., 2026; Liu & Ye, 2025). For this purpose, we split $\mathcal{D}$ into a context set $\mathcal{D}^{(c)}$ and a loss-evaluation set $\mathcal{D}^{(e)}$. Let $n_c = |\mathcal{D}^{(c)}|$ and $n_e = |\mathcal{D}^{(e)}|$, and let $\widehat{P}^{(e)}$ denote the empirical distribution of $\mathcal{D}^{(e)}$. We partition the model parameters as $\theta = (\theta_F, \theta_L)$, where θ_F and θ_L denote the frozen and learnable parameters, respectively. Standard fine-tuning initializes both components from their corresponding pretrained values in θ_0, keeps θ_F fixed, and solves

$$\min_{\theta_L} \mathbb{E}_{\widehat{P}^{(e)}} \left[\ell\Big(f_\theta(X; \mathcal{D}^{(c)}), Y \Big) \right]. \tag{2}$$

4 Proposed Method

4.1 Distributionally Robust Formulation

General formulation. Combining the fine-tuning objective in (2) with the standard DRO formulation in (1), we formulate DR-TFM as

$$\min_{\theta_L} \sup_{Q \in \mathcal{B}} \mathbb{E}_Q \left[\ell\Big(f_\theta(X; \mathcal{D}^{(c)}), Y \Big) \right]. \tag{3}$$

A key ingredient is the choice of an ambiguity set $\mathcal{B}$ that captures uncertainty arising from subpopulation shift. Drawing on established formulations in distributionally robust learning, we consider two ambiguity-set constructions tailored to subpopulation shift. Both rely on auxiliary structure, such as group assignments or source subsets, but this structure is derived from the observed training data rather than supplied as group annotations. Consequently, DR-TFM does not require true group labels for training. We present the key ideas below; detailed optimization formulations and algorithms are provided in Appendix A.2.

DR-TFM (GR). To motivate the construction, suppose first that each observation i in the training dataset $\mathcal{D}$ is associated with a group index $g_i \in [G]$. Following the GroupDRO framework (Sagawa et al., 2020), we instantiate (3) with the ambiguity set

$$\mathcal{B}_{\mathrm{GR}} = \left\{ Q = \sum_{k=1}^{G} q_k \widehat{P}_k^{(e)} \ : \ q \in \Delta_{G-1} \right\}, \tag{4}$$

where $\widehat{P}_k^{(e)}$ is the empirical distribution of the samples $(x_i, y_i) \in \mathcal{D}^{(e)}$ with $g_i = k$. In practice, however, we do not assume that these group indices are observed. Instead, we estimate them from pretrained TFM representations. Specifically, we first obtain an embedding from the frozen pretrained TFM for each training observation and then cluster the embeddings within each response

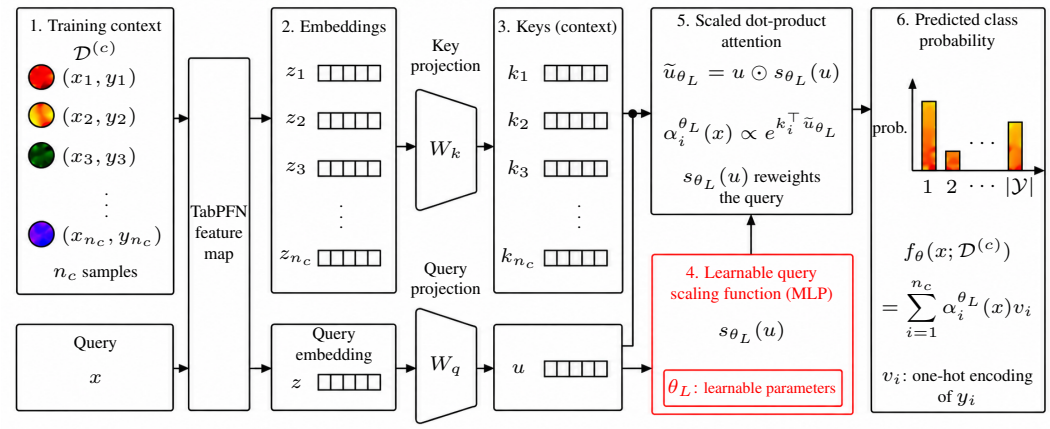

Figure 2: Query scaling for classification during adaptation. A query is predicted from the labeled training context. The scaling function s_{θ_L} adjusts the attention weights used to aggregate context labels. Only θ_L is updated; θ_F remains fixed.

class using a simple method such as k-means or a Gaussian mixture model. The resulting class–cluster pairs define the estimated group indices. In (4), we use these estimated assignments in place of the unobserved g_i.

DR-TFM (MS). Motivated by the strong empirical performance of the multi-source DRO formulation of Kim et al. (2026) under subpopulation shift, we construct an ambiguity set from multiple source-conditioned predictive distributions. We first form source subsets $\mathcal{S}_1, \ldots, \mathcal{S}_M \subseteq \mathcal{D}^{(c)}$ using either resampling or clustering. Using each $\mathcal{S}_k$ as context, the pretrained TFM with fixed parameters θ_0 defines a source-conditioned predictive distribution $\widehat{P}^{(k)}_{Y|X}(\cdot \mid x) = f_{\theta_0}(x; \mathcal{S}_k)$. For radii $\epsilon_1, \epsilon_2 \geq 0$, we define

$$\mathcal{B}_{\mathrm{MS}} = \left\{ Q = Q_X Q_{Y|X} : \begin{array}{ll} Q_{Y|X} = \sum_{k=1}^{M} \beta_k \widehat{P}^{(k)}_{Y|X}, & \beta \in \Delta_{M-1}, \\ W_\infty(Q_X, \widehat{P}^{(e)}_X) \leq \epsilon_1, & \|\beta - \bar{\beta}\|_2 \leq \epsilon_2 \end{array} \right\}, \tag{5}$$

where W_∞ denotes the infinite-order Wasserstein distance and $\bar{\beta}$ is the uniform probability vector. The radii ϵ_1 and ϵ_2 control the uncertainty in the input marginal and the source-mixture weights, respectively.

4.2 Parameter-Efficient Adaptation

We now describe a parameter-efficient adaptation procedure for both DRO instantiations defined by (4) and (5), focusing on TabPFN. We optimize the objective in (3) by fine-tuning only the *query scaling function* s_{θ_L}, parameterized by a small neural network within the classification decoder (Figure 2). The function produces scaling factors that multiply the query vector element-wise, thereby reweighting attention to context examples.

The key idea is to improve robustness by adapting how pretrained representations are used for prediction while preserving the representations themselves. TabPFN's pretrained embeddings have been shown to capture useful latent structure in tabular data (Ye et al., 2025; Grinsztajn et al., 2026). However, these informative embeddings alone do not ensure accurate predictions for underrepresented subpopulations. We implement this idea through query scaling, which adjusts how each query attends to labeled context examples. We fine-tune only the scaling function under either DRO objective while keeping all remaining model parameters fixed, concentrating adaptation on how the context contributes to predictions.

This adaptation strategy offers three advantages: computational and memory efficiency, scalability, and applicability across TFMs. First, we update only 8,320 parameters (0.016% of TabPFN), compared with nearly all parameters in full-model fine-tuning. This effectively reduces the computational and memory costs of parameter updates. Second, the query scaling network is shared across

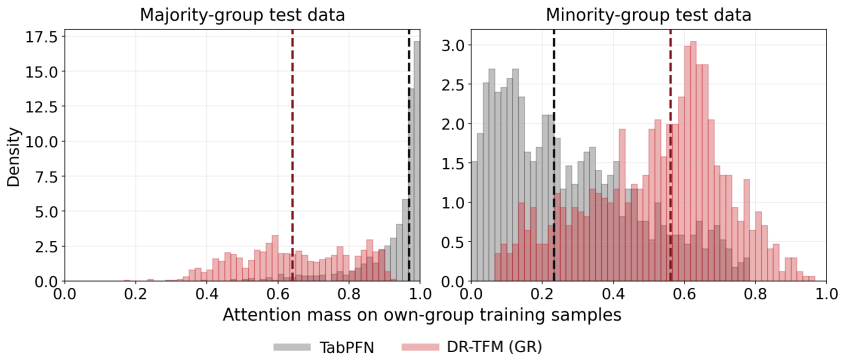

Figure 3: Histograms of the sum of attention weights assigned to each test sample's own true group, for majority (left) and minority (right) test samples. Gray and red denote TabPFN and DR-TFM (GR), respectively; dashed lines mark medians. For adaptation, DR-TFM estimates groups using a Gaussian mixture model (GMM).

attention heads. The number of trainable parameters does not increase with encoder size or the number of heads, provided that the head dimension and scaling network architecture remain fixed. Third, the same mechanism can be applied to other TFMs: we fine-tune an existing query scaling network when available, or add and train one to scale queries before attention weights are computed.

We next describe how query scaling reweights attention to context examples. Specifically, we consider a single query x from a sample $(x, y) \in \mathcal{D}^{(e)}$, given the context set $\mathcal{D}^{(c)}$. Let z denote the query embedding and $z_1, \ldots, z_{n_c}$ the context embeddings, all obtained from the pretrained TabPFN feature map. For a single attention head, the fixed projections W_k and W_q map these embeddings to context keys $k_i \in \mathbb{R}^r$ and a query vector $u \in \mathbb{R}^r$, where r is the dimension of the key and query vectors in each attention head. Query scaling is applied as follows:

$$k_i = W_k z_i, \qquad u = W_q z, \qquad \widetilde{u}_{\theta_L} = u \odot s_{\theta_L}(u). \tag{6}$$

Here $s_{\theta_L} : \mathbb{R}^r \to \mathbb{R}^r$ assigns a scaling factor to each coordinate of u, and $\odot$ denotes element-wise multiplication. Then, we compute the attention weights and class probabilities as

$$\alpha_i^{\theta_L}(x) = \frac{\exp(k_i^\top \widetilde{u}_{\theta_L})}{\sum_{j=1}^{n_c} \exp(k_j^\top \widetilde{u}_{\theta_L})}, \qquad f_\theta(x; \mathcal{D}^{(c)}) = \sum_{i=1}^{n_c} \alpha_i^{\theta_L}(x) v_i,$$

where $v_i \in \{0, 1\}^{|\mathcal{Y}|}$ is the one-hot vector corresponding to the context label y_i. The probability assigned to each class is the total attention weight on context samples with that label.

To see how scaling changes the relative contribution of two context samples, note that

$$\log \frac{\alpha_i^{\theta_L}(x)}{\alpha_j^{\theta_L}(x)} = (k_i - k_j)^\top \big(u \odot s_{\theta_L}(u)\big).$$

A positive right-hand side indicates that context example i receives more attention than example j; a negative value indicates the reverse. Query scaling can change this attention ratio by rescaling individual dimensions of the query vector.

Appendix A.2.4 specifies the scaling network and its adapted parameters, while Appendix A.2.6 discusses the regression setting.

5 Experiments

5.1 Simulation

In this section, we analyze changes in attention weights before and after adaptation. We use the same data-generating model as in Figure 1. Specifically, the training distribution assigns probability 45% to each of two majority groups and 5% to each of two minority groups. At test time, all four groups have probability 25%. Only group proportions change; the distribution within each group remains fixed. Appendix A.3.1 provides the simulation details.

Table 1: Classification results on five tabular benchmarks for methods without true group labels. Throughout the tables, Avg. denotes the average of each metric across the datasets or settings in its block. Bold and underlined values indicate the best and second-best results in each column. †: accuracy results from Tong et al. (2025).

Methods	Adult		Bank		Default		Shoppers		Taxi		Avg.	
	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst
Standard methods												
ERM-MLP	73.45	45.29	71.86	42.03	63.70	28.52	76.44	50.11	75.27	57.37	72.14	44.67
XGBoost	77.09	52.58	68.69	33.87	64.78	30.72	78.03	55.42	75.81	57.57	72.88	46.03
CatBoost	76.50	51.33	68.94	34.43	64.75	31.03	77.50	54.24	76.04	58.71	72.75	45.95
TFMs												
TabICL	76.40	51.17	71.90	40.38	64.70	31.48	78.71	56.70	76.07	57.86	73.56	47.52
BETA	74.70	47.36	66.26	30.51	64.46	30.93	78.44	56.84	75.03	57.14	71.78	44.56
DistPFN	75.02	48.37	72.63	42.04	64.61	30.71	78.00	54.63	75.76	57.26	73.20	46.60
EXAONE	77.47	53.46	72.67	42.35	64.74	31.10	77.74	53.72	75.99	57.77	73.72	47.68
TabFM	77.39	52.88	73.75	44.42	65.02	31.91	78.28	54.96	76.34	58.11	74.15	48.46
Causilo	77.25	52.89	72.95	43.27	64.70	31.27	78.60	56.44	76.28	58.98	73.95	48.57
Robust methods												
EIIL†	69.37	38.97	61.85	21.69	65.05	28.91	74.82	46.18	69.52	58.00	68.12	38.75
LSR†	74.33	54.79	69.50	41.28	62.62	38.78	79.68	60.73	67.85	63.14	70.80	51.74
GEORGE	77.48	60.18	81.72	73.32	62.73	56.60	77.47	71.56	64.03	52.82	72.69	62.89
BPA	64.21	54.34	65.52	54.04	57.45	51.97	62.88	50.18	59.96	50.19	62.00	52.14
SPARE	69.81	46.64	76.34	57.52	57.68	34.70	79.93	69.67	66.08	51.24	69.97	51.95
TabPFN adaptation												
TabPFN	75.33	49.17	73.07	42.96	64.77	31.09	79.59	58.29	76.01	58.19	73.75	47.94
TabPFN (fine-tuned)	76.33	50.80	73.10	42.99	65.03	31.71	78.88	56.25	76.11	58.23	73.89	48.00
DR-TFM (GR)	<u>79.64</u>	**67.90**	<u>85.01</u>	**80.14**	**69.42**	**60.37**	**85.79**	**81.66**	<u>76.69</u>	**72.79**	**79.31**	**72.57**
DR-TFM (MS)	**79.78**	<u>67.41</u>	**85.02**	<u>78.41</u>	<u>69.18</u>	<u>57.86</u>	<u>84.48</u>	<u>77.09</u>	**77.13**	<u>65.80</u>	<u>79.12</u>	<u>69.31</u>

For each test input $x' \in \mathcal{D}'_X$ with true group g', we first compute the decoder attention weights $\alpha_i(x')$ over the context samples. We then sum these weights over context samples in the same true group to obtain $A_{\text{own}}(x') = \sum_{i:\, g_i = g'} \alpha_i(x')$. Figure 3 shows histograms of $A_{\text{own}}(x')$ separately for majority- and minority-group test samples. Since the attention weights sum to one, the remaining weight is assigned to context samples in the other three groups. Before adaptation, the median own-group attention weight is 0.97 for majority test samples and 0.23 for minority test samples. Thus, majority test samples predominantly attend to their own group, whereas minority test samples predominantly attend to other groups.

Updating only the query scaling function under our DRO objective changes the attention weights assigned to context samples. After adaptation, the median attention weights on the own group and all other groups change from $(0.97, 0.03)$ to $(0.64, 0.36)$ for majority test samples. For minority test samples, they change from $(0.23, 0.77)$ to $(0.56, 0.44)$. After adaptation, minority test samples assign a much larger share of their attention to context samples in their own group.

5.2 Real data analysis

We follow the experimental setup of Tong et al. (2025), adopting their dataset settings, evaluation group definitions, and evaluation metrics. Details are provided in Appendix A.3.

5.2.1 Experimental setup

Datasets. We evaluate the proposed method on Adult (Becker & Kohavi, 1996), Bank (Moro et al., 2014), Default (Yeh, 2009), Shoppers (Sakar & Kastro, 2018), Taxi (Navas, 2017), and ACS Income (Ding et al., 2021). For ACS Income, we consider three within-state settings and three transfers between Arizona, Massachusetts, and Michigan. All tasks are binary classification. Each evaluated binary attribute is paired with the label to define four groups.

Baselines and evaluation. We compare DR-TFM with 30 baseline methods, including standard classifiers, TFMs and their adaptations, and robust learning methods. Tables 1 and 2 report results for our TabPFN adaptations and representative baselines that do not use true group labels. Appendix A.3.3 describes the baselines. We evaluate performance using mean group accuracy Acc_{mean} and worst-group accuracy $\text{Acc}_{\text{worst}}$. Let $\mathcal{A}$ denote the set of evaluated attributes and $a_{A,g}$ the test accuracy of group $g \in [4]$ for attribute $A \in \mathcal{A}$. These metrics are defined as

Table 2: Classification results on ACS Income for methods without true group labels: within-state prediction (left) and transfers between states (right).

Methods	AZ		MA		MI		Avg.		AZ → MA		MA → MI		MI → AZ		Avg.	
	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst
Standard methods																
ERM-MLP	77.20	59.42	80.82	73.82	75.69	57.99	77.90	63.74	76.82	60.23	79.13	71.34	75.56	56.31	77.17	62.63
XGBoost	77.64	58.99	82.26	76.38	77.50	62.77	79.13	66.05	77.96	61.99	79.67	72.01	76.82	58.99	78.15	64.33
CatBoost	77.35	58.26	81.97	75.59	77.27	62.43	78.86	65.43	77.66	61.15	79.60	72.31	76.68	58.71	77.98	64.06
TFMs																
TabICL	78.14	59.46	82.89	76.50	77.84	64.30	79.62	66.75	78.03	62.08	79.73	71.78	76.98	59.69	78.25	64.52
BETA	76.73	58.08	80.67	73.25	76.80	62.12	78.07	64.49	77.25	60.70	79.39	**72.55**	76.30	58.35	77.65	63.87
DistPFN	78.09	59.55	82.98	76.96	77.82	63.78	79.63	66.76	78.76	65.16	79.43	71.58	76.70	58.56	78.29	65.10
EXAONE	78.06	58.93	82.70	76.05	77.80	63.50	79.52	66.16	78.06	61.52	79.85	72.02	76.90	59.03	78.27	64.19
TabFM	77.90	59.64	82.95	**77.10**	78.05	64.60	79.63	67.11	77.97	61.11	**79.90**	72.14	77.09	59.86	78.32	64.37
Causilo	77.77	59.45	82.87	76.52	77.77	64.04	79.47	66.67	77.90	61.53	79.85	72.02	76.93	59.31	78.23	64.29
Robust methods																
EIIL†	74.98	55.22	78.18	68.42	75.60	63.77	76.25	62.47	75.35	56.80	76.92	65.62	74.78	57.32	75.68	59.91
LSR†	76.28	66.42	78.35	73.33	75.53	67.10	76.72	68.95	75.00	62.85	74.75	68.75	74.73	64.10	74.83	65.23
GEORGE	74.76	58.70	76.65	68.82	72.82	66.52	74.74	64.68	73.25	61.69	73.22	64.80	74.21	62.10	73.56	62.86
BPA	63.26	54.44	65.78	58.90	63.42	58.18	64.15	57.17	65.65	59.36	62.72	57.64	63.57	58.96	63.98	58.65
SPARE	74.35	69.94	77.51	70.17	71.05	61.67	74.31	67.26	74.23	65.44	73.23	65.67	70.27	58.96	72.58	63.36
TabPFN adaptation																
TabPFN	78.19	59.87	**83.01**	77.02	77.72	63.50	79.64	66.80	77.95	61.57	79.69	71.86	76.81	59.08	78.15	64.17
TabPFN (fine-tuned)	78.23	60.35	82.95	77.03	78.23	65.02	79.80	67.47	77.95	62.05	79.62	72.20	76.76	58.77	78.11	64.34
DR-TFM (GR)	78.81	**70.36**	79.96	70.93	78.87	**71.90**	79.21	71.06	78.43	69.70	78.73	66.43	78.46	68.84	78.54	68.32
DR-TFM (MS)	**79.46**	69.30	81.24	72.75	**79.30**	71.42	**80.00**	**71.16**	**80.08**	**73.75**	78.82	70.53	**78.55**	**71.38**	**79.15**	**71.89**

Table 3: Runtime in seconds and peak GPU memory in GB per result, each averaged over the five tabular benchmarks and three seeds. "ft" denotes fine-tuning. Runtime includes hyperparameter selection.

	GEORGE	LSR	MixturePFN	BETA	TabPFN	TabPFN (ft)	**DR-TFM (GR)**	**DR-TFM (MS)**
Seconds	77.0	5,156.4	229.5	1,020.6	5.4	214.1	23.8	30.6
Memory (GB)	1.21	3.12	2.17	14.33	0.61	9.25	1.07	0.98

$\mathrm{Acc}_{\mathrm{mean}} = \frac{1}{|\mathcal{A}|}\sum_{A\in\mathcal{A}}\frac{1}{4}\sum_{g=1}^{4} a_{A,g}$ and $\mathrm{Acc}_{\mathrm{worst}} = \frac{1}{|\mathcal{A}|}\sum_{A\in\mathcal{A}}\min_{g\in[4]} a_{A,g}$. Both metrics are expressed as percentages and denoted by Mean and Worst in the tables.

DR-TFM does not require true group labels for either training or model selection. We use k-means to estimate groups for GR and construct source subsets for MS. Appendix A.3 provides the training protocols, hyperparameter selection procedures, and ablations on the number of clusters.

5.2.2 Results

Tables 1 and 2 summarize benchmark performance on the five tabular benchmarks and ACS Income, respectively. Full results, including additional baselines and standard deviations, are reported in Appendices A.4 and A.8. We report accuracy differences in percentage points (pp).

Five tabular benchmarks. Table 1 shows that our proposed methods achieve the best performance on all five tabular benchmarks. The corresponding averages of worst-group accuracies are 72.57% and 69.31%, yielding improvements of 24.63 and 21.37 pp over TabPFN. DR-TFM (GR) and DR-TFM (MS) improve the average of mean group accuracies over TabPFN by 5.56 and 5.37 pp and the average of worst-group accuracies over TabPFN fine-tuning by 24.57 and 21.31 pp, respectively. The corresponding gains over GEORGE (k-means) are 9.68 and 6.42 pp.

ACS Income. In Table 2, our proposed methods achieve the best and second-best averages of worst-group accuracies in both within-state and transfer settings. Within states, DR-TFM (GR) and DR-TFM (MS) achieve 71.06% and 71.16%, exceeding TabPFN by 4.26 and 4.36 pp, respectively. Across transfers, they achieve 68.32% and 71.89%. These exceed TabPFN by 4.15 and 7.72 pp and LSR by 3.09 and 6.66 pp, respectively. Both also outperform TabPFN and fine-tuned TabPFN in the average of mean group accuracies in the transfer settings.

Computational and memory efficiency. Table 3 reports runtime in seconds and peak GPU memory in GB per result, each averaged over the five tabular benchmarks. DR-TFM (GR) and DR-TFM (MS) require 23.8 and 30.6 seconds per result, respectively. These correspond to 9.0- and 7.0-fold

Table 4: Average of worst-group accuracies over five tabular benchmarks and three seeds. Trainable (%) is the percentage of model parameters updated. Encoder refers to all model parameters outside the decoder. Parenthesized values denote improvements over ERM in percentage points. Seconds and Memory (GB) are runtime and peak GPU memory per result under DRO-based adaptation.

Fine-tuned parameters	Trainable (%)	ERM (%)	DRO (%)	Seconds	Memory (GB)
Input adapter	0.063–0.155	45.45	66.64 (+21.19)	43.5	9.80
Encoder	99.196	48.39	68.92 (+20.53)	241.2	10.22
Full backbone	≈ 100	48.00	68.98 (+20.98)	246.6	10.31
Decoder	0.804	48.68	71.93 (+23.25)	24.0	1.10
Query scaling	0.016	47.34	72.57 (+25.23)	23.8	1.07

Table 5: GR adaptation across tabular foundation models without true group labels. Results are averaged over three seeds. "+ Ours" denotes DR-TFM (GR) applied to the model above, with parentheses indicating the percentage of model parameters updated. Bold parentheses report the change from the corresponding original model, in percentage points.

Method	Five tabular benchmarks (Avg.)		ACS within-state (Avg.)		ACS transfer (Avg.)	
	Mean	Worst	Mean	Worst	Mean	Worst
TabPFN	73.75	47.94	79.64	66.80	78.15	64.17
+ Ours (0.016%)	79.31 **(+5.56)**	72.57 **(+24.63)**	79.21 **(-0.43)**	71.06 **(+4.26)**	78.54 **(+0.39)**	68.32 **(+4.15)**
TabPFN-3.5	74.39	49.26	79.79	67.10	78.25	64.34
+ Ours (0.004%)	79.42 **(+5.03)**	71.47 **(+22.21)**	79.88 **(+0.09)**	72.48 **(+5.38)**	79.19 **(+0.94)**	71.51 **(+7.17)**
EXAONE	73.72	47.68	79.52	66.16	78.27	64.19
+ Ours (0.020%)	79.44 **(+5.72)**	69.45 **(+21.77)**	80.28 **(+0.76)**	71.39 **(+5.23)**	79.63 **(+1.36)**	70.37 **(+6.18)**
TabFM	74.15	48.46	79.63	67.11	78.32	64.37
+ Ours (0.002%)	79.43 **(+5.28)**	66.47 **(+18.01)**	79.65 **(+0.02)**	69.41 **(+2.30)**	78.98 **(+0.66)**	66.60 **(+2.23)**
Causilo	73.95	48.57	79.47	66.67	78.23	64.29
+ Ours (0.046%)	78.47 **(+4.52)**	63.56 **(+14.99)**	80.22 **(+0.75)**	72.15 **(+5.48)**	78.37 **(+0.14)**	68.21 **(+3.92)**

speedups over TabPFN standard fine-tuning (214.1 seconds), and 42.9- and 33.3-fold speedups over BETA ($1{,}020.6$ seconds). Both also use less GPU memory: their average peak memory is 1.07 and 0.98 GB, respectively, compared with 9.25 GB for TabPFN standard fine-tuning. Further details are provided in Appendix A.5.

Training objective and adaptation strategy. Table 4 compares five fine-tuning strategies under ERM (2) and DRO (4). Our DRO-based adaptation substantially improves the average of worst-group accuracies over ERM across all five strategies. Under DRO-based adaptation, query scaling achieves the highest average of worst-group accuracies (72.57%) with the fewest trainable parameters (0.016% of the model). Its accuracy is slightly higher than that of decoder fine-tuning (71.93%), while using approximately $1/51$ as many trainable parameters. Appendix A.6 provides the adaptation settings and per-dataset results.

Extension to other tabular foundation models. Table 5 evaluates our robust adaptation on TabPFN-3.5 (Prior Labs, 2026) and three other TFMs: EXAONE (Eo et al., 2026), TabFM (Google Research, 2026), and Causilo (Nums AI Inc., 2026). We fine-tune the pretrained query-scaling network when present, and otherwise add and train a new one. All four models consistently improve the average of worst-group accuracies on the five tabular benchmarks, with an average gain of 19.25 pp across models. These results support the applicability of our adaptation beyond TabPFN. Detailed results and experimental settings are provided in Appendix A.8.

6 Conclusion

We presented DR-TFM, a parameter-efficient framework for distributionally robust TFM adaptation under subpopulation shift without true group labels. Its two instantiations construct ambiguity sets from estimated groups or source conditional distributions. Query scaling preserves pretrained representations while offering computational and memory efficiency and scalability. Experiments demonstrate improved average worst-group accuracy with competitive mean group accuracy across multiple TFMs. Future work will extend the framework to a broader range of distribution shifts.

REFERENCES

Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *Proc. International Conference on Machine Learning*, pp. 2189–2200, 2021.

Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring Adult: New datasets for fair machine learning. In *Proc. Advances in Neural Information Processing Systems*, volume 34, pp. 6478–6490, 2021.

John C. Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3):1378–1406, 2021.

John C. Duchi, Peter W. Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *Mathematics of Operations Research*, 46(3):946–969, 2021.

Moonjung Eo, Min-Kook Suh, Hye-Seung Cho, Jiwon Kim, Seoyoon Kim, Sangjun Nam, and Soonyoung Lee. EXAONE Tabular 1.0: Technical report. *ArXiv:2608.25774*, 2026.

Benjamin Feuer, Robin Tibor Schirrmeister, Valeriia Cherepanova, Chinmay Hegde, Frank Hutter, Micah Goldblum, Niv Cohen, and Colin White. TuneTables: Context optimization for scalable prior-data fitted networks. In *Proc. Advances in Neural Information Processing Systems*, volume 37, pp. 83430–83464, 2024.

Google Research. TabFM: Tabular foundation models. `https://github.com/google-research/tabfm`, 2026.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin Hoo, Magnus Bühler, Anurag Garg, et al. TabPFN-3: Technical report. *ArXiv:2605.13986*, 2026.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Jun-Peng Jiang, Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, and Han-Jia Ye. Representation learning for tabular data: A comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(6):6488–6508, 2026.

Sung Ho Jo, Seonghwi Kim, and Minwoo Chae. Mitigating spurious correlation via distributionally robust learning with hierarchical ambiguity sets. In *Proc. International Conference on Learning Representations*, 2026.

Patrik Kenfack, Samira Ebrahimi Kahou, and Ulrich Aïvodji. Towards fair in-context learning with tabular foundation models. *Transactions on Machine Learning Research*, 2026.

Seonghwi Kim, Sung Ho Jo, Wooseok Ha, and Minwoo Chae. Distributionally robust classification for multi-source unsupervised domain adaptation. In *Proc. International Conference on Learning Representations*, 2026.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (REx). In *Proc. International Conference on Machine Learning*, pp. 5815–5826, 2021.

Seunghan Lee. Mitigating label shift in tabular in-context learning via test-time posterior adjustment. In *Proc. International Conference on Machine Learning*, 2026.

Daniel Levy, Yair Carmon, John C. Duchi, and Aaron Sidford. Large-scale methods for distributionally robust optimization. In *Proc. Advances in Neural Information Processing Systems*, volume 33, pp. 8847–8860, 2020.

Evan Zheran Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *Proc. International Conference on Machine Learning*, pp. 6781–6792, 2021.

Jiashuo Liu, Tianyu Wang, Peng Cui, and Hongseok Namkoong. On the need for a language describing distribution shifts: Illustrations on tabular datasets. In *Proc. Advances in Neural Information Processing Systems*, volume 36, pp. 51371–51408, 2023.

Si-Yang Liu and Han-Jia Ye. TabPFN unleashed: A scalable and effective solution to tabular classification problems. In *Proc. International Conference on Machine Learning*, pp. 40043–40068, 2025.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Alex Labach, Hamidreza Kamkari, Jesse C. Cresswell, Keyvan Golestan, Guangwei Yu, Anthony L. Caterini, and Maksims Volkovs. TabDPT: Scaling tabular foundation models on real data. In *Proc. Advances in Neural Information Processing Systems*, volume 38, pp. 172692–172722, 2025.

Peyman Mohajerin Esfahani and Daniel Kuhn. Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1–2):115–166, 2018.

Sérgio Moro, Paulo Rita, and Paulo Cortez. Bank marketing. UCI Machine Learning Repository, 2014.

Emilio Navas. Taxi routes of Mexico City, Quito and more. Kaggle, 2017. https://www.kaggle.com/datasets/mnavas/taxi-routes-for-mexico-city-and-quito.

Nums AI Inc. Causilo. `https://github.com/nums-ai/causilo`, 2026.

Henning Petzka, Michael Kamp, Linara Adilova, Cristian Sminchisescu, and Mario Boley. Relative flatness and generalization. In *Proc. Advances in Neural Information Processing Systems*, volume 34, pp. 18420–18432, 2021.

Prior Labs. TabPFN-3.5. `https://huggingface.co/Prior-Labs/tabpfn_3_5`, 2026.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. CatBoost: Unbiased boosting with categorical features. In *Proc. Advances in Neural Information Processing Systems*, volume 31, pp. 6638–6648, 2018.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *Proc. International Conference on Machine Learning*, pp. 50817–50847, 2025.

Jake Robertson, Noah Hollmann, Samuel Müller, Noor Awad, and Frank Hutter. FairPFN: A tabular foundation model for causal fairness. In *Proc. International Conference on Machine Learning*, pp. 51787–51808, 2025.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *Proc. International Conference on Learning Representations*, 2020.

C. Okan Sakar and Yomi Kastro. Online shoppers purchasing intention dataset. UCI Machine Learning Repository, 2018.

Seonguk Seo, Joon-Young Lee, and Bohyung Han. Unsupervised learning of debiased representations with pseudo-attributes. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16742–16751, 2022.

Zheyan Shen, Peng Cui, Tong Zhang, and Kun Kuang. Stable learning via sample reweighting. In *Proc. AAAI Conference on Artificial Intelligence*, volume 34, pp. 5692–5699, 2020.

Nimit Sohoni, Jared A. Dunnmon, Geoffrey Angus, Albert Gu, and Christopher Ré. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. In *Proc. Advances in Neural Information Processing Systems*, volume 33, pp. 19339–19352, 2020.

Valentin Thomas, Junwei Ma, Rasa Hosseinzadeh, Keyvan Golestan, Guangwei Yu, Maksims Volkovs, and Anthony Caterini. Retrieval & fine-tuning for in-context tabular models. In *Proc. Advances in Neural Information Processing Systems*, volume 37, pp. 108439–108467, 2024.

Yunze Tong, Fengda Zhang, Zihao Tang, Kaifeng Gao, Kai Huang, Pengfei Lyu, Jun Xiao, and Kun Kuang. Latent score-based reweighting for robust classification on imbalanced tabular data. In *Proc. International Conference on Machine Learning*, pp. 59846–59866, 2025.

Derek Xu, Olcay Cirit, Reza Asadi, Yizhou Sun, and Wei Wang. Mixture of in-context prompters for tabular PFNs. In *Proc. International Conference on Learning Representations*, 2025.

Yu Yang, Eric Gan, Gintare Karolina Dziugaite, and Baharan Mirzasoleiman. Identifying spurious biases early in training through the lens of simplicity bias. In *Proc. International Conference on Artificial Intelligence and Statistics*, pp. 2953–2961, 2024.

Yuzhe Yang, Haoran Zhang, Dina Katabi, and Marzyeh Ghassemi. Change is hard: A closer look at subpopulation shift. In *Proc. International Conference on Machine Learning*, pp. 39584–39622, 2023.

Han-Jia Ye, Si-Yang Liu, and Wei-Lun Chao. A closer look at TabPFN v2: Understanding its strengths and extending its capabilities. In *Proc. Advances in Neural Information Processing Systems*, volume 38, pp. 135605–135637, 2025.

I-Cheng Yeh. Default of credit card clients. UCI Machine Learning Repository, 2009.

Yingtian Zou, Kenji Kawaguchi, Yingnan Liu, Jiashuo Liu, Mong-Li Lee, and Wynne Hsu. Towards robust out-of-distribution generalization bounds via sharpness. In *Proc. International Conference on Learning Representations*, 2024.

A APPENDIX

CONTENTS

A.1 Theoretical analysis of subpopulation shift 14
A.2 Method details 17
A.2.1 Inputs and data preparation 17
A.2.2 Optimization for DR-TFM (GR) 18
A.2.3 Optimization for DR-TFM (MS) 18
A.2.4 Query scaling architecture 20
A.2.5 Illustration of query scaling 21
A.2.6 Regression 21
A.3 Experimental setup 23
A.3.1 Simulation details 23
A.3.2 Datasets 24
A.3.3 Baselines 25
A.3.4 Group and source construction for DR-TFM 27
A.3.5 Evaluation metrics 27
A.3.6 Training protocol 28
A.3.7 Model selection and hyperparameters 28
A.3.8 Ablation on the number of clusters 30
A.4 Full result tables 31
A.5 Computational cost 34
A.6 Training objective and adaptation strategy 35
A.7 Discussion of the χ^2-DRO formulation 36
A.8 Extension to other tabular foundation models 37

A.1 THEORETICAL ANALYSIS OF SUBPOPULATION SHIFT

We characterize how changes in subpopulation proportions affect predictive risk, then examine how group imbalance can produce differences in group risks.

- **Proposition A.1** shows when subpopulation shift can have a large effect on predictive risk. The effect depends on how much the group proportions change, how different the group risks are, and the direction of the shift.
- **Proposition A.3** explains how group imbalance can create differences in risk across subpopulations. Under average-risk minimization, stronger imbalance-induced spurious correlation can lead to a larger group risk gap.

Proposition A.1 (Risk variation under subpopulation mixture shift)**.** *Suppose that the training and test distributions are mixtures of the same K subpopulation distributions,*

$$P^{\mathrm{tr}} = \sum_{k=1}^{K} \pi_k^{\mathrm{tr}} P_k, \qquad P^{\mathrm{te}} = \sum_{k=1}^{K} \pi_k^{\mathrm{te}} P_k,$$

where $\pi^{\mathrm{tr}}, \pi^{\mathrm{te}} \in \Delta_{K-1}$.

For a fixed model parameter θ, define the group risks by

$$R_k(\theta) := \mathbb{E}_{P_k}\big[\ell(f_\theta(X), Y)\big].$$

Assume these risks are finite, and define the training and test risks by $R^{\mathrm{tr}}(\theta) = \sum_k \pi_k^{\mathrm{tr}} R_k(\theta)$ and $R^{\mathrm{te}}(\theta) = \sum_k \pi_k^{\mathrm{te}} R_k(\theta)$. Let

$$R_{\max}(\theta) := \max_{k=1,\dots,K} R_k(\theta), \qquad R_{\min}(\theta) := \min_{k=1,\dots,K} R_k(\theta).$$

Then, there exists $A(\theta; \pi^{\mathrm{tr}}, \pi^{\mathrm{te}}) \in [-1, 1]$ such that

$$R^{\mathrm{te}}(\theta) - R^{\mathrm{tr}}(\theta) = d_{\mathrm{TV}}(\pi^{\mathrm{te}}, \pi^{\mathrm{tr}})\big(R_{\max}(\theta) - R_{\min}(\theta)\big) A(\theta; \pi^{\mathrm{tr}}, \pi^{\mathrm{te}}),$$

where

$$d_{\mathrm{TV}}(\pi^{\mathrm{te}}, \pi^{\mathrm{tr}}) := \frac{1}{2} \sum_{k=1}^{K} \big|\pi_k^{\mathrm{te}} - \pi_k^{\mathrm{tr}}\big|$$

denotes the total variation distance.

Consequently,

$$\big|R^{\mathrm{te}}(\theta) - R^{\mathrm{tr}}(\theta)\big| \le d_{\mathrm{TV}}(\pi^{\mathrm{te}}, \pi^{\mathrm{tr}})\big(R_{\max}(\theta) - R_{\min}(\theta)\big).$$

The bound depends on the magnitude of the mixture shift and the range of group risks. The alignment factor A determines the direction and size of the actual risk change.

Proof of Proposition A.1. For a fixed θ,

$$R^{\mathrm{te}}(\theta) - R^{\mathrm{tr}}(\theta) = \sum_{k=1}^{K} \big(\pi_k^{\mathrm{te}} - \pi_k^{\mathrm{tr}}\big)\, R_k(\theta).$$

Let

$$\Delta_k := \pi_k^{\mathrm{te}} - \pi_k^{\mathrm{tr}}.$$

Since both π^{tr} and π^{te} are probability vectors,

$$\sum_{k=1}^{K} \Delta_k = 0.$$

By the definition of total variation distance,

$$s := d_{\mathrm{TV}}(\pi^{\mathrm{te}}, \pi^{\mathrm{tr}}) = \frac{1}{2}\sum_{k=1}^{K} |\Delta_k| \\ = \sum_{k=1}^{K} [\Delta_k]_+ = \sum_{k=1}^{K} [-\Delta_k]_+, \tag{7}$$

where $[a]_+ := \max\{a, 0\}$ denotes the positive part of a. If $s = 0$, then $\pi^{\mathrm{te}} = \pi^{\mathrm{tr}}$ and hence $R^{\mathrm{te}}(\theta) = R^{\mathrm{tr}}(\theta)$, so the result holds by taking $A(\theta; \pi^{\mathrm{tr}}, \pi^{\mathrm{te}}) = 0$.

Suppose now that $s > 0$, and define

$$q_k^+ := \frac{[\Delta_k]_+}{s}, \qquad q_k^- := \frac{[-\Delta_k]_+}{s}, \qquad k = 1, \ldots, K.$$

By (7), both q^+ and q^- belong to Δ_{K-1}. Moreover, by definition,

$$\Delta_k = s\left(q_k^+ - q_k^-\right).$$

Therefore,

$$R^{\mathrm{te}}(\theta) - R^{\mathrm{tr}}(\theta) = s\left(\sum_{k=1}^{K} q_k^+ R_k(\theta) - \sum_{k=1}^{K} q_k^- R_k(\theta)\right). \tag{8}$$

Here, q^+ represents the normalized probability mass added to subpopulations at test time, whereas q^- represents the normalized probability mass removed from subpopulations.

Since both q^+ and q^- are probability vectors,

$$R_{\min}(\theta) \le \sum_{k=1}^{K} q_k^\sigma R_k(\theta) \le R_{\max}(\theta), \qquad \sigma \in \{+, -\}.$$

Therefore,

$$-\big(R_{\max}(\theta) - R_{\min}(\theta)\big) \le \sum_{k=1}^{K} q_k^+ R_k(\theta) - \sum_{k=1}^{K} q_k^- R_k(\theta) \le R_{\max}(\theta) - R_{\min}(\theta). \tag{9}$$

If $R_{\max}(\theta) = R_{\min}(\theta)$, all group risks are identical, and (8) is zero. Thus, the result holds by taking $A(\theta; \pi^{\mathrm{tr}}, \pi^{\mathrm{te}}) = 0$.

Otherwise, define

$$A(\theta; \pi^{\mathrm{tr}}, \pi^{\mathrm{te}}) := \frac{\sum_{k=1}^{K} q_k^+ R_k(\theta) - \sum_{k=1}^{K} q_k^- R_k(\theta)}{R_{\max}(\theta) - R_{\min}(\theta)}. \tag{10}$$

By (9),

$$A(\theta; \pi^{\mathrm{tr}}, \pi^{\mathrm{te}}) \in [-1, 1].$$

Hence, A measures the alignment between the direction of the mixture shift and the group risks: $A > 0$ indicates a shift toward higher-risk subpopulations, whereas $A < 0$ indicates a shift toward lower-risk subpopulations.

Substituting (10) into (8) gives

$$R^{\mathrm{te}}(\theta) - R^{\mathrm{tr}}(\theta) = d_{\mathrm{TV}}(\pi^{\mathrm{te}}, \pi^{\mathrm{tr}})\big(R_{\max}(\theta) - R_{\min}(\theta)\big)A(\theta; \pi^{\mathrm{tr}}, \pi^{\mathrm{te}}).$$

Finally, since $|A(\theta; \pi^{\mathrm{tr}}, \pi^{\mathrm{te}})| \le 1$,

$$\big|R^{\mathrm{te}}(\theta) - R^{\mathrm{tr}}(\theta)\big| \le d_{\mathrm{TV}}(\pi^{\mathrm{te}}, \pi^{\mathrm{tr}})\big(R_{\max}(\theta) - R_{\min}(\theta)\big),$$

which proves the result. □

The following corollary illustrates this shift-risk alignment more explicitly in the two-subpopulation case.

Corollary A.2 (Two-subpopulation case)**.** *Suppose $K = 2$ and $R_2(\theta) \geq R_1(\theta)$. Let*

$$\Delta := \pi_2^{\mathrm{te}} - \pi_2^{\mathrm{tr}}.$$

Then,

$$R^{\mathrm{te}}(\theta) - R^{\mathrm{tr}}(\theta) = \Delta\big(R_2(\theta) - R_1(\theta)\big).$$

Moreover,

$$d_{\mathrm{TV}}(\pi^{\mathrm{te}}, \pi^{\mathrm{tr}}) = |\Delta|,$$

and hence

$$\big|R^{\mathrm{te}}(\theta) - R^{\mathrm{tr}}(\theta)\big| = d_{\mathrm{TV}}(\pi^{\mathrm{te}}, \pi^{\mathrm{tr}})\big(R_{\max}(\theta) - R_{\min}(\theta)\big).$$

For $\Delta \neq 0$ and $R_2(\theta) > R_1(\theta)$, the alignment factor in Proposition A.1 satisfies

$$A(\theta; \pi^{\mathrm{tr}}, \pi^{\mathrm{te}}) = \mathrm{sign}(\Delta).$$

The result follows directly from Proposition A.1 by setting $K = 2$.

The preceding results concern a fixed predictor. We next examine how fitting a predictor to an imbalanced distribution can lead to unequal group risks.

Proposition A.3 (Risk heterogeneity under imbalance-induced spurious correlation)**.** *Let $Y, S \in \{-1, +1\}$ and let $X = (Z, S)$, where S denotes a binary spurious feature and Z collects the remaining features. Let P_1 and P_2 be two subpopulation distributions over (X, Y) such that*

$$S = Y \quad P_1\text{-a.s.}, \qquad S = -Y \quad P_2\text{-a.s.},$$

and assume that Y is balanced under both P_1 and P_2.

For $\rho \in [1/2, 1)$, define

$$P_\rho = \rho P_1 + (1 - \rho)P_2.$$

Then

$$\mathrm{Corr}_{P_\rho}(S, Y) = 2\rho - 1.$$

For $g \in \{1, 2\}$, define

$$R_g(f) = \mathbb{E}_{P_g}\left[\ell(f(X), Y)\right],$$

and let

$$f_\rho \in \arg\min_{f \in \mathcal{F}} \left\{\rho R_1(f) + (1 - \rho)R_2(f)\right\}.$$

Here $\mathcal{F}$ is a fixed hypothesis class; the group risks are assumed finite and the minimum is assumed to be attained.

Then, for any $1/2 \leq \rho_1 < \rho_2 < 1$,

$$R_2(f_{\rho_1}) - R_1(f_{\rho_1}) \leq R_2(f_{\rho_2}) - R_1(f_{\rho_2}).$$

Moreover, if

$$R_1(f_{1/2}) = R_2(f_{1/2}),$$

then, defining

$$R_{\max}(f) = \max_{g \in \{1,2\}} R_g(f), \qquad R_{\min}(f) = \min_{g \in \{1,2\}} R_g(f),$$

the group risk gap

$$R_{\max}(f_\rho) - R_{\min}(f_\rho)$$

is non-decreasing in ρ.

Proof. We first compute the correlation between S and Y, then compare the group risks at two mixture proportions. Since Y is balanced under both subpopulations and $S = Y$ under P_1 while $S = -Y$ under P_2, both variables are balanced under P_ρ. Thus, their means are zero and their variances are one, giving

$$\mathrm{Corr}_{P_\rho}(S, Y) = \mathbb{E}_{P_\rho}[SY] = \rho\,\mathbb{E}_{P_1}[SY] + (1-\rho)\,\mathbb{E}_{P_2}[SY] = 2\rho - 1.$$

Define $L_\rho(f) = \rho R_1(f) + (1-\rho)R_2(f)$, and let $f_1 = f_{\rho_1}$ and $f_2 = f_{\rho_2}$ for $1/2 \le \rho_1 < \rho_2 < 1$. By optimality,

$$L_{\rho_1}(f_1) \le L_{\rho_1}(f_2), \qquad L_{\rho_2}(f_2) \le L_{\rho_2}(f_1).$$

Combining these inequalities yields

$$L_{\rho_2}(f_2) - L_{\rho_1}(f_2) \le L_{\rho_2}(f_1) - L_{\rho_1}(f_1).$$

For any $f \in \mathcal{F}$,

$$L_{\rho_2}(f) - L_{\rho_1}(f) = (\rho_2 - \rho_1)\big(R_1(f) - R_2(f)\big).$$

Since $\rho_2 - \rho_1 > 0$, we obtain

$$R_2(f_{\rho_1}) - R_1(f_{\rho_1}) \le R_2(f_{\rho_2}) - R_1(f_{\rho_2}).$$

Finally, if $R_1(f_{1/2}) = R_2(f_{1/2})$, this monotonicity implies $R_2(f_\rho) - R_1(f_\rho) \ge 0$ for every $\rho \in [1/2, 1)$. Consequently,

$$R_{\max}(f_\rho) - R_{\min}(f_\rho) = R_2(f_\rho) - R_1(f_\rho),$$

which is non-decreasing in ρ by the preceding inequality. □

A.2 Method details

We first describe the inputs, group estimation, and data preparation used for classification. We then detail the optimization procedures for DR-TFM (GR) and DR-TFM (MS), followed by the query scaling architecture and an extension to regression.

A.2.1 Inputs and data preparation

The data preparation below applies to the real-data variants without true group labels. Appendix A.3.4 specifies the variants with true groups, and Appendix A.3.1 describes the simulations.

Inputs. Both methods take a training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ and a pretrained TabPFN f_{θ_0}. Training response labels y_i are available, but true group labels are not required. A separate validation set is used for selecting hyperparameters and the adapted model; its samples are not included in the context or the adaptation objective.

Embeddings for group estimation. Using the full training dataset $\mathcal{D}$ as context, we extract a frozen pretrained TabPFN embedding for each training input, averaging across ensemble members. We standardize each coordinate using its training mean and standard deviation to obtain the features $\widetilde{z}_i$ used for clustering. The same standardization is applied to validation embeddings.

Class-wise group estimation. For each response class $c \in \mathcal{Y}$, we fit a separate clustering rule C_c with G_c clusters to $\{\widetilde{z}_i : y_i = c\}$. Each training sample is assigned to the pair $(y_i, C_{y_i}(\widetilde{z}_i))$. We reindex these pairs as $\hat{g}_i \in [G]$, where $G = \sum_{c\in\mathcal{Y}} G_c$. The main tables use DR-TFM (GR; k-means), which assigns each training sample to the nearest fitted centroid within its class. For DR-TFM (GR), the number of clusters per class is selected following GEORGE (Sohoni et al., 2020) (Appendix A.3.7). The assignments remain fixed throughout adaptation.

The additional GMM variant fits a diagonal-covariance GMM within each class and assigns each training sample to the component with the largest posterior probability. It uses the same criterion to select the number of clusters.

Context and loss-evaluation sets. We split $\mathcal{D}$ into a context $\mathcal{D}^{(c)}$ and a loss-evaluation set $\mathcal{D}^{(e)}$ in a $70/30$ ratio, stratifying by $\hat{g}_i$. The two sets are disjoint and both retain response labels. After this split, we recompute embeddings for adaptation using $\mathcal{D}^{(c)}$ as context, writing $z(x)$ for the frozen pretrained embedding of input x. The context samples and their labels remain fixed during adaptation and prediction. DR-TFM (GR) uses the response labels in $\mathcal{D}^{(e)}$ to compute group losses. DR-TFM (MS) uses the inputs in $\mathcal{D}^{(e)}$ together with soft labels obtained from source-conditioned pretrained predictions, as described below.

Adapted parameters. In both classification methods, θ_L collects the pretrained query scaling network parameters selected for fine-tuning. The feature map and all remaining parameters constitute θ_F and are kept fixed; the architecture is specified in Appendix A.2.4. The algorithms below describe T optimization iterations. We write $\theta^{(t)} = (\theta_F, \theta_L^{(t)})$ for the full model parameters after t updates of θ_L. We select the adapted model with the highest worst-group validation accuracy over estimated groups. Training schedules and model selection details are provided in Appendices A.3.6 and A.3.7. For either method, prediction for a new input x uses $f_\theta(x; \mathcal{D}^{(c)})$ with the selected parameters and the retained context. It requires neither a group assignment for x nor adversarial updates.

A.2.2 Optimization for DR-TFM (GR)

Group losses. Using the fixed assignments from Appendix A.2.1, define the loss-evaluation indices for each estimated group k as $\widehat{\mathcal{I}}_k = \{i : (x_i, y_i) \in \mathcal{D}^{(e)}, \hat{g}_i = k\}$. Let $n_k = |\widehat{\mathcal{I}}_k| > 0$ denote its size. Its empirical joint distribution is $\widehat{P}_k^{(e)} = n_k^{-1} \sum_{i \in \widehat{\mathcal{I}}_k} \delta_{(x_i, y_i)}$. Substituting the ambiguity set in (4) into (3) yields

$$\min_{\theta_L} \max_{q \in \Delta_{G-1}} \sum_{k=1}^{G} q_k \left[\frac{1}{|\widehat{\mathcal{I}}_k|} \sum_{i \in \widehat{\mathcal{I}}_k} \ell\Big(f_\theta(x_i; \mathcal{D}^{(c)}), y_i \Big) \right]. \tag{11}$$

Here, q_k is the weight assigned to estimated group k, and the bracketed quantity is its mean loss, denoted by $L_k(\theta)$. For classification, ℓ is cross-entropy with the observed response label. Maximizing over $q \in \Delta_{G-1}$ selects the largest empirical group loss.

Alternating updates. We initialize q uniformly and θ_L from the pretrained query scaling network. Each iteration first computes the mean loss of each estimated group, then increases the weights of groups with larger losses by exponentiated gradient ascent. Holding these updated weights fixed, we update θ_L to reduce the weighted group losses. Algorithm 1 summarizes the basic updates for (11). In our experiments, we use the group-adjusted DRO variant of Sagawa et al. (2020), with group-size adjustments $C/\sqrt{n_k}$, where we set $C = 5$. Further optimization settings are provided in Appendix A.3.7.

A.2.3 Optimization for DR-TFM (MS)

Source construction. We first prepare $\mathcal{D}^{(c)}$ and $\mathcal{D}^{(e)}$ as in Appendix A.2.1. We then form labeled source subsets $\mathcal{S}_1, \ldots, \mathcal{S}_M$ from $\mathcal{D}^{(c)}$. Here, M denotes the number of retained sources. For MS-C, we apply k-means to the fixed context embeddings jointly across response classes. Each retained cluster forms a source; clusters containing only one response class are excluded. For MS-R, we construct each source independently by sampling without replacement from the full context. All context samples are available again when constructing the next source, so different sources may overlap. The initial source count and subset sizes are specified in Appendix A.3.4.

Source conditional predictions. For each source $\mathcal{S}_k$, we condition the frozen pretrained TabPFN on its inputs and response labels and evaluate every input x_i in $\mathcal{D}^{(e)}$: $\widehat{P}^{(k)}_{Y|X}(\cdot \mid x_i) = f_{\theta_0}(x_i; \mathcal{S}_k)$. Thus, each source provides a class probability vector for every loss-evaluation input. We compute these probabilities at the original inputs and keep them and the source subsets fixed throughout adaptation. Given mixture weights β, the soft label is $y^\circ(\beta, x_i) = \sum_{k=1}^{M} \beta_k \widehat{P}^{(k)}_{Y|X}(\cdot \mid x_i)$. The mixture weights represent uncertainty over the relative contributions of these source-conditioned predictions. For a predicted probability vector p and a soft label v, we use $\ell(p, v) = -\sum_{y \in \mathcal{Y}} v_y \log p_y$. The adapted model always uses the full context $\mathcal{D}^{(c)}$.

Algorithm 1 DR-TFM (GR) for (11)

Input: training dataset $\mathcal{D}$; validation set; pretrained TabPFN f_{θ_0}; group estimation rule; learning rates η_{θ_L}, η_q; update steps T
Estimate groups $\hat{g}_i \in [G]$ from frozen pretrained embeddings
Prepare the fixed context $\mathcal{D}^{(c)}$ and loss-evaluation set $\mathcal{D}^{(e)}$ as in Appendix A.2.1
Obtain frozen embeddings using $\mathcal{D}^{(c)}$ as context
Define $\widehat{\mathcal{I}}_k \leftarrow \{i : (x_i, y_i) \in \mathcal{D}^{(e)}, \hat{g}_i = k\}$ and $n_k \leftarrow |\widehat{\mathcal{I}}_k|$ for $k \in [G]$
Initialize $q^{(0)} \leftarrow (1/G, \ldots, 1/G)$ and $\theta_L^{(0)}$ from the pretrained query scaling network; freeze all other parameters
for $t = 0, \ldots, T-1$ **do**
    Compute the group-wise losses

$$L_k(\theta^{(t)}) \leftarrow \frac{1}{n_k} \sum_{i \in \widehat{\mathcal{I}}_k} \ell\Big(f_{\theta^{(t)}}(x_i; \mathcal{D}^{(c)}), y_i\Big)$$

    Update group weights by exponentiated gradient ascent:

$$q_k^{(t+1)} \propto q_k^{(t)} \exp\Big(\eta_q L_k(\theta^{(t)})\Big), \qquad k \in [G]$$

    Normalize $q^{(t+1)}$ such that $\sum_{k=1}^{G} q_k^{(t+1)} = 1$
    Update the fine-tuning parameters:

$$\theta_L^{(t+1)} \leftarrow \theta_L^{(t)} - \eta_{\theta_L} \nabla_{\theta_L} \sum_{k=1}^{G} q_k^{(t+1)} L_k(\theta)\Big|_{\theta_L = \theta_L^{(t)}}$$

end for
Select an iterate using validation, as in Appendix A.2.1
Output: selected adapted TabPFN and fixed context $\mathcal{D}^{(c)}$

Surrogate objective. The Wasserstein distance in (5) uses transport cost $c(x, x') = \|z(x) - z(x')\|_2$, using the adaptation embeddings defined in Appendix A.2.1. Let $f_{\theta,Z}(z')$ denote the decoder prediction at the input embedding z', with $\mathcal{D}^{(c)}$ fixed as context, so that $f_{\theta,Z}(z(x)) = f_\theta(x; \mathcal{D}^{(c)})$. We adopt the surrogate objective proposed by Kim et al. (2026) and optimize it over the query scaling parameters:

$$\min_{\theta_L} \sup_{\substack{\beta \in \Delta_{M-1} \\ \|\beta - \bar{\beta}\|_2 \le \epsilon_2}} \mathbb{E}_{X \sim \widehat{P}_X^{(e)}} \left[\sup_{\|z' - z(X)\|_2 \le \epsilon_1} \ell(f_{\theta,Z}(z'), y^\circ(\beta, X)) \right].$$

Here, the expectation is over the empirical input distribution of $\mathcal{D}^{(e)}$. The mixture weights β are shared across samples, while each sample has its own feature perturbation.

Alternating updates. We initialize θ_L from the pretrained query scaling network and β at the uniform nominal vector $\bar{\beta}$. For each minibatch, we initialize each perturbed embedding at $z(x_i)$. Holding θ_L and β fixed, we take a normalized gradient ascent step on each embedding and project it onto its Euclidean ball of radius ϵ_1. We then increase the shared weights of sources with larger average losses and enforce the simplex and radius constraints on β. Finally, holding the perturbed embeddings and the updated β fixed, we update θ_L using their soft-label loss. The updates of θ_L and β continue across minibatches.

Algorithm 2 approximates the inner maximization with one feature step and one mixture step per model update, as used in our experiments. The operator $\Pi_{\mathbb{B}_i}$ projects onto $\mathbb{B}_i = \{u : \|u - z(x_i)\|_2 \le \epsilon_1\}$. For the mixture update, $\mathcal{P}_\beta$ first projects onto Δ_{M-1} and then contracts the result toward $\bar{\beta}$ if its distance exceeds ϵ_2. This preserves the simplex constraint while satisfying the radius constraint. The normalized gradient $\widehat{\nabla}$ has unit Euclidean norm for each sample; a zero gradient leaves its embedding unchanged.

Algorithm 2 DR-TFM (MS)

1: **Input:** training dataset $\mathcal{D}$; validation set; pretrained TabPFN f_{θ_0}; source construction rule; radii ϵ_1, ϵ_2; learning rates $\eta_{\theta_L}, \eta_\beta, \eta_z$; update steps T
2: Prepare the fixed context $\mathcal{D}^{(c)}$ and loss-evaluation set $\mathcal{D}^{(e)}$ as in Appendix A.2.1
3: Obtain frozen embeddings using $\mathcal{D}^{(c)}$ as context
4: Construct source subsets $\mathcal{S}_1, \dots, \mathcal{S}_M \subseteq \mathcal{D}^{(c)}$
5: Compute fixed source predictions $\widehat{P}^{(k)}_{Y|X}(\cdot \mid x_i) \leftarrow f_{\theta_0}(x_i; \mathcal{S}_k)$ for every $(x_i, y_i) \in \mathcal{D}^{(e)}$ and $k \in [M]$
6: Initialize $\beta^{(0)} \leftarrow \bar{\beta} = (1/M, \dots, 1/M)$ and $\theta_L^{(0)}$ from the pretrained query scaling network; freeze all other parameters
7: **for** $t = 0, \dots, T-1$ **do**
8: Draw minibatch indices B from $\mathcal{D}^{(e)}$ and set $z'_i \leftarrow z(x_i)$ for each $i \in B$
9: Form the soft labels $y^\circ(\beta^{(t)}, x_i) \leftarrow \sum_{k=1}^M \beta_k^{(t)} \widehat{P}^{(k)}_{Y|X}(\cdot \mid x_i)$
10: Ascend on each embedding and enforce its radius constraint, for $i \in B$:

$$z'_i \leftarrow \Pi_{\mathbb{B}_i}\Big(z'_i + \eta_z \widehat{\nabla}_{z'_i} \ell\big(f_{\theta^{(t)},Z}(z'_i), y^\circ(\beta^{(t)}, x_i)\big)\Big)$$

11: Ascend on the shared mixture weights and enforce their constraints:

$$\beta^{(t+1)} \leftarrow \mathcal{P}_\beta\big(\beta^{(t)} + \eta_\beta g\big),$$
$$g_k = \frac{1}{|B|}\sum_{i \in B} \ell\big(f_{\theta^{(t)},Z}(z'_i), \widehat{P}^{(k)}_{Y|X}(\cdot \mid x_i)\big), \quad k \in [M].$$

12: Update θ_L, holding z'_i and $\beta^{(t+1)}$ fixed:

$$\theta_L^{(t+1)} \leftarrow \theta_L^{(t)} - \eta_{\theta_L} \nabla_{\theta_L} \frac{1}{|B|}\sum_{i \in B} \ell\big(f_{\theta,Z}(z'_i), y^\circ(\beta^{(t+1)}, x_i)\big)\Big|_{\theta_L = \theta_L^{(t)}}$$

13: **end for**
14: Select an iterate using validation, as in Appendix A.2.1
15: **Output:** selected adapted TabPFN and fixed context $\mathcal{D}^{(c)}$

A.2.4 Query scaling architecture

We describe the architecture of the scaling function in (6). Let g_{θ_L} denote the small neural network used to construct s_{θ_L}. It takes a projected query $u \in \mathbb{R}^r$ as input and returns a vector of the same dimension. The network consists of two linear layers, each including a bias, with a GELU activation between them:

$$\mathbb{R}^r \xrightarrow{\text{Linear}} \mathbb{R}^{64} \xrightarrow{\text{GELU}} \mathbb{R}^{64} \xrightarrow{\text{Linear}} \mathbb{R}^r.$$

Equivalently,

$$g_{\theta_L}(u) = W_2\,\mathrm{GELU}(W_1 u + a_1) + a_2,$$

where $W_1 \in \mathbb{R}^{64\times r}$, $a_1 \in \mathbb{R}^{64}$, $W_2 \in \mathbb{R}^{r\times 64}$, and $a_2 \in \mathbb{R}^r$. In the TabPFN classification model used in our main experiments, $r = 64$ and the decoder has six attention heads. The same network is applied separately to each head's query vector.

The network output is converted into a coordinate-wise modulation through $\mathbf{1} + \tanh(g_{\theta_L}(u))$. This modulation multiplies the decoder's existing base scaling. As in the main text, we omit the head index and write $b(n_c) \in \mathbb{R}^r$ for the base scaling of one head, computed from $\log n_c$ by a separate frozen pretrained network. The combined scaling function is

$$s_{\theta_L}(u) = \frac{1}{\sqrt{r}}\, b(n_c) \odot \big[\mathbf{1} + \tanh(g_{\theta_L}(u))\big],$$

where $\mathbf{1}$ is the vector of ones and $\tanh$ is applied element-wise. The parameters of the base scaling network are kept fixed. Thus, for a fixed context size n_c, fine-tuning changes only the query-dependent modulation, while $b(n_c)$ remains unchanged. We suppress the dependence of s_{θ_L} on n_c

in the main text. We include the standard attention normalization factor $1/\sqrt{r}$ in s_{θ_L} for notational convenience.

In both simulated and real-data classification experiments with TabPFN, we update both linear layers of the query scaling network, with $\theta_L = (W_1, a_1, W_2, a_2)$. This gives 8,320 trainable parameters, approximately 0.016% of the pretrained model's parameters. The pretrained TabPFN feature map, the key and query projections, the base scaling network, and all other decoder parameters are kept fixed.

A.2.5 Illustration of query scaling

Figure 4 illustrates the effect of a uniform query scale. For fixed query–key scores, a smaller positive uniform scale produces a more diffuse attention distribution, whereas a larger scale concentrates attention on the context samples with the highest scores.

The learned scaling function in DR-TFM is coordinate-wise, so it can also change the relative query–key scores. Figure 3 examines the learned attention distributions empirically.

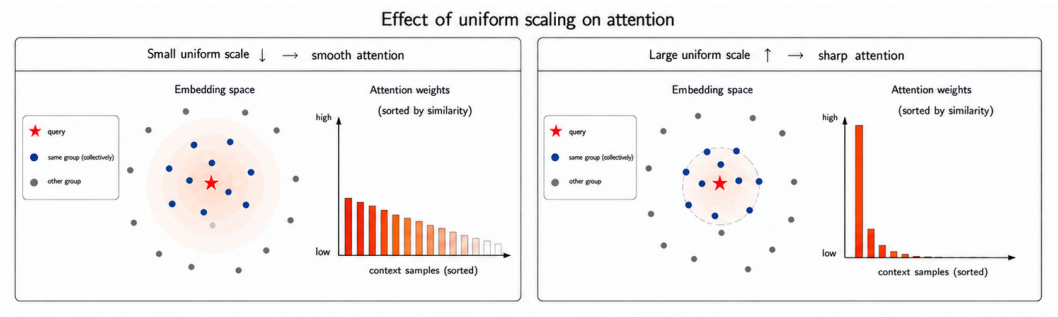

Figure 4: Schematic attention distributions under smaller and larger positive uniform query scales.

A.2.6 Regression

DR-TFM also applies to regression. The pretrained TabPFN regression model used here does not include a query-scaling network. We therefore fine-tune the first linear layer of its output network, keeping the pretrained TabPFN feature map and the second linear layer fixed. The objective uses squared-error loss in place of the classification loss.

Given a query embedding $z \in \mathbb{R}^{512}$, the regression output network produces logits for $B = 5{,}000$ response buckets. It consists of two linear layers, each including a bias, with a GELU activation between them:

$$\mathbb{R}^{512} \xrightarrow{\text{Linear (updated)}} \mathbb{R}^{1024} \xrightarrow{\text{GELU}} \mathbb{R}^{1024} \xrightarrow{\text{Linear (fixed)}} \mathbb{R}^{B}.$$

Equivalently,

$$t_\theta = W_2\,\mathrm{GELU}(W_1 z + a_1) + a_2, \qquad p_\theta = \mathrm{softmax}(t_\theta) \in \Delta_{B-1}.$$

The trainable parameters are $\theta_L = (W_1, a_1)$, with $W_1 \in \mathbb{R}^{1024\times 512}$ and $a_1 \in \mathbb{R}^{1024}$, while $W_2 \in \mathbb{R}^{B\times 1024}$ and $a_2 \in \mathbb{R}^B$ are fixed. This updates 525,312 parameters, approximately 0.90% of the pretrained regression model's 58.3 million parameters. The prediction is $\hat{y}_\theta = \sum_{b=1}^{B} p_{\theta,b}\mu_b$, where μ_b is the conditional mean of response bucket b. Figure 5 summarizes the architecture.

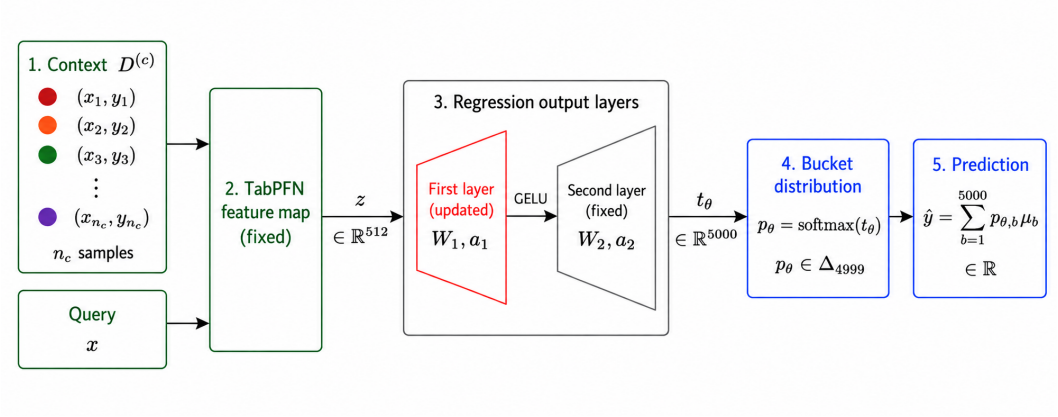

Figure 5: Regression architecture. Only the first linear layer of the output network is updated; the pretrained TabPFN feature map and the second linear layer remain fixed. Here μ_b denotes the conditional mean of response bucket b.

Regression simulation. We draw $A \sim \mathrm{Bernoulli}(1/2)$ and generate

$$Y \mid A \sim \mathcal{N}\big(m_p(2A-1), 0.6^2\big), \quad X_1 \mid A \sim \mathcal{N}\big(0.8(2A-1), 0.5^2\big),$$
$$X_2 = Y + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, 0.8^2),$$

where Y and X_1 are conditionally independent given A, and ε is independent of (A, Y, X_1). We set $m_p = 0.6\Phi^{-1}(p)$, where Φ is the standard normal cumulative distribution function. The four evaluation groups are defined by $(A, \mathbf{1}\{Y \geq 0\})$, with probabilities $p/2$, $(1-p)/2$, $(1-p)/2$, and $p/2$. We vary the training and validation value of p over $\{0.5, 0.6, 0.7, 0.8, 0.9\}$ and fix $p = 0.5$ at test time. In this regression design, changing p changes both the group proportions and the distributions within groups.

Each run uses 2,000 training, 1,000 validation, and 2,000 test samples. We estimate four groups by fitting a two-component GMM to the input features separately within each response-sign stratum, without using A. Figure 6 reports mean squared error (MSE) and mean absolute error (MAE), averaged over 50 runs. Mean denotes the average of the four group errors, and Worst group denotes their maximum. Consistent with the classification results, increasing training imbalance leads to a larger deterioration in TabPFN's worst-group performance than in its mean performance. DR-TFM (GR; GMM) mitigates this deterioration in both MSE and MAE.

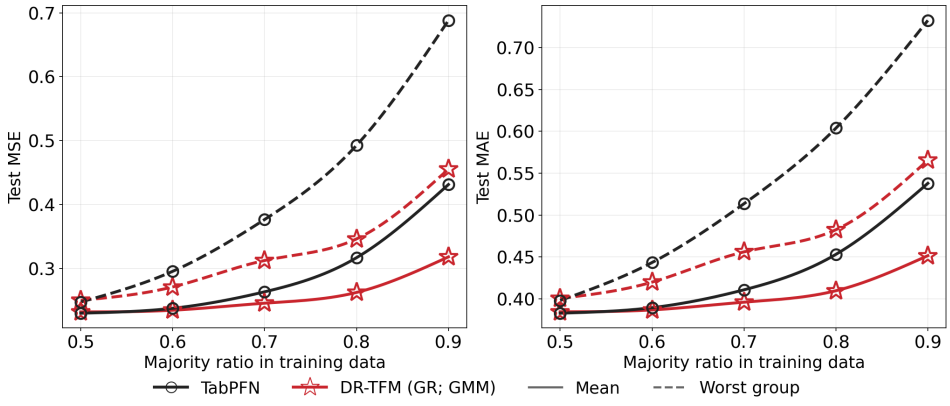

Figure 6: Regression results as the proportion of majority-group samples in training increases. Panels show test MSE (left) and MAE (right). Solid and dashed curves denote the mean and maximum group errors, respectively.

A.3 Experimental setup

We describe the simulation and real-data settings, evaluation metrics, baseline methods, hyperparameter selection, and an ablation on the number of clusters. The real-data benchmark and evaluation protocol follow Tong et al. (2025).

A.3.1 Simulation details

Data generation and groups. We generate each sample by first drawing a binary attribute $A \sim \text{Bernoulli}(1/2)$, then setting $Y = A$ with probability p and $Y = 1 - A$ otherwise. Conditional on (A, Y), we independently draw the two features as

$$X_1 \mid A \sim \mathcal{N}\big(\delta_a(2A-1), \sigma_a^2\big)\,, \qquad X_2 \mid Y \sim \mathcal{N}\big(\delta_y(2Y-1), \sigma_y^2\big)\,.$$

For the attention analysis in Figure 3, we use $\delta_a = 1.0$, $\sigma_a = 0.5$, $\delta_y = 0.5$, and $\sigma_y = 0.6$. The four groups are $G_1 = (A = 0, Y = 0)$, $G_2 = (A = 1, Y = 0)$, $G_3 = (A = 0, Y = 1)$, and $G_4 = (A = 1, Y = 1)$, with probabilities $p/2$, $(1-p)/2$, $(1-p)/2$, and $p/2$, respectively. The class probabilities remain balanced for every p; the imbalance is across groups rather than labels.

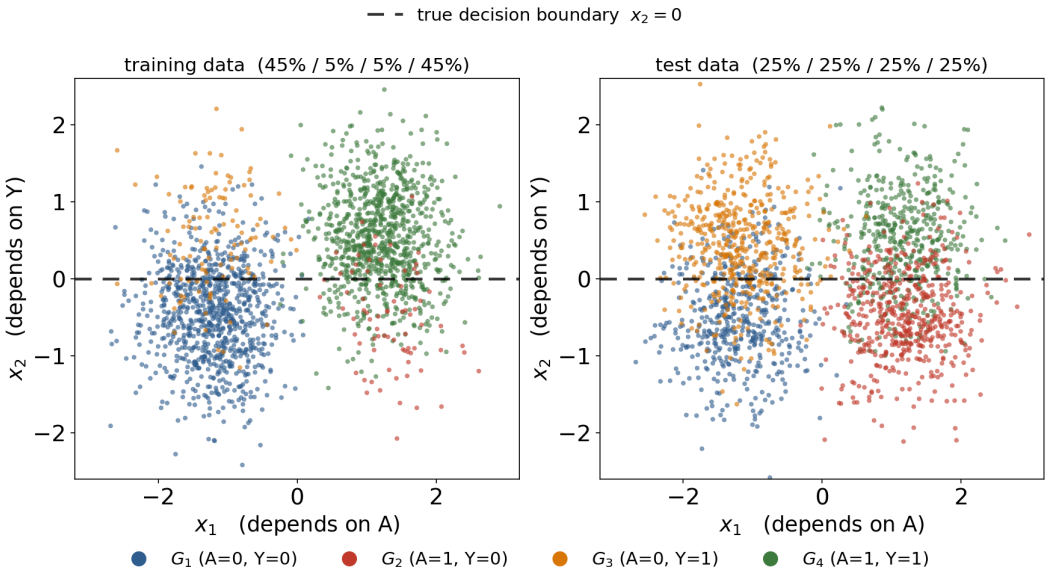

Figure 7: Illustration of the four-group simulation design. Colors identify the four (A, Y) groups. Their probabilities are 45%, 5%, 5%, and 45% in training (left), and 25% each at test time (right). The dashed line shows the Bayes-optimal boundary $x_2 = 0$ for the balanced test distribution, displayed in both panels for reference.

Sample sizes and subpopulation shift. We independently generate $2{,}000$ training samples and $1{,}000$ validation samples with $p = 0.9$, and $2{,}000$ test samples with $p = 0.5$. Training therefore contains an expected 900 samples from each majority group (G_1, G_4) and 100 from each minority group (G_2, G_3). The corresponding expected counts are 450 and 50 in validation, and 500 per group at test time. Realized counts vary across random draws. Only p changes between training and test; the feature distribution within each group remains fixed. Consequently, X_1 is associated with Y through A in training but is independent of Y under the balanced test distribution. Figure 7 illustrates this construction.

For Figure 1, we use the same data-generating parameters and training, validation, and test set sizes. We vary the training and validation value of p over $\{0.5, 0.6, 0.7, 0.8, 0.9\}$, keeping the test value fixed at 0.5. The reported curves average results over 50 runs.

Group estimation and adaptation. We fit a two-component GMM to the input features within each class, fixing the number of estimated groups at $G = 4$. We update both linear layers of

the query scaling network, totaling $8{,}320$ trainable parameters (Appendix A.2.4). We use AdamW with learning rate 10^{-3}, weight decay 10^{-4}, and at most 500 updates. We evaluate worst-group validation accuracy over the estimated groups every 10 updates and use it for model selection and early stopping, with patience 10. Test samples are used only for final evaluation.

For the attention analysis in Figure 3, the training set serves as the fixed labeled context $\mathcal{D}^{(c)}$. We independently draw an additional $2{,}000$ labeled samples from the same training distribution to form $\mathcal{D}^{(e)}$. The context and loss-evaluation sets are disjoint. The GMMs are fitted only on the context and assign estimated groups to loss-evaluation and validation samples within each response class.

Labels and the decision boundary. The labels are generated by the stochastic rule above, not by deterministically thresholding the observed features. Under the balanced test distribution, X_1 carries no label information, and the two classes have equal prior probabilities and equal variances along X_2. The Bayes-optimal classifier for test classification error is therefore

$$f^{\star}_{\mathrm{te}}(x) = \mathbf{1}\{x_2 > 0\},$$

with decision boundary $x_2 = 0$. The Gaussian noise produces overlap between classes, so a sample's label need not agree with the side of this boundary on which it falls. This is the test-optimal boundary; the optimal training classifier can also use X_1 because of the training association between A and Y.

Attention computation. For Figure 3, a test sample's own group is determined by its true (A, Y) group defined above. For each attention head and ensemble member, we sum the attention weights assigned to context samples in that same group. We then average these sums across heads and ensemble members to obtain one value per test sample. The left histogram pools these values for test samples in G_1 and G_4, while the right histogram pools them for test samples in G_2 and G_3.

A.3.2 DATASETS

Benchmarks.

- Adult (Becker & Kohavi, 1996). Census records. The task is to predict whether a person's annual income exceeds \$50K.
- Bank (Moro et al., 2014). Marketing calls made by a Portuguese bank. The task is to predict whether the client subscribes to a term deposit.
- Default (Yeh, 2009). Credit card clients in Taiwan. The task is to predict whether the client defaults in the following month.
- Shoppers (Sakar & Kastro, 2018). Online browsing sessions. The task is to predict whether the session ends in a purchase. About 85% of sessions do not, so the label itself is heavily imbalanced.
- Taxi (Navas, 2017). Taxi rides in Mexico City. The task is to predict whether the ride lasts more than thirty minutes.
- ACS Income (Ding et al., 2021). US Census records, with one table per state. The task is to predict whether income exceeds \$50K.

All six tasks are binary classification.

Transfer across states. ACS Income additionally evaluates transfer across states. We use Arizona (AZ), Massachusetts (MA), and Michigan (MI) in six settings. The three within-state settings train and test within one state, following the benchmark splits. The three transfer settings, AZ→MA, MA→MI and MI→AZ, train on one state and test on another. In a transfer setting the entire source table is split $80/20$ into training and validation, and the entire target table is used as the test set, so the target state contributes no training or validation data. Sizes for all six settings are given in Table 7.

Groups. Evaluation groups are defined by pairing a binary attribute with the label. During training and model selection, true group labels are used only by the methods explicitly marked as using them. The evaluated attributes are the ones selected in Appendix B.2 of Tong et al. (2025): marital status,

race, and sex for Adult; age, housing status, marital status, and last contact duration for Bank; age, sex, and the amount of the given credit for Default; traffic type, visitor type, and a weekend indicator for Shoppers; pickup month, a weekday indicator, and direction for Taxi; and race and sex for ACS Income. We use their dataset-specific binary encodings and average the metrics over these attributes, matching the reported baselines. Each attribute defines four (A, Y) groups; Appendix A.3.5 gives the metric formulas.

Data preparation. For the five tabular benchmarks and ACS Income, we follow the data preparation procedure of Tong et al. (2025), with the evaluated attributes specified above. Taxi and ACS Income use the WHYSHIFT preprocessing procedure (Liu et al., 2023) adopted in that benchmark. The resulting sizes are listed in Tables 6 and 7. Categorical columns are one-hot encoded for the MLP-based baselines.

Table 6: The five tabular benchmarks and their splits.

Dataset	Train	Validation	Test	Total
Adult	26,048	6,513	16,281	48,842
Bank	32,551	8,138	4,522	45,211
Default	21,600	5,400	3,000	30,000
Shoppers	8,877	2,220	1,233	12,330
Taxi	9,139	2,285	1,270	12,694

Table 7: ACS Income settings. The first three train and test on the same state; the last three transfer to a held-out state.

Setting	Train	Validation	Test	Total
AZ	23,959	5,990	3,328	33,277
MA	28,881	7,221	4,012	40,114
MI	36,005	9,002	5,001	50,008
AZ→MA	26,621	6,656	40,114	73,391
MA→MI	32,091	8,023	50,008	90,122
MI→AZ	40,006	10,002	33,277	83,285

A.3.3 BASELINES

We compare standard classifiers, TFMs and their adaptations, robust methods without true group labels, and methods that use true group labels. TabPFN, TabPFN-3.5, TabPFN (fine-tuned), and TabPFN (class-balanced) are counted as separate baselines. For this count, TabDPT with retrieved and full contexts is treated as one method, GEORGE with k-means and Gaussian mixture clustering as one method, and Fair-TabICL with group-balanced and uncertainty-based context selection as one method. Tables 8 and 10 summarize the training schedules and hyperparameter settings.

Standard methods.

- ERM-MLP. The classifier of Tong et al. (2025) with the reweighting removed, trained by empirical risk minimization.
- XGBoost (Chen & Guestrin, 2016). Gradient-boosted trees.
- CatBoost (Prokhorenkova et al., 2018). Gradient-boosted trees with ordered boosting and native handling of categorical columns.

TFMs.

- TabPFN (Grinsztajn et al., 2026). Predicts a query from a labeled context in a single forward pass, with no gradient update. This is the pretrained model we adapt.
- TabPFN-3.5 (Prior Labs, 2026). A TabPFN model with a shared pretrained checkpoint for classification and regression.

- TabICL (Qu et al., 2025). Scales in-context learning to larger tables through a two-stage encoder.
- TabFM (Google Research, 2026). Combines row and column attention with row compression and a separate transformer for in-context learning.
- EXAONE (Eo et al., 2026). Interleaves feature-axis and sample-axis attention through summary tokens for in-context learning.
- Causilo (Nums AI Inc., 2026). A compact in-context model that summarizes the context per feature, mixes features within each row, and predicts through a stack of attention layers in which queries read the labeled context.
- TabDPT (Ma et al., 2025). Pretrained on real rather than synthetic tables. We evaluate it with nearest-neighbor retrieval and with the full training set as context.
- TuneTables (Feuer et al., 2024). Replaces the context with a learned prompt, so the training set is compressed into a small number of tokens. Its prompt is tied to the input representation of TabPFN v1, which we use for this method. Unless otherwise specified, the other TabPFN-based methods use v3. Comparisons with TuneTables therefore reflect differences in both the pretrained model and the adaptation method.
- TabPFN (fine-tuned) (Grinsztajn et al., 2026). Adapts the pretrained transformer using the fine-tuning procedure, with context and loss-evaluation batches drawn from the training set. We use the standard fine-tuning settings, including an $80/20$ split into context and loss-evaluation sets.
- TabPFN (class-balanced). The same fine-tuning procedure and settings as TabPFN (fine-tuned), with the cross-entropy weighted by the inverse class frequency of the training labels:

$$\min_{\theta} \; \mathbb{E}_{\widehat{P}^{(e)}}\Big[w_Y \, \ell\Big(f_\theta(X; \mathcal{D}^{(c)}), Y\Big)\Big], \qquad w_c = \frac{1}{|\mathcal{Y}|\,\widehat{p}_c},$$

  where $\widehat{p}_c = n^{-1}\sum_{i=1}^{n} \mathbf{1}\{y_i = c\}$ is the empirical frequency of class c in the full training dataset $\mathcal{D}$, and w_Y is the weight for the observed class Y.
- MixturePFN (Xu et al., 2025). Partitions the training set with k-means, fine-tunes one expert per cluster on that cluster, and routes a test point to the expert of its nearest centroid. We adapt this procedure to the pretrained TabPFN model used in our experiments, following Liu & Ye (2025) with a context budget of 3000 samples and $\lceil n/3000 \rceil$ experts.
- LoCalPFN (Thomas et al., 2024). Replaces the full training set as context with the query's k nearest neighbors. We evaluate the retrieval variant (TabPFN-kNN) with $k = 1000$ and no parameter updates, using the same pretrained TabPFN model as our method.
- BETA (Liu & Ye, 2025). Trains a lightweight input adapter placed before the frozen pretrained TabPFN transformer. We use a two-layer MLP with 16 batch-ensemble members. Each member receives a bootstrapped context, and their predictions are averaged. We use the authors' settings with a context size of 1000 and the same pretrained TabPFN model as our method.
- DistPFN (Lee, 2026). Adjusts the pretrained TabPFN's class probabilities at test time using the ratio of their test-set average to the class prior of the context, without updating model parameters.

Robust methods without true group labels. For CVaR-DRO, χ^2-DRO, KL-DRO, EIIL, JTT, FAM, SRDO, and LSR, we use the dataset-level accuracies and standard deviations across seeds reported by Tong et al. (2025), marked † in the accuracy tables, and compute the Avg. columns across the corresponding datasets or settings.

- CVaR-DRO and χ^2-DRO (Levy et al., 2020), and KL-DRO (Duchi & Namkoong, 2021). Optimize the worst risk over an f-divergence ball around the training distribution, without constructing groups.
- JTT (Liu et al., 2021). Trains once, then upweights the samples the first model misclassifies and trains again.
- EIIL (Creager et al., 2021). Infers environments that maximally violate an invariance criterion, then optimizes over them.
- FAM (Petzka et al., 2021; Zou et al., 2024). Regularizes the flatness of the minimum reached by training.

- SRDO (Shen et al., 2020). Reweights samples so that the covariates become decorrelated.
- LSR (Tong et al., 2025). Reweights samples using a score-based density estimate on a learned latent space. This is the method whose benchmark we adopt.
- GEORGE (Sohoni et al., 2020). Clusters the penultimate representation of a trained model within each class, then minimizes the GroupDRO objective over the resulting clusters. We evaluate both k-means and Gaussian mixture clustering.
- BPA (Seo et al., 2022). Clusters a trained representation and reweights each cluster by its inverse frequency and its running loss.
- SPARE (Yang et al., 2024). Clusters the model output early in training, when the spurious feature dominates, then resamples by inverse cluster size.

Methods using true group labels. Results for methods that use true group labels are reported separately in Tables 15 and 17. A superscript * denotes access to true group labels.

- GroupDRO (Sagawa et al., 2020). Minimizes the worst risk over the true groups.
- Fair-TabICL (Kenfack et al., 2026). Selects in-context examples using the group label, either to balance the groups or by uncertainty in predicting the attribute.

A.3.4 Group and source construction for DR-TFM

MS source construction. We use the MS-C and MS-R constructions described in Appendix A.2.3. Following Kim et al. (2026), we start with ten candidate sources on each dataset. Each MS-R source contains approximately $n_c/5$ context samples. The main tables report DR-TFM (MS-C) as DR-TFM (MS).

True groups for GR. On the five tabular benchmarks, DR-TFM (GR)* and GroupDRO* use the four groups formed by each binary attribute $A \in \mathcal{A}$ and the class label Y. This yields $4|\mathcal{A}|$ groups: twelve on Adult, Default, Shoppers, and Taxi, and sixteen on Bank. Each sample belongs to one group per attribute, so groups associated with different attributes may overlap. This per-attribute construction avoids sparse intersections across three or four attributes.

On ACS Income, both methods use the eight disjoint $(Y, \mathrm{sex}, \mathrm{race})$ groups. Each evaluation group is a union of two of these groups. These true-group variants require no cluster selection.

Sampling with true groups. To stratify the context and loss-evaluation split on the five tabular benchmarks, we use the four cells of Y paired with sex (Adult and Default), marital status (Bank), weekend status (Shoppers), or pickup month (Taxi). On ACS Income, we use the eight joint cells above. GR* and MS* split the training set $70/30$ into context and loss-evaluation sets, stratified by these cells. These sampling partitions are distinct from the overlapping groups used in the GR objective on the five tabular benchmarks.

MS sources with true groups. DR-TFM (MS)* forms sources from the joint attribute values, omitting Y: eight combinations on Adult, Default, Shoppers, and Taxi, sixteen on Bank, and four sex–race combinations on ACS Income. We retain a source only if its context contains both response classes. Model selection for MS* maximizes worst-group validation accuracy over the same groups used for GR*: the $4|\mathcal{A}|$ overlapping groups on the five tabular benchmarks and the eight joint groups on ACS Income.

A.3.5 Evaluation metrics

We report the evaluation metrics of Tong et al. (2025). Let $\mathcal{A}$ denote the evaluated attributes of a dataset. For an attribute $A \in \mathcal{A}$, the test set splits into the four groups

$$\mathcal{C}_A(v, c) = \big\{\, i \;:\; A_i = v,\; Y_i = c \,\big\}, \qquad (v, c) \in \{0, 1\}^2,$$

each with accuracy

$$\mathrm{acc}_A(v, c) = \frac{1}{|\mathcal{C}_A(v, c)|} \sum_{i \in \mathcal{C}_A(v,c)} \mathbf{1}\big\{\widehat{Y}_i = Y_i\big\}.$$

The two reported quantities are

$$\mathrm{Acc}_{\mathrm{worst}} = \frac{1}{|\mathcal{A}|} \sum_{A \in \mathcal{A}} \min_{(v,c) \in \{0,1\}^2} \mathrm{acc}_A(v, c), \qquad \mathrm{Acc}_{\mathrm{mean}} = \frac{1}{|\mathcal{A}|} \sum_{A \in \mathcal{A}} \frac{1}{4} \sum_{(v,c) \in \{0,1\}^2} \mathrm{acc}_A(v, c).$$

Here, $\mathrm{Acc}_{\mathrm{mean}}$ assigns equal weight to the four groups, while $\mathrm{Acc}_{\mathrm{worst}}$ measures the accuracy of the least accurate group for each attribute. Both metrics are averaged over attributes and reported as percentages. For summaries across datasets or settings, we report the average of mean group accuracies and the average of worst-group accuracies, giving each dataset or setting equal weight.

A.3.6 Training protocol

Group estimation and data splitting. The TabPFN-based real-data variants without true group labels use 512-dimensional pretrained embeddings and the preparation procedure in Appendix A.2.1. The true-group variants follow Appendix A.3.4.

Classifier architecture. The MLP-based baselines use the classifier architecture of Tong et al. (2025): a four-layer neural network $d \to 1024 \to 1024 \to 512 \to 2$ with LeakyReLU activations. These baselines use the one-hot-encoded original features, except for LSR, which retains its VAE-based latent representation.

Optimization. DR-TFM (GR) uses Adam and DR-TFM (MS) uses AdamW. ERM-MLP and GroupDRO use Adam, following Tong et al. (2025). Learning rates, batch sizes, and schedules are listed in Tables 8 and 10. Unless otherwise noted, results are averaged over seeds 0, 1, and 2.

Schedule. Table 8 lists the training budgets and model selection criteria. The DR-TFM variants without true group labels use early stopping based on worst-group validation accuracy over estimated groups. For MS-C and MS-R, we evaluate this metric every five parameter updates and stop after five consecutive validation checks without improvement. GR* and MS* use the true groups specified in Appendix A.3.4. Table 10 specifies the patience.

Table 8: Training budgets and model selection criteria for DR-TFM and the MLP-based baselines.

Method	Training	Epochs / steps	Selection criterion	Learning rate schedule
ERM-MLP	ERM, from scratch	1000	validation accuracy	multiply by 0.9 after 20 epochs without improvement in mean training loss
GroupDRO	GroupDRO over the true groups, from scratch	1000	validation worst-group accuracy (true groups)	same as above
GEORGE stage 1	ERM, from scratch (feature extractor)	1000	validation accuracy	none
GEORGE stage 2	GroupDRO over the estimated clusters, newly initialized model	300	worst-group accuracy over the estimated validation clusters	none
BPA	base ERM, then retraining with cluster reweighting	1000 / 300	worst-group accuracy over the estimated validation clusters	none
SPARE	base ERM, then retraining with importance sampling	1000 / 300	worst-group accuracy over the estimated validation clusters	none
LSR	original LSR training procedure	4000 / 4000 / 1000	lowest training loss	multiply by 0.9 after 20 epochs without improvement
DR-TFM	query scaling with a fixed TabPFN feature map	at most 1000 steps	validation worst-group accuracy over the estimated clusters	—

A.3.7 Model selection and hyperparameters

Selection rule. The following settings apply to the real-data experiments. We follow the original methods' hyperparameter settings and selection procedures where applicable. Group-based robust baselines select remaining hyperparameters using worst-group validation accuracy, evaluated on true groups when available and estimated groups otherwise. XGBoost and CatBoost use validation log-loss. Test data are used only for final evaluation.

Selected hyperparameters. Table 10 lists the search spaces. For SPARE, we select T_{init} using worst-group validation accuracy over estimated groups; the numbers of clusters of SPARE and BPA follow GEORGE. ERM-MLP uses the fixed hyperparameters of Tong et al. (2025), including zero weight decay.

Cluster selection for DR-TFM (GR). Following GEORGE (Sohoni et al., 2020), we select the number of clusters separately for each class over $G_c \in \{2, 4, 6, 8\}$ by the silhouette score. This selection uses neither true group labels nor validation data. After selection, validation samples are assigned to the nearest cluster center within their class.

For both clustering methods, this criterion selects two clusters per class ($G = 4$) across all datasets and three seeds reported in Tables 14 and 16.

Hyperparameters for DR-TFM (MS). For both source constructions in Appendix A.3.4, we fix $\epsilon_2 = 1$, following Kim et al. (2026). With the uniform nominal vector, this radius allows all convex mixtures of the source conditionals.

Kim et al. (2026) develop their method for unsupervised domain adaptation and also evaluate it under subpopulation shift. In their formulation, ϵ_1 controls a Wasserstein neighborhood around the empirical target input distribution, accounting for uncertainty when target data are limited. Our setting focuses on subpopulation shift and does not use target-domain data for adaptation or model selection. Here, ϵ_1 controls the size of perturbations to the loss-evaluation embeddings. Table 9 shows that worst-group validation accuracy on Adult varies little across the tested values of ϵ_1. Given this limited sensitivity, we select $\epsilon_1 = 0.2$ once using these validation results and keep it fixed across all datasets. Each model update uses one feature ascent step and one mixture-weight ascent step, with $\eta_z = \epsilon_1$ and $\eta_\beta = 0.05$.

Table 9: Worst-group validation accuracy of DR-TFM (MS) on Adult for selecting ϵ_1, with $\epsilon_2 = 1$ and ten initial candidate sources. Accuracy is evaluated over estimated groups. Results are averaged over three seeds; parentheses report standard deviations.

ϵ_1	0	0.2	0.4	0.6	0.8	1
Validation	70.43 (1.21)	70.44 (1.18)	70.44 (1.13)	70.38 (1.15)	70.37 (1.15)	70.39 (1.12)

GroupDRO updates and regularization. The GroupDRO updates use the group-size adjustment described in Appendix A.2.2, with an adversarial learning rate of 0.1 and $C = 5$. DR-TFM (GR) fixes weight decay at 5×10^{-3}. For GEORGE, BPA, SPARE, and GroupDRO, weight decay is selected for the robust training stage. The initial ERM stages of GEORGE, BPA, and SPARE retain zero weight decay, so their estimated partitions remain fixed across the search.

Table 10: Hyperparameters. The final column lists selected quantities; all other settings are fixed. LR denotes the model learning rate.

Method	η	C	LR	Batch size	Budget	Patience	Weight decay	Selected quantities
DR-TFM (GR; k-means)	0.1	5	0.01	full	1000	5	0.005	G_c per class by silhouette over $\{2, 4, 6, 8\}$
DR-TFM (GR; GMM)	0.1	5	0.01	full	1000	5	0.005	G_c per class by silhouette over $\{2, 4, 6, 8\}$
DR-TFM (GR)*	0.1	5	0.01	full	1000	5	0.005	none (groups given)
DR-TFM (MS)	—	—	0.001	1024	300	5	10^{-4}	ϵ_1 once on Adult; fixed thereafter
DR-TFM (MS)*	—	—	0.001	1024	300	5	10^{-4}	none (groups given)
ERM-MLP	—	—	0.001	4096	1000	20	0	none
GroupDRO	0.1	5	0.001	4096	1000	20	$\{10^{-4}, 5\times10^{-3}, 10^{-2}\}$	weight decay
GEORGE (k-means / GMM)	0.1	5	0.001	4096	1000 + 300 epochs	20	$\{10^{-4}, 5\times10^{-3}, 10^{-2}\}$	G_{cl} by silhouette over $\{2, 4, 6, 8\}$; weight decay
SPARE	0.1	5	0.001	4096	1000 + 300 epochs	20	$\{10^{-4}, 5\times10^{-3}, 10^{-2}\}$	T_{init}; weight decay
BPA	0.1	5	0.001	4096	1000 + 300 epochs	20	$\{10^{-4}, 5\times10^{-3}, 10^{-2}\}$	weight decay; clusters as GEORGE
XGBoost	—	—	$\{0.03, 0.05, 0.1\}$	—	≤ 600	—	—	depth, learning rate, tree count
CatBoost	—	—	$\{0.03, 0.05, 0.1\}$	—	≤ 600	—	—	depth, learning rate, tree count
TabICL	—	—	—	—	—	—	—	none
TabDPT	—	—	—	256	—	—	—	none

Table 11: Additional baseline settings.

Method	Trained component	Settings
TabPFN (fine-tuned)	Full backbone	Learning rate 10^{-5}, weight decay 0.02, 30 epochs, patience 8
TabPFN (class-balanced)	Full backbone	As TabPFN (fine-tuned), with inverse-class-frequency weights in the cross-entropy
BETA	Input adapter	16 ensemble members, learning rate 0.003, weight decay 0.02, batch size 1024, 30 epochs, patience 5
MixturePFN	Expert models	Context size 3000; experts use the TabPFN fine-tuning settings above
LoCalPFN	None	Retrieval context of 1000 neighbors per query batch
TuneTables	Soft prompt	10 tokens, ≤ 31 epochs, patience 5, TabPFN v1
Fair-TabICL	None	Context balanced by group or selected by predictive uncertainty
TabICL / TabDPT	None	TabDPT inference batch size 256
TabFM	None	Full training context; 32 ensemble members
EXAONE	None	Full training context; 8 ensemble members
XGBoost / CatBoost	Trees	Tree counts $\{25, 50, 100, 200, 300, 400, 600\}$, selected by minimum validation log loss

A.3.8 Ablation on the number of clusters

We vary the number of clusters per class, denoted by G_{cl}, over $\{2, 4, 6, 8\}$ and compare DR-TFM (GR) with GEORGE using class-wise k-means and GMM on the five tabular benchmarks and three within-state ACS Income settings. The ablation isolates G_{cl}, so every remaining hyperparameter is held fixed and both methods use a weight decay of 5×10^{-3}. For this ablation, each class has $G_c = G_{\text{cl}}$ clusters, giving $G = 2G_{\text{cl}}$ estimated groups. Tables 12 and 13 report both clustering methods. Their standard deviations are computed across seeds after averaging accuracies over datasets or within-state settings.

Results. DR-TFM (GR) outperforms GEORGE in both metrics at every tested value of G_{cl} for both clustering methods. With k-means, its gains in the average of worst-group accuracies range from 8.77 to 24.41 pp on the five tabular benchmarks and from 4.91 to 25.66 pp on ACS within-state settings. Compared with GEORGE, DR-TFM (GR) shows substantially more stable performance as the number of clusters varies, with smaller ranges in both metrics for both clustering methods.

Table 12: Ablation on G_{cl} for the five tabular benchmarks, averaged over datasets and three seeds. Parentheses report standard deviations across seeds of the dataset-averaged accuracies. Δ is DR-TFM (GR) minus GEORGE. Range is the maximum minus minimum within each row across the four values of G_{cl}. Ranges are computed before rounding.

Method	$G_{\text{cl}} = 2$		$G_{\text{cl}} = 4$		$G_{\text{cl}} = 6$		$G_{\text{cl}} = 8$		Range (pp)	
	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst
DR-TFM (GR; k-means)	79.31 (0.17)	72.57 (0.64)	77.42 (0.33)	68.37 (1.29)	75.97 (0.63)	63.90 (3.00)	75.09 (0.77)	61.39 (1.12)	4.22	11.18
GEORGE (k-means)	73.79 (0.48)	63.80 (1.21)	61.80 (2.28)	43.96 (4.56)	64.22 (1.04)	48.25 (2.78)	58.64 (5.89)	44.45 (9.45)	15.15	19.84
Δ (pp)	**+5.52**	**+8.77**	**+15.62**	**+24.41**	**+11.75**	**+15.65**	**+16.45**	**+16.94**	10.93	15.64
DR-TFM (GR; GMM)	79.19 (0.15)	71.81 (0.38)	77.12 (1.08)	67.71 (1.29)	75.99 (0.90)	64.09 (2.26)	74.57 (0.38)	59.67 (3.08)	4.62	12.13
GEORGE (GMM)	73.75 (1.18)	65.26 (1.66)	68.68 (1.37)	59.69 (0.56)	63.40 (1.66)	49.30 (4.99)	62.26 (2.29)	46.10 (8.85)	11.49	19.16
Δ (pp)	**+5.44**	**+6.55**	**+8.44**	**+8.02**	**+12.59**	**+14.79**	**+12.31**	**+13.57**	7.15	8.24

Table 13: Ablation on G_{cl} for ACS Income, averaged over the three within-state settings and three seeds. Parentheses report standard deviations across seeds of the setting-averaged accuracies. Δ is DR-TFM (GR) minus GEORGE. Range is the maximum minus minimum within each row across the four values of G_{cl}. Ranges are computed before rounding.

Method	$G_{\mathrm{cl}} = 2$		$G_{\mathrm{cl}} = 4$		$G_{\mathrm{cl}} = 6$		$G_{\mathrm{cl}} = 8$		Range (pp)	
	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst
DR-TFM (GR; k-means)	79.21 (0.14)	71.06 (0.12)	80.00 (0.12)	72.99 (0.40)	79.53 (0.38)	72.22 (1.11)	79.70 (0.04)	72.59 (0.39)	0.78	1.93
GEORGE (k-means)	74.36 (3.03)	66.15 (2.31)	64.70 (1.35)	54.67 (4.89)	65.41 (2.71)	56.41 (1.27)	66.75 (6.77)	46.94 (9.64)	9.66	19.22
Δ (pp)	**+4.85**	**+4.91**	**+15.29**	**+18.32**	**+14.11**	**+15.81**	**+12.95**	**+25.66**	10.44	20.75
DR-TFM (GR; GMM)	79.82 (0.16)	72.17 (0.82)	79.96 (0.24)	72.86 (0.38)	79.42 (0.47)	71.85 (1.18)	79.69 (0.43)	72.32 (0.47)	0.54	1.01
GEORGE (GMM)	64.98 (2.71)	59.52 (2.97)	63.69 (1.40)	56.20 (2.65)	64.28 (2.39)	55.68 (3.18)	65.08 (3.25)	55.01 (1.81)	1.40	4.51
Δ (pp)	**+14.84**	**+12.65**	**+16.27**	**+16.66**	**+15.13**	**+16.18**	**+14.61**	**+17.31**	1.66	4.66

A.4 Full result tables

Method names follow Table 1; MS-C and MS-R denote clustered and randomly sampled sources, respectively. Tables 14–17 report the full results, grouped by method family and access to true group labels. Detailed results for the additional TFMs are reported separately in Appendix A.8. Results are averaged over three seeds. Values in parentheses are standard deviations over seeds where available. The Avg. columns report the average of mean group accuracies and the average of worst-group accuracies across the constituent datasets or settings. Parenthesized values in these columns are the averages of the per-dataset or per-setting standard deviations across seeds.

Comparisons with true group labels. The true-group protocols for Tables 15 and 17 are described in Appendices A.3.3 and A.3.4. On the five tabular benchmarks, DR-TFM (GR) with estimated groups achieves an average of worst-group accuracies of 72.57%, exceeding GroupDRO with true groups (71.79%) and coming within 1.09 pp of DR-TFM (GR) with true groups (73.66%).

The recovery with true groups under the same query-scaling architecture suggests that the degradation on MA→MI may reflect a mismatch between the estimated groups used for adaptation and model selection and the target evaluation groups (Tables 16 and 17).

Table 14: Results on the five tabular benchmarks for methods without true group labels. Avg. denotes the average of each metric across datasets. Values in parentheses are standard deviations over three seeds. Accuracy results marked †, including standard deviations, are taken from Tong et al. (2025). Bold method names indicate our variants.

Methods	Adult		Bank		Default		Shoppers		Taxi		Avg.	
	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst
Standard methods												
ERM-MLP	73.45 (0.16)	45.29 (0.90)	71.86 (1.80)	42.03 (4.18)	63.70 (0.48)	28.52 (0.21)	76.44 (0.47)	50.11 (1.85)	75.27 (0.60)	57.37 (1.44)	72.14 (0.70)	44.67 (1.72)
XGBoost	77.09 (0.17)	52.58 (0.48)	68.69 (0.31)	33.87 (0.92)	64.78 (0.11)	30.72 (0.13)	78.03 (0.19)	55.42 (0.24)	75.81 (0.21)	57.57 (0.16)	72.88 (0.20)	46.03 (0.39)
CatBoost	76.50 (0.22)	51.33 (0.73)	68.94 (0.49)	34.43 (0.65)	64.75 (0.05)	31.03 (0.00)	77.50 (0.62)	54.24 (2.30)	76.04 (0.07)	58.71 (0.54)	72.75 (0.29)	45.95 (0.84)
TFMs												
TabICL	76.40 (0.66)	51.17 (1.26)	71.90 (0.18)	40.38 (0.54)	64.70 (0.09)	31.48 (0.19)	78.71 (0.05)	56.70 (0.00)	76.07 (0.18)	57.86 (0.35)	73.56 (0.23)	47.52 (0.47)
TabDPT (retrieval)	74.13 (0.22)	46.80 (0.66)	70.06 (0.25)	37.47 (1.12)	64.25 (0.18)	30.18 (0.30)	78.70 (0.35)	58.11 (1.07)	75.90 (0.12)	58.44 (0.20)	72.61 (0.23)	46.20 (0.67)
TabDPT (full context)	72.58 (1.46)	41.77 (4.91)	62.28 (2.03)	23.68 (4.62)	63.54 (2.63)	28.96 (7.04)	77.36 (1.09)	52.98 (5.21)	75.39 (0.28)	57.44 (1.06)	70.23 (1.50)	40.97 (4.57)
TuneTables	68.98 (1.31)	34.03 (2.61)	64.61 (2.28)	27.84 (4.18)	60.84 (2.42)	21.53 (6.60)	77.64 (1.28)	53.82 (2.99)	71.51 (2.07)	46.10 (5.42)	68.72 (1.87)	36.66 (4.36)

Continued on next page

Table 14: Five tabular benchmarks, methods without true group labels (continued).

Methods	Adult		Bank		Default		Shoppers		Taxi		Avg.	
	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst
MixturePFN	74.17 (0.59)	46.00 (1.67)	70.32 (0.84)	38.24 (2.03)	64.16 (0.17)	30.55 (0.37)	78.60 (1.62)	55.82 (4.95)	75.54 (0.28)	57.51 (0.63)	72.56 (0.70)	45.62 (1.93)
LoCalPFN (retrieval)	75.61 (0.14)	49.34 (0.39)	71.75 (0.17)	40.86 (0.42)	64.45 (0.15)	31.13 (0.14)	79.24 (0.38)	58.11 (1.11)	75.72 (0.10)	57.57 (0.29)	73.35 (0.19)	47.40 (0.47)
BETA	74.70 (0.35)	47.36 (0.87)	66.26 (1.28)	30.51 (2.49)	64.46 (0.12)	30.93 (0.72)	78.44 (0.54)	56.84 (2.08)	75.03 (0.05)	57.14 (0.25)	71.78 (0.47)	44.56 (1.28)
DistPFN	75.02 (0.17)	48.37 (0.59)	72.63 (0.10)	42.04 (0.19)	64.61 (0.05)	30.71 (0.05)	78.00 (0.13)	54.63 (0.60)	75.76 (0.13)	57.26 (0.44)	73.20 (0.12)	46.60 (0.37)
Robust methods												
CVaR-DRO†	71.95 (0.68)	49.02 (2.63)	68.93 (1.52)	38.71 (3.59)	62.43 (0.33)	34.60 (2.44)	76.65 (0.31)	51.26 (2.01)	65.55 (2.00)	54.51 (5.93)	69.10 (0.97)	45.62 (3.32)
χ^2-DRO†	71.85 (0.73)	50.09 (1.78)	70.88 (0.81)	40.04 (3.56)	62.03 (0.19)	34.16 (2.12)	76.95 (1.72)	52.40 (3.60)	68.27 (0.99)	62.43 (1.08)	70.00 (0.89)	47.82 (2.43)
KL-DRO†	74.33 (0.14)	49.17 (0.90)	69.95 (0.60)	39.89 (1.75)	59.03 (3.25)	19.39 (5.10)	79.53 (1.08)	59.74 (3.98)	62.30 (8.96)	47.93 (23.68)	69.03 (2.81)	43.22 (7.08)
EIIL†	69.37 (2.50)	38.97 (8.15)	61.85 (2.49)	21.69 (6.91)	65.05 (1.06)	28.91 (7.55)	74.82 (7.04)	46.18 (15.02)	69.52 (0.07)	58.00 (4.43)	68.12 (2.63)	38.75 (8.41)
JTT†	71.46 (1.89)	49.93 (3.65)	68.78 (0.67)	37.77 (1.50)	62.47 (0.90)	35.51 (1.58)	78.10 (0.75)	52.59 (3.35)	67.47 (0.42)	60.01 (2.60)	69.66 (0.93)	47.16 (2.54)
FAM†	72.85 (1.65)	49.59 (1.95)	71.03 (3.89)	41.09 (5.25)	62.50 (0.19)	37.12 (2.07)	76.60 (0.09)	53.51 (2.05)	67.90 (0.75)	62.24 (0.63)	70.18 (1.31)	48.71 (2.39)
SRDO†	71.27 (2.43)	46.44 (6.82)	66.34 (2.55)	32.08 (5.44)	62.38 (8.06)	33.34 (10.94)	76.24 (0.35)	53.11 (2.31)	64.57 (2.07)	58.18 (2.50)	68.16 (3.09)	44.63 (5.60)
LSR†	74.33 (0.28)	54.79 (2.01)	69.50 (0.39)	41.28 (1.56)	62.62 (0.12)	38.78 (1.22)	79.68 (1.44)	60.73 (3.39)	67.85 (0.16)	63.14 (2.02)	70.80 (0.48)	51.74 (2.04)
GEORGE (k-means)	77.48 (0.42)	60.18 (2.37)	81.72 (0.54)	73.32 (2.36)	62.73 (2.89)	56.60 (3.62)	77.47 (10.40)	71.56 (14.47)	64.03 (2.31)	52.82 (6.26)	72.69 (3.31)	62.89 (5.82)
GEORGE (GMM)	76.67 (0.37)	57.68 (3.46)	70.44 (3.73)	58.98 (4.11)	64.95 (1.24)	59.42 (1.80)	75.14 (6.57)	69.90 (6.98)	75.30 (0.34)	71.95 (0.63)	72.50 (2.45)	63.59 (3.40)
BPA	64.21 (2.69)	54.34 (4.11)	65.52 (6.83)	54.04 (6.41)	57.45 (1.05)	51.97 (1.95)	62.88 (1.96)	50.18 (3.19)	59.96 (1.32)	50.19 (3.25)	62.00 (2.77)	52.14 (3.78)
SPARE	69.81 (0.84)	46.64 (2.84)	76.34 (1.41)	57.52 (3.99)	57.68 (5.48)	34.70 (15.51)	79.93 (0.87)	69.67 (5.36)	66.08 (8.24)	51.24 (9.91)	69.97 (3.37)	51.95 (7.52)
TFM adaptation												
TabPFN	75.33 (0.05)	49.17 (0.28)	73.07 (0.31)	42.96 (0.76)	64.77 (0.04)	31.09 (0.04)	79.59 (0.32)	58.29 (0.52)	76.01 (0.25)	58.19 (0.53)	73.75 (0.19)	47.94 (0.43)
TabPFN (fine-tuned)	76.33 (0.35)	50.80 (0.47)	73.10 (0.02)	42.99 (0.21)	65.03 (0.40)	31.71 (2.04)	78.88 (0.76)	56.25 (1.98)	76.11 (0.27)	58.23 (0.66)	73.89 (0.36)	48.00 (1.07)
TabPFN (class-balanced)	75.92 (0.40)	50.98 (1.14)	73.22 (0.51)	43.42 (1.42)	65.27 (0.60)	32.76 (1.73)	77.79 (2.89)	53.24 (8.24)	76.48 (0.96)	61.37 (5.68)	73.74 (1.07)	48.35 (3.64)
DR-TFM (GR; k-means)	79.64 (0.20)	**67.90** (1.61)	85.01 (0.36)	**80.14** (0.53)	69.42 (0.62)	60.37 (1.70)	85.79 (0.32)	81.66 (0.50)	76.69 (0.19)	**72.79** (0.62)	**79.31** (0.34)	**72.57** (0.99)
DR-TFM (GR; GMM)	79.10 (0.16)	65.58 (1.47)	84.74 (0.40)	79.12 (1.03)	**69.80** (0.37)	**60.45** (1.66)	**85.86** (0.32)	**82.40** (1.77)	76.45 (0.28)	71.48 (1.06)	79.19 (0.31)	71.81 (1.40)
DR-TFM (MS-C)	**79.78** (0.24)	67.41 (0.64)	**85.02** (1.33)	78.41 (1.49)	69.18 (0.52)	57.86 (4.76)	84.48 (0.35)	77.09 (0.66)	77.13 (0.13)	65.80 (0.70)	79.12 (0.51)	69.31 (1.65)
DR-TFM (MS-R)	79.70 (0.05)	66.87 (0.32)	82.99 (1.52)	75.71 (3.24)	69.05 (0.49)	57.72 (5.56)	84.97 (0.73)	78.35 (1.21)	**77.37** (0.52)	67.91 (1.79)	78.82 (0.66)	69.31 (2.42)

Table 15: Results on the five tabular benchmarks for methods using true group labels. Avg. denotes the average of each metric across datasets. Values in parentheses are standard deviations over three seeds. Bold method names indicate our variants.

Methods	Adult		Bank		Default		Shoppers		Taxi		Avg.	
	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst
Robust methods												
Fair-TabICL (group-balanced)	**80.11** (0.23)	72.34 (0.91)	**86.58** (0.09)	78.54 (0.10)	**69.86** (0.23)	57.73 (0.71)	**86.24** (0.85)	**83.38** (0.91)	**77.45** (0.32)	68.03 (0.49)	**80.05** (0.34)	72.00 (0.62)
Fair-TabICL (uncertainty)	61.80 (0.74)	21.35 (1.65)	71.52 (0.14)	40.51 (0.32)	64.93 (0.12)	31.18 (0.13)	77.80 (0.43)	52.81 (1.57)	75.94 (0.14)	57.61 (0.52)	70.40 (0.31)	40.69 (0.84)
GroupDRO	78.16 (0.22)	**73.90** (0.69)	81.56 (0.57)	75.47 (1.25)	67.16 (0.45)	61.82 (1.09)	80.30 (1.26)	76.28 (2.80)	75.21 (0.04)	**71.47** (1.26)	76.48 (0.51)	71.79 (1.42)
TFM adaptation												
DR-TFM (GR)	78.36 (0.52)	72.26 (0.79)	86.12 (0.93)	**81.94** (0.34)	69.58 (0.26)	**65.30** (1.19)	85.21 (0.39)	80.52 (2.17)	77.03 (0.39)	68.28 (2.32)	79.26 (0.50)	**73.66** (1.36)
DR-TFM (MS)	78.86 (1.55)	72.32 (1.08)	86.56 (0.39)	77.72 (0.63)	69.85 (0.39)	54.71 (1.70)	85.09 (0.99)	79.71 (3.23)	**77.45** (0.27)	68.48 (0.96)	79.56 (0.72)	70.59 (1.52)

Table 16: ACS Income results for methods without true group labels. The left block reports within-state results and the right block reports transfers between states. Avg. denotes the average of each metric across the three settings in its block. Values in parentheses are standard deviations over three seeds. Accuracy results marked †, including standard deviations, are taken from Tong et al. (2025). Bold method names indicate our variants.

Methods	AZ		MA		MI		Avg.		AZ → MA		MA → MI		MI → AZ		Avg.	
	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst
Standard methods																
ERM-MLP	77.20 (0.54)	59.42 (1.59)	80.82 (0.07)	73.82 (0.61)	75.69 (0.32)	57.99 (1.88)	77.90 (0.31)	63.74 (1.36)	76.82 (0.62)	60.23 (2.91)	79.13 (0.09)	71.34 (0.66)	75.56 (0.30)	56.31 (2.40)	77.17 (0.34)	62.63 (1.99)
XGBoost	77.64 (0.27)	58.99 (0.76)	82.26 (0.25)	76.38 (0.12)	77.50 (0.11)	62.77 (0.40)	79.13 (0.21)	66.05 (0.43)	77.96 (0.14)	61.99 (0.36)	79.67 (0.06)	72.01 (0.13)	76.82 (0.10)	58.99 (0.30)	78.15 (0.10)	64.33 (0.27)
CatBoost	77.35 (0.20)	58.26 (0.60)	81.97 (0.20)	75.59 (0.19)	77.27 (0.12)	62.43 (0.13)	78.86 (0.17)	65.43 (0.30)	77.66 (0.14)	61.15 (0.25)	79.60 (0.05)	72.31 (0.42)	76.68 (0.09)	58.71 (0.04)	77.98 (0.09)	64.06 (0.23)
TFMs																
TabICL	78.14 (0.09)	59.46 (0.21)	82.89 (0.03)	76.50 (0.33)	77.84 (0.23)	64.30 (0.38)	79.62 (0.12)	66.75 (0.31)	78.03 (0.07)	62.08 (0.19)	**79.73** (0.06)	71.78 (0.17)	76.98 (0.05)	59.69 (0.22)	78.25 (0.06)	64.52 (0.19)
TabDPT (retrieval)	77.25 (0.33)	57.85 (0.58)	81.39 (0.25)	75.43 (0.24)	76.80 (0.20)	61.13 (0.52)	78.48 (0.26)	64.80 (0.44)	76.92 (0.03)	59.51 (0.20)	78.89 (0.04)	70.70 (0.19)	75.21 (0.06)	54.67 (0.22)	77.01 (0.04)	61.63 (0.21)
TabDPT (full context)	74.13 (0.41)	50.99 (1.71)	79.51 (0.85)	72.34 (3.16)	74.95 (0.58)	58.01 (3.56)	76.20 (0.61)	60.44 (2.81)	75.89 (0.73)	58.09 (2.07)	77.70 (0.33)	71.51 (2.02)	73.51 (0.47)	51.16 (2.25)	75.70 (0.51)	60.25 (2.11)
TuneTables	74.34 (1.02)	52.56 (3.03)	78.13 (0.29)	70.04 (1.74)	72.22 (1.95)	49.83 (4.90)	74.90 (1.09)	57.48 (3.22)	72.98 (0.03)	46.66 (1.77)	77.35 (0.27)	67.66 (2.57)	72.42 (0.51)	46.06 (1.21)	74.25 (0.27)	53.46 (1.85)
MixturePFN	76.62 (0.61)	56.62 (2.21)	81.89 (0.19)	76.01 (0.68)	77.77 (0.18)	64.52 (0.85)	78.76 (0.33)	65.72 (1.25)	77.18 (0.22)	60.52 (0.62)	79.32 (0.03)	72.02 (0.42)	76.07 (0.07)	58.36 (0.23)	77.52 (0.11)	63.63 (0.42)
LoCalPFN (retrieval)	78.05 (0.26)	60.44 (0.11)	82.58 (0.07)	**77.31** (0.14)	77.42 (0.09)	63.22 (0.53)	79.35 (0.14)	66.99 (0.26)	77.48 (0.07)	60.85 (0.12)	79.63 (0.05)	**72.64** (0.10)	76.42 (0.07)	58.32 (0.20)	77.84 (0.07)	63.94 (0.14)
BETA	76.73 (0.17)	58.08 (0.87)	80.67 (0.22)	73.25 (0.75)	76.80 (0.21)	62.12 (1.29)	78.07 (0.20)	64.49 (0.97)	77.25 (0.49)	60.70 (1.65)	79.39 (0.04)	72.55 (0.83)	76.30 (0.03)	58.35 (0.42)	77.65 (0.19)	63.87 (0.97)
DistPFN	78.09 (0.19)	59.55 (0.26)	82.98 (0.11)	76.96 (0.27)	77.82 (0.07)	63.78 (0.27)	79.63 (0.13)	66.76 (0.27)	78.76 (0.03)	65.16 (0.02)	79.43 (0.03)	71.58 (0.13)	76.70 (0.11)	58.56 (0.23)	78.29 (0.05)	65.10 (0.13)
Robust methods																
CVaR-DRO†	77.00 (0.14)	64.41 (3.70)	77.85 (1.27)	72.85 (1.88)	73.48 (0.74)	61.88 (4.97)	76.11 (0.72)	66.38 (3.52)	74.10 (0.35)	59.42 (2.51)	74.85 (0.14)	68.57 (0.65)	71.75 (0.28)	54.67 (2.55)	73.57 (0.26)	60.89 (1.90)
χ^2-DRO†	76.95 (0.14)	64.07 (4.60)	77.55 (0.92)	71.82 (1.58)	73.13 (1.38)	59.03 (2.06)	75.88 (0.81)	64.97 (2.75)	74.53 (0.25)	61.18 (1.15)	74.23 (0.18)	68.53 (0.67)	71.33 (0.60)	53.05 (1.76)	73.36 (0.34)	60.92 (1.19)
KL-DRO†	75.68 (1.23)	60.98 (3.00)	78.23 (0.32)	71.22 (2.41)	76.10 (0.07)	62.88 (0.37)	76.67 (0.54)	65.03 (1.93)	74.20 (1.20)	57.78 (2.91)	75.35 (0.78)	68.72 (1.71)	74.73 (0.25)	57.20 (0.97)	74.76 (0.74)	61.23 (1.86)
EIIL†	74.98 (1.80)	55.22 (6.65)	78.18 (0.67)	68.42 (5.86)	75.60 (0.49)	63.77 (3.19)	76.25 (0.99)	62.47 (5.23)	75.35 (1.70)	56.80 (5.52)	76.92 (1.59)	65.62 (5.88)	74.78 (1.51)	57.32 (4.95)	75.68 (1.60)	59.91 (5.45)
JTT†	75.70 (1.06)	57.73 (2.49)	77.48 (0.25)	68.48 (2.70)	74.48 (0.53)	60.63 (1.84)	75.89 (0.61)	62.28 (2.34)	73.90 (0.78)	55.17 (3.14)	74.00 (0.28)	66.57 (2.33)	72.65 (0.35)	55.17 (0.62)	73.52 (0.47)	58.97 (2.03)
FAM†	75.00 (0.57)	57.82 (2.17)	78.73 (0.03)	70.12 (3.33)	74.38 (2.09)	59.45 (6.29)	76.04 (0.90)	62.46 (3.93)	72.93 (0.53)	54.60 (2.45)	74.68 (0.11)	66.35 (2.44)	73.40 (1.27)	54.53 (5.24)	73.67 (0.64)	58.49 (3.38)
SRDO†	75.17 (2.22)	60.03 (10.23)	77.85 (1.01)	69.52 (2.49)	73.30 (1.22)	54.63 (5.75)	75.44 (1.48)	61.39 (6.16)	74.00 (1.83)	58.37 (8.57)	74.78 (0.49)	61.77 (4.26)	73.83 (0.35)	60.05 (0.96)	74.20 (0.89)	60.06 (4.60)

Continued on next page

Table 16: ACS Income, methods without true group labels (continued).

Methods	AZ		MA		MI		Avg.		AZ → MA		MA → MI		MI → AZ		Avg.	
	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst
LSR†	76.28 (0.11)	66.42 (0.89)	78.35 (0.14)	73.33 (1.48)	75.53 (1.77)	67.10 (1.61)	76.72 (0.67)	68.95 (1.33)	75.00 (0.07)	62.85 (0.40)	74.75 (0.35)	68.75 (0.30)	74.73 (0.35)	64.10 (0.35)	74.83 (0.26)	65.23 (0.35)
GEORGE (k-means)	74.76 (2.95)	58.70 (14.27)	76.65 (1.68)	68.82 (5.90)	72.82 (1.78)	66.52 (1.05)	74.74 (2.14)	64.68 (7.07)	73.25 (4.66)	61.69 (3.06)	73.22 (1.18)	64.80 (4.67)	74.21 (2.42)	62.10 (5.66)	73.56 (2.75)	62.86 (4.46)
GEORGE (GMM)	65.41 (5.34)	61.03 (5.03)	61.45 (1.48)	55.58 (2.59)	65.06 (0.83)	59.28 (0.87)	63.97 (2.55)	58.63 (2.83)	66.73 (2.57)	61.40 (3.87)	62.24 (0.36)	58.05 (1.89)	61.35 (4.65)	55.30 (8.68)	63.44 (2.53)	58.25 (4.81)
BPA	63.26 (1.32)	54.44 (4.12)	65.78 (1.46)	58.90 (1.97)	63.42 (0.68)	58.18 (1.31)	64.15 (1.15)	57.17 (2.47)	65.65 (0.57)	59.36 (2.86)	62.72 (1.20)	57.64 (1.08)	63.57 (0.16)	58.96 (0.71)	63.98 (0.65)	58.65 (1.55)
SPARE	74.35 (2.61)	69.94 (4.73)	77.51 (1.49)	70.17 (1.32)	71.05 (1.82)	61.67 (5.26)	74.31 (1.97)	67.26 (3.77)	74.23 (0.89)	65.44 (2.30)	73.23 (0.97)	65.67 (1.74)	70.27 (1.20)	58.96 (5.30)	72.58 (1.02)	63.36 (3.11)
TFM adaptation																
TabPFN	78.19 (0.32)	59.87 (0.50)	**83.01** (0.12)	77.02 (0.27)	77.72 (0.03)	63.50 (0.20)	79.64 (0.16)	66.80 (0.32)	77.95 (0.07)	61.57 (0.18)	79.69 (0.08)	71.86 (0.25)	76.81 (0.05)	59.08 (0.12)	78.15 (0.07)	64.17 (0.18)
TabPFN (fine-tuned)	78.23 (0.20)	60.35 (0.39)	82.95 (0.11)	77.03 (0.33)	78.23 (0.24)	65.02 (0.56)	79.80 (0.18)	67.47 (0.43)	77.95 (0.10)	62.05 (0.68)	79.62 (0.07)	72.20 (0.21)	76.76 (0.08)	58.77 (0.63)	78.11 (0.08)	64.34 (0.50)
TabPFN (class-balanced)	78.52 (0.35)	61.44 (1.76)	82.98 (0.09)	76.72 (0.20)	78.12 (0.55)	64.74 (2.05)	79.87 (0.33)	67.63 (1.34)	78.55 (1.13)	64.46 (5.11)	79.67 (0.04)	70.23 (0.17)	76.84 (0.10)	59.15 (0.23)	78.35 (0.42)	64.61 (1.84)
DR-TFM (GR; k-means)	78.81 (0.13)	70.36 (1.00)	79.96 (0.59)	70.93 (1.31)	78.87 (0.30)	**71.90** (0.86)	79.21 (0.34)	71.06 (1.06)	78.43 (0.15)	69.70 (0.72)	78.73 (0.68)	66.43 (0.34)	78.46 (0.15)	68.84 (0.59)	78.54 (0.32)	68.32 (0.55)
DR-TFM (GR; GMM)	79.22 (0.26)	**71.29** (1.61)	81.51 (0.42)	73.97 (0.90)	78.74 (0.65)	71.25 (1.46)	79.82 (0.44)	**72.17** (1.33)	78.88 (0.42)	70.08 (1.01)	79.10 (0.55)	66.99 (1.32)	78.34 (0.32)	68.97 (0.85)	78.77 (0.43)	68.68 (1.06)
DR-TFM (MS-C)	**79.46** (0.36)	69.30 (2.39)	81.24 (1.00)	72.75 (2.43)	**79.30** (0.22)	71.42 (1.84)	80.00 (0.53)	71.16 (2.22)	80.08 (0.05)	73.75 (1.44)	78.82 (0.48)	70.53 (1.12)	**78.55** (0.09)	71.38 (0.35)	79.15 (0.21)	**71.89** (0.97)
DR-TFM (MS-R)	79.43 (0.16)	70.31 (0.58)	81.91 (0.15)	73.94 (1.09)	78.90 (0.12)	71.10 (1.90)	**80.08** (0.14)	71.78 (1.19)	**80.15** (0.17)	**74.93** (0.26)	79.51 (0.09)	66.86 (0.32)	78.54 (0.06)	**71.50** (0.31)	**79.40** (0.11)	71.10 (0.30)

Table 17: ACS Income results for methods using true group labels. Avg. denotes the average of each metric across the three settings in its block. Values in parentheses are standard deviations over three seeds. Bold method names indicate our variants.

Methods	AZ		MA		MI		Avg.		AZ → MA		MA → MI		MI → AZ		Avg.	
	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst
Robust methods																
Fair-TabICL (group-balanced)	79.43 (0.24)	74.38 (0.09)	81.42 (0.03)	78.68 (0.82)	**78.96** (0.20)	75.19 (1.21)	**79.94** (0.16)	76.09 (0.71)	80.10 (0.08)	78.18 (0.03)	79.56 (0.06)	75.01 (0.54)	**78.87** (0.15)	75.41 (1.33)	**79.51** (0.10)	76.20 (0.63)
Fair-TabICL (uncertainty)	77.72 (0.17)	58.40 (0.57)	**82.36** (0.05)	77.46 (0.21)	76.71 (0.31)	62.07 (0.61)	78.93 (0.18)	65.98 (0.47)	77.41 (0.08)	60.40 (0.39)	**79.59** (0.12)	73.40 (0.25)	76.24 (0.13)	57.21 (0.13)	77.75 (0.11)	63.67 (0.26)
GroupDRO	**79.74** (0.09)	**78.39** (0.20)	81.39 (0.43)	78.91 (0.62)	78.23 (0.34)	**76.70** (0.38)	79.79 (0.29)	**78.00** (0.40)	79.19 (0.10)	76.22 (0.26)	79.26 (0.09)	76.24 (0.30)	78.40 (0.17)	75.50 (0.53)	78.95 (0.12)	75.99 (0.37)
TFM adaptation																
DR-TFM (GR)	79.55 (0.40)	76.36 (0.91)	81.45 (0.34)	**79.27** (0.49)	78.64 (0.25)	76.09 (0.80)	79.88 (0.33)	77.24 (0.73)	79.61 (0.47)	78.02 (1.54)	79.37 (0.03)	76.36 (0.93)	78.50 (0.05)	**76.25** (0.37)	79.16 (0.19)	**76.88** (0.95)
DR-TFM (MS)	79.57 (0.26)	75.05 (1.67)	81.35 (0.40)	77.59 (0.57)	78.47 (0.26)	75.83 (0.44)	79.80 (0.30)	76.16 (0.89)	**80.13** (0.33)	**78.60** (0.31)	79.31 (0.05)	**76.60** (0.64)	78.76 (0.03)	74.77 (0.95)	79.40 (0.14)	76.66 (0.63)

A.5 COMPUTATIONAL COST

All runtimes were measured on a server equipped with two Intel Xeon Gold 6230R CPUs (2.10 GHz), using a single NVIDIA GeForce RTX 3090 GPU. Each experiment was timed separately, with no other experiments running concurrently.

Table 18: Runtime in seconds per reported result, averaged over three seeds. Times include hyperparameter selection and all training or adaptation stages; TabPFN requires no parameter updates.

Method	Adult	Bank	Default	Shoppers	Taxi	Avg.
XGBoost	5.5	6.4	7.1	4.6	4.1	5.5
CatBoost	22.8	25.2	24.1	12.8	12.8	19.5
TabPFN	7.5	7.8	5.5	3.2	3.1	5.4
ERM-MLP	32.0	36.6	29.9	15.1	14.8	25.7
DR-TFM (GR; k-means)	25.6	50.2	24.6	9.1	9.4	23.8
DR-TFM (MS; k-means)	35.9	45.0	36.6	16.9	18.8	30.6
GEORGE (k-means)	98.4	117.4	89.1	40.2	39.8	77.0
SPARE	136.6	212.5	104.4	34.4	29.7	103.5
GroupDRO	183.0	213.0	163.8	76.2	84.3	144.1
TabPFN (fine-tuned)	452.3	347.0	123.1	53.1	94.6	214.1
LSR (all stages)	6,602.0	6,171.6	5,619.8	3,700.6	3,688.0	5,156.4
MixturePFN	327.9	322.0	290.8	91.4	115.4	229.5
LoCalPFN (retrieval)	354.3	126.1	74.1	29.1	26.1	121.9
BETA	1,903.5	1,399.6	887.5	657.3	255.0	1,020.6

Table 19 reports peak GPU memory for the methods in Table 3, measured as the maximum memory allocated by PyTorch during a run and averaged over three seeds.

Table 19: Peak GPU memory in GB per reported result, averaged over three seeds.

Method	Adult	Bank	Default	Shoppers	Taxi	Avg.
TabPFN	0.77	0.77	0.62	0.44	0.43	0.61
DR-TFM (GR; k-means)	1.37	1.62	1.15	0.60	0.61	1.07
DR-TFM (MS; k-means)	1.27	1.41	1.02	0.61	0.58	0.98
GEORGE (k-means)	1.62	1.63	1.60	0.71	0.51	1.21
TabPFN (fine-tuned)	10.98	14.45	11.81	5.25	3.75	9.25
LSR (all stages)	3.56	5.07	4.82	1.50	0.67	3.12
MixturePFN	2.14	2.17	2.46	2.25	1.83	2.17
BETA	14.32	14.32	14.40	14.34	14.30	14.33

A.6 Training objective and adaptation strategy

For the comparison in Table 4, DRO-based adaptation uses the objective and model selection of DR-TFM (GR), while ERM-based adaptation follows the standard TabPFN fine-tuning objective in (2). Input-adapter tuning uses a single ensemble member of the BETA adapter architecture (Liu & Ye, 2025) and the training schedule used for query scaling. Table 20 reports the per-dataset results for input-adapter, full-backbone, encoder, decoder, and query-scaling adaptation. Tables 21 and 22 report the corresponding runtime and peak GPU memory per result.

Fine-tuned parameters. The pretrained TabPFN contains $53{,}153{,}144$ parameters. Encoder fine-tuning updates all parameters outside the decoder ($52{,}725{,}752$). Decoder fine-tuning updates its key and query projections and scaling networks ($427{,}392$), whereas query scaling updates only $8{,}320$ parameters. The input adapter contains $33{,}481$–$82{,}536$ parameters across the five benchmarks (0.06%–0.16% of the combined adapter and pretrained model), depending on the input dimension and categorical cardinalities.

Table 20: Mean group accuracy and worst-group accuracy under ERM-based and DRO-based adaptation (GR) for five adaptation strategies, averaged over three seeds. Avg. denotes the average of each metric across the five tabular benchmarks. Values in parentheses are standard deviations over three seeds. Parenthesized values in the Avg. columns are the averages of the per-dataset standard deviations.

Adaptation	Adult		Bank		Default		Shoppers		Taxi		Avg.	
	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst
Input adapter												
ERM	74.34 (0.67)	46.89 (1.58)	69.39 (1.53)	36.98 (2.81)	64.39 (0.10)	30.34 (0.53)	78.31 (0.91)	55.67 (2.12)	75.38 (0.14)	57.36 (0.22)	72.36 (0.67)	45.45 (1.45)
DRO	78.83 (0.51)	65.90 (1.58)	82.70 (1.34)	73.21 (1.38)	67.92 (0.38)	47.05 (1.44)	84.96 (0.61)	79.60 (0.57)	**77.23** (0.27)	67.43 (0.97)	78.33 (0.62)	66.64 (1.19)
Full backbone												
ERM	76.33 (0.35)	50.80 (0.47)	73.10 (0.02)	42.99 (0.21)	65.03 (0.40)	31.71 (2.04)	78.88 (0.76)	56.25 (1.98)	76.11 (0.27)	58.23 (0.66)	73.89 (0.36)	48.00 (1.07)
DRO	78.01 (0.86)	59.56 (3.82)	**85.32** (0.80)	79.81 (1.36)	69.02 (0.70)	60.18 (2.86)	83.34 (1.99)	72.73 (8.66)	75.72 (0.91)	72.64 (1.65)	78.28 (1.05)	68.98 (3.67)
Encoder												
ERM	77.05 (0.13)	52.15 (0.56)	73.50 (0.20)	44.01 (1.13)	65.02 (0.38)	31.29 (1.32)	78.88 (0.76)	56.25 (1.98)	76.13 (0.33)	58.23 (0.66)	74.11 (0.36)	48.39 (1.13)
DRO	77.96 (0.85)	59.40 (3.70)	85.23 (0.69)	79.41 (0.96)	69.18 (0.47)	**60.42** (2.34)	83.30 (1.93)	72.94 (9.01)	75.40 (0.73)	72.41 (1.44)	78.22 (0.93)	68.92 (3.49)
Decoder												
ERM	74.84 (0.48)	48.17 (1.27)	72.78 (1.02)	42.91 (1.70)	64.89 (1.03)	31.66 (2.44)	79.89 (0.19)	61.31 (0.80)	76.30 (0.08)	59.32 (0.26)	73.74 (0.56)	48.68 (1.29)
DRO	79.24 (0.33)	65.50 (0.88)	84.49 (0.48)	78.88 (1.35)	69.33 (0.58)	59.42 (1.13)	**85.86** (0.40)	**82.51** (0.17)	76.25 (0.69)	**73.31** (0.93)	79.03 (0.49)	71.93 (0.89)
Query scaling												
ERM	74.86 (0.16)	48.58 (0.52)	72.29 (1.02)	41.83 (1.99)	64.85 (0.54)	30.94 (1.31)	78.62 (0.32)	56.42 (1.45)	76.05 (0.27)	58.92 (0.79)	73.34 (0.46)	47.34 (1.21)
DRO	**79.64** (0.20)	**67.90** (1.61)	85.01 (0.36)	**80.14** (0.53)	**69.42** (0.62)	60.37 (1.70)	85.79 (0.32)	81.66 (0.50)	76.69 (0.19)	72.79 (0.62)	**79.31** (0.34)	**72.57** (0.99)

Table 21: Runtime in seconds per reported result under DRO-based adaptation for the adaptation strategies in Table 4, averaged over three seeds. Times include group estimation, adaptation, and prediction.

Fine-tuned parameters	Adult	Bank	Default	Shoppers	Taxi	Avg.
Input adapter	56.5	63.0	40.8	26.8	30.3	43.5
Encoder	214.2	439.1	364.7	80.0	107.8	241.2
Full backbone	213.8	438.3	383.8	90.3	106.8	246.6
Decoder	29.9	45.3	25.9	9.6	9.5	24.0
Query scaling	25.6	50.2	24.6	9.1	9.4	23.8

Table 22: Peak GPU memory in GB per reported result under DRO-based adaptation for the adaptation strategies in Table 4, averaged over three seeds.

Fine-tuned parameters	Adult	Bank	Default	Shoppers	Taxi	Avg.
Input adapter	12.58	15.45	10.86	5.02	5.09	9.80
Encoder	12.22	16.10	13.20	5.32	4.27	10.22
Full backbone	12.34	16.24	13.29	5.36	4.31	10.31
Decoder	1.38	1.68	1.18	0.62	0.63	1.10
Query scaling	1.37	1.62	1.15	0.60	0.61	1.07

A.7 Discussion of the χ^2-DRO formulation

We examine a variant that reweights individual samples within a χ^2 ambiguity set, without estimating groups.

Formulation. The variant retains the fixed pretrained TabPFN feature map and adapts the same query scaling network. Write $\ell_\theta(x, y) = \ell(f_\theta(x; \mathcal{D}^{(c)}), y)$. We minimize the robust risk

$$\sup_{Q\,:\,D_{\chi^2}(Q\,\|\,\widehat{P}^{(e)})\le\rho} \mathbb{E}_Q[\ell_\theta] \;=\; \inf_{\eta\in\mathbb{R}} \left\{ \sqrt{2\rho+1}\left(\mathbb{E}_{\widehat{P}^{(e)}}\left[(\ell_\theta-\eta)_+^2\right]\right)^{1/2} + \eta \right\}$$

over θ_L. Here $D_{\chi^2}(Q\|\widehat{P}^{(e)}) = \frac{1}{2}\mathbb{E}_{\widehat{P}^{(e)}}[(dQ/d\widehat{P}^{(e)} - 1)^2]$ for $Q \ll \widehat{P}^{(e)}$, $[a]_+ = \max\{a, 0\}$, and η is a scalar threshold. The right-hand side is the dual formulation of Duchi et al. (2021). We parameterize the radius as $\rho_\alpha = (1/\alpha - 1)^2$ for $\alpha \in (0, 1]$. Thus, $\alpha = 1$ recovers the empirical mean, while smaller values allow more weight to be assigned to high-loss samples.

Empirical results. Table 23 reports results for each tested α. In both evaluation suites, the empirical mean objective ($\alpha = 1$) achieves a higher average of worst-group accuracies than every tested $\alpha < 1$. These results suggest that emphasizing high-loss samples alone may be insufficient to improve worst-group performance. This is consistent with the observation of Tong et al. (2025) that upweighting misclassified samples does not necessarily address the underlying data imbalance.

Table 23: Average of mean group accuracies and average of worst-group accuracies for the χ^2-DRO variant at each tested α, over the five tabular datasets or all six ACS settings, and over three seeds. $\alpha = 1$ uses the empirical mean loss. All tested values are reported. † marks averages computed from accuracy results reported by Tong et al. (2025).

Method	Five tabular benchmarks		ACS Income	
	Mean	Worst	Mean	Worst
ERM-MLP	72.14	44.67	77.54	63.19
χ^2-DRO†	70.00	47.82	74.62	62.95
TabPFN-χ^2-DRO ($\alpha = 1$)	73.34	47.34	78.73	65.26
TabPFN-χ^2-DRO ($\alpha = 0.6$)	73.16	47.19	78.02	63.27
TabPFN-χ^2-DRO ($\alpha = 0.4$)	69.65	37.75	76.93	59.40
TabPFN-χ^2-DRO ($\alpha = 0.2$)	69.49	37.21	76.96	59.55
TabPFN-χ^2-DRO ($\alpha = 0.1$)	69.43	37.14	77.15	60.08

A.8 Extension to other tabular foundation models

Tables 24 and 25 report the per-dataset and per-setting results summarized in Table 5. For each model, we list the pretrained model, DR-TFM (GR) without true group labels ("+ Ours"), and DR-TFM (GR) with true group labels ("+ Ours*"). The starred rows follow the true-group protocol of Appendix A.3.4; the TabPFN + Ours* results are those of Tables 15 and 17. Parenthesized values in the Avg. columns are the averages of the per-dataset or per-setting standard deviations across seeds.

Settings. We follow the training settings in Appendix A.3.6 and use the inference settings specified for each model. Query scaling is applied to the queries in the final attention layer that attends to the context. For models without a pretrained scaling network, we add the network g_{θ_L} of Appendix A.2.4 with its second layer initialized at zero; its modulation $\mathbf{1} + \tanh(g_{\theta_L}(u))$ multiplies the queries while each model's own attention scaling is kept. The remaining model parameters are kept fixed. Without true group labels, groups are estimated from the representations entering the output network, averaged over ensemble members, as for TabPFN.

- TabPFN-3.5. This model uses the same decoder architecture as TabPFN. We use 4 ensemble members and fine-tune its pretrained query scaling network, updating 0.004% of the model's parameters.
- EXAONE. We add the scaling network to scale the queries in the final attention layer operating across samples. We use 8 ensemble members and update 0.020% of the adapted model's parameters.
- TabFM. We use 32 ensemble members and update 0.002% of the adapted model's parameters.

- Causilo. We add the scaling network to scale the queries in the final attention layer of its prediction network. We use 8 ensemble members and update 0.046% of the adapted model's parameters. Groups are instead estimated from the representations entering the final linear layer of the output network, since context representations form a separate cluster at the input to the output network.

Table 24: GR adaptation across tabular foundation models on the five tabular benchmarks. "+ Ours" denotes DR-TFM (GR) applied to the model above; * indicates use of true group labels. Avg. denotes the average of each metric across datasets. Non-bold parentheses report standard deviations over three seeds. Bold parentheses report changes from the corresponding pretrained model, in percentage points.

Methods	Adult		Bank		Default		Shoppers		Taxi		Avg.	
	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst
TabPFN	75.33 (0.05)	49.17 (0.28)	73.07 (0.31)	42.96 (0.76)	64.77 (0.04)	31.09 (0.04)	79.59 (0.32)	58.29 (0.52)	76.01 (0.25)	58.19 (0.53)	73.75 (0.19)	47.94 (0.43)
+ Ours	79.64 (0.20) **(+4.31)**	67.90 (1.61) **(+18.73)**	85.01 (0.36) **(+11.94)**	80.14 (0.53) **(+37.18)**	69.42 (0.62) **(+4.65)**	60.37 (1.70) **(+29.28)**	85.79 (0.32) **(+6.20)**	81.66 (0.50) **(+23.37)**	76.69 (0.19) **(+0.68)**	72.79 (0.62) **(+14.60)**	79.31 (0.34) **(+5.56)**	72.57 (0.99) **(+24.63)**
+ Ours*	78.36 (0.52) **(+3.03)**	72.26 (0.79) **(+23.09)**	86.12 (0.93) **(+13.05)**	81.94 (0.34) **(+38.98)**	69.58 (0.26) **(+4.81)**	65.30 (1.19) **(+34.21)**	85.21 (0.39) **(+5.62)**	80.52 (2.17) **(+22.23)**	77.03 (0.39) **(+1.02)**	68.28 (2.32) **(+10.09)**	79.26 (0.50) **(+5.51)**	73.66 (1.36) **(+25.72)**
TabPFN-3.5	77.79 (0.09)	54.25 (0.30)	73.83 (0.06)	45.31 (0.38)	64.87 (0.11)	31.60 (0.26)	78.92 (0.16)	56.12 (0.55)	76.56 (0.16)	59.00 (0.59)	74.39 (0.12)	49.26 (0.42)
+ Ours	81.96 (0.26) **(+4.17)**	71.82 (0.44) **(+17.57)**	86.40 (0.41) **(+12.57)**	79.81 (0.40) **(+34.50)**	68.92 (0.27) **(+4.05)**	58.94 (0.17) **(+27.34)**	85.17 (0.46) **(+6.25)**	79.92 (1.13) **(+23.80)**	74.66 (2.81) **(-1.90)**	66.83 (2.06) **(+7.83)**	79.42 (0.84) **(+5.03)**	71.47 (0.84) **(+22.21)**
+ Ours*	80.20 (0.84) **(+2.41)**	74.69 (1.08) **(+20.44)**	86.90 (0.22) **(+13.07)**	81.81 (0.52) **(+36.50)**	69.00 (0.33) **(+4.13)**	65.94 (0.99) **(+34.34)**	85.88 (0.21) **(+6.96)**	82.58 (0.57) **(+26.46)**	77.47 (0.33) **(+0.91)**	68.68 (0.23) **(+9.68)**	79.89 (0.39) **(+5.50)**	74.74 (0.68) **(+25.48)**
EXAONE	77.47 (0.02)	53.46 (0.05)	72.67 (0.18)	42.35 (0.26)	64.74 (0.12)	31.10 (0.27)	77.74 (0.16)	53.72 (0.29)	75.99 (0.06)	57.77 (0.16)	73.72 (0.11)	47.68 (0.21)
+ Ours	81.12 (0.07) **(+3.65)**	67.82 (0.35) **(+14.36)**	84.53 (0.53) **(+11.86)**	77.46 (1.39) **(+35.11)**	68.86 (0.07) **(+4.12)**	56.15 (5.45) **(+25.05)**	85.37 (0.16) **(+7.63)**	80.48 (1.35) **(+26.76)**	77.32 (0.50) **(+1.33)**	65.34 (2.48) **(+7.57)**	79.44 (0.27) **(+5.72)**	69.45 (2.21) **(+21.77)**
+ Ours*	81.24 (0.24) **(+3.77)**	74.34 (1.32) **(+20.88)**	86.73 (0.04) **(+14.06)**	80.45 (0.69) **(+38.10)**	69.56 (0.39) **(+4.82)**	63.70 (3.41) **(+32.60)**	85.61 (0.29) **(+7.87)**	80.85 (2.79) **(+27.13)**	77.37 (0.27) **(+1.38)**	68.49 (0.70) **(+10.72)**	80.10 (0.25) **(+6.38)**	73.57 (1.78) **(+25.89)**
TabFM	77.39 (0.02)	52.88 (0.10)	73.75 (0.06)	44.42 (0.10)	65.02 (0.03)	31.91 (0.30)	78.28 (0.25)	54.96 (0.63)	76.34 (0.13)	58.11 (0.28)	74.15 (0.10)	48.46 (0.28)
+ Ours	80.06 (2.09) **(+2.67)**	64.25 (7.86) **(+11.37)**	86.21 (0.66) **(+12.46)**	73.96 (1.28) **(+29.54)**	69.07 (0.75) **(+4.05)**	52.03 (0.94) **(+20.12)**	85.38 (0.56) **(+7.10)**	81.67 (1.97) **(+26.71)**	76.42 (0.65) **(+0.08)**	60.44 (1.65) **(+2.33)**	79.43 (0.94) **(+5.28)**	66.47 (2.74) **(+18.01)**
+ Ours*	81.38 (0.24) **(+3.99)**	72.66 (0.66) **(+19.78)**	87.49 (0.23) **(+13.74)**	78.83 (0.18) **(+34.41)**	69.87 (0.26) **(+4.85)**	59.67 (1.13) **(+27.76)**	85.71 (0.57) **(+7.43)**	81.36 (0.55) **(+26.40)**	77.62 (0.38) **(+1.28)**	68.70 (0.65) **(+10.59)**	80.41 (0.33) **(+6.26)**	72.25 (0.63) **(+23.79)**
Causilo	77.25 (0.12)	52.89 (0.40)	72.95 (0.23)	43.27 (0.47)	64.70 (0.13)	31.27 (0.29)	78.60 (0.05)	56.44 (0.00)	76.28 (0.11)	58.98 (0.07)	73.95 (0.13)	48.57 (0.25)
+ Ours	80.53 (0.13) **(+3.28)**	65.96 (0.62) **(+13.07)**	83.25 (0.13) **(+10.30)**	65.25 (0.97) **(+21.98)**	66.82 (0.31) **(+2.12)**	44.44 (0.65) **(+13.17)**	84.96 (0.14) **(+6.36)**	79.77 (1.70) **(+23.33)**	76.77 (0.35) **(+0.49)**	62.41 (1.58) **(+3.43)**	78.47 (0.21) **(+4.52)**	63.56 (1.10) **(+14.99)**
+ Ours*	80.68 (0.49) **(+3.43)**	73.52 (0.61) **(+20.63)**	86.40 (0.36) **(+13.45)**	81.82 (0.50) **(+38.55)**	69.61 (0.51) **(+4.91)**	64.09 (4.86) **(+32.82)**	85.59 (1.01) **(+6.99)**	81.59 (0.95) **(+25.15)**	77.68 (0.40) **(+1.40)**	68.38 (0.79) **(+9.40)**	79.99 (0.55) **(+6.04)**	73.88 (1.54) **(+25.31)**

Table 25: GR adaptation across tabular foundation models on ACS Income. "+ Ours" denotes DR-TFM (GR) applied to the model above; * indicates use of true group labels. The left block reports within-state results and the right block reports transfers between states. Avg. denotes the average of each metric across the three settings in its block. Non-bold parentheses report standard deviations over three seeds. Bold parentheses report changes from the corresponding pretrained model, in percentage points.

Methods	AZ		MA		MI		Avg.		AZ → MA		MA → MI		MI → AZ		Avg.	
	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst
TabPFN	78.19 (0.32)	59.87 (0.50)	83.01 (0.12)	77.02 (0.27)	77.72 (0.03)	63.50 (0.20)	79.64 (0.16)	66.80 (0.32)	77.95 (0.07)	61.57 (0.18)	79.69 (0.08)	71.86 (0.25)	76.81 (0.05)	59.08 (0.12)	78.15 (0.07)	64.17 (0.18)
+ Ours	78.81 (0.13) **(+0.62)**	70.36 (1.00) **(+10.49)**	79.96 (0.59) **(-3.05)**	70.93 (1.31) **(-6.09)**	78.87 (0.30) **(+1.15)**	71.90 (0.86) **(+8.40)**	79.21 (0.34) **(-0.43)**	71.06 (1.06) **(+4.26)**	78.43 (0.15) **(+0.48)**	69.70 (0.72) **(+8.13)**	78.73 (0.68) **(-0.96)**	66.43 (0.34) **(-5.43)**	78.46 (0.15) **(+1.65)**	68.84 (0.59) **(+9.76)**	78.54 (0.32) **(+0.39)**	68.32 (0.55) **(+4.15)**
+ Ours*	79.55 (0.40) **(+1.36)**	76.36 (0.91) **(+16.49)**	81.45 (0.34) **(-1.56)**	79.27 (0.49) **(+2.25)**	78.64 (0.25) **(+0.92)**	76.09 (0.80) **(+12.59)**	79.88 (0.33) **(+0.24)**	77.24 (0.73) **(+10.44)**	79.61 (0.47) **(+1.66)**	78.02 (1.54) **(+16.45)**	79.37 (0.03) **(-0.32)**	76.36 (0.93) **(+4.50)**	78.50 (0.05) **(+1.69)**	76.25 (0.37) **(+17.17)**	79.16 (0.19) **(+1.01)**	76.88 (0.95) **(+12.71)**
TabPFN-3.5	78.18 (0.29)	59.60 (1.01)	83.19 (0.25)	77.23 (0.44)	78.00 (0.03)	64.47 (0.20)	79.79 (0.19)	67.10 (0.55)	78.00 (0.02)	61.70 (0.06)	79.80 (0.08)	72.23 (0.23)	76.94 (0.04)	59.08 (0.16)	78.25 (0.05)	64.34 (0.15)
+ Ours	79.77 (0.40) **(+1.59)**	71.01 (0.78) **(+11.41)**	81.28 (0.11) **(-1.91)**	73.80 (1.25) **(-3.43)**	78.60 (0.46) **(+0.60)**	72.62 (1.73) **(+8.15)**	79.88 (0.32) **(+0.09)**	72.48 (1.25) **(+5.38)**	79.77 (0.25) **(+1.77)**	74.91 (1.06) **(+13.21)**	79.19 (0.20) **(-0.61)**	66.85 (3.39) **(-5.38)**	78.59 (0.01) **(+1.65)**	72.77 (1.60) **(+13.69)**	79.19 (0.15) **(+0.94)**	71.51 (2.02) **(+7.17)**
+ Ours*	79.90 (0.29) **(+1.72)**	76.20 (0.39) **(+16.60)**	81.42 (0.28) **(-1.77)**	78.95 (0.89) **(+1.72)**	78.96 (0.23) **(+0.96)**	75.11 (0.65) **(+10.64)**	80.09 (0.27) **(+0.30)**	76.75 (0.64) **(+9.65)**	79.73 (0.26) **(+1.73)**	77.74 (0.39) **(+16.04)**	79.37 (0.07) **(-0.43)**	74.99 (2.30) **(+2.76)**	78.56 (0.22) **(+1.62)**	76.24 (0.57) **(+17.16)**	79.22 (0.18) **(+0.97)**	76.32 (1.09) **(+11.98)**
EXAONE	78.06 (0.02)	58.93 (0.19)	82.70 (0.05)	76.05 (0.16)	77.80 (0.07)	63.50 (0.05)	79.52 (0.05)	66.16 (0.13)	78.06 (0.02)	61.52 (0.11)	79.85 (0.02)	72.02 (0.06)	76.90 (0.08)	59.03 (0.21)	78.27 (0.04)	64.19 (0.13)
+ Ours	80.26 (0.17) **(+2.20)**	71.46 (1.12) **(+12.53)**	81.73 (0.22) **(-0.97)**	74.28 (1.42) **(-1.77)**	78.85 (0.45) **(+1.05)**	68.43 (3.68) **(+4.93)**	80.28 (0.28) **(+0.76)**	71.39 (2.07) **(+5.23)**	80.33 (0.41) **(+2.27)**	73.44 (1.22) **(+11.92)**	79.52 (0.43) **(-0.33)**	66.19 (3.40) **(-5.83)**	79.03 (0.13) **(+2.13)**	71.48 (0.91) **(+12.45)**	79.63 (0.32) **(+1.36)**	70.37 (1.84) **(+6.18)**
+ Ours*	79.93 (0.57) **(+1.87)**	76.07 (1.05) **(+17.14)**	81.47 (0.15) **(-1.23)**	78.65 (0.98) **(+2.60)**	79.08 (0.60) **(+1.28)**	75.23 (0.38) **(+11.73)**	80.16 (0.44) **(+0.64)**	76.65 (0.80) **(+10.49)**	80.02 (0.37) **(+1.96)**	78.41 (0.43) **(+16.89)**	79.50 (0.03) **(-0.35)**	73.49 (1.32) **(+1.47)**	78.73 (0.06) **(+1.83)**	76.05 (0.37) **(+17.02)**	79.42 (0.16) **(+1.15)**	75.99 (0.71) **(+11.80)**
TabFM	77.90 (0.06)	59.64 (0.24)	82.95 (0.05)	77.10 (0.08)	78.05 (0.03)	64.60 (0.13)	79.63 (0.05)	67.11 (0.15)	77.97 (0.01)	61.11 (0.02)	79.90 (0.01)	72.14 (0.03)	77.09 (0.01)	59.86 (0.08)	78.32 (0.01)	64.37 (0.04)
+ Ours	79.38 (0.36) **(+1.48)**	69.88 (1.42) **(+10.24)**	80.25 (0.32) **(-2.70)**	64.83 (0.99) **(-12.27)**	79.32 (0.10) **(+1.27)**	73.51 (0.98) **(+8.91)**	79.65 (0.26) **(+0.02)**	69.41 (1.13) **(+2.30)**	79.46 (0.22) **(+1.49)**	70.70 (3.47) **(+9.59)**	78.50 (0.04) **(-1.40)**	58.87 (0.17) **(-13.27)**	78.96 (0.13) **(+1.87)**	70.22 (0.55) **(+10.36)**	78.98 (0.13) **(+0.66)**	66.60 (1.40) **(+2.23)**
+ Ours*	79.59 (0.26) **(+1.69)**	75.88 (0.83) **(+16.24)**	81.40 (0.09) **(-1.55)**	79.29 (0.89) **(+2.19)**	79.20 (0.51) **(+1.15)**	75.52 (0.34) **(+10.92)**	80.06 (0.29) **(+0.43)**	76.90 (0.69) **(+9.79)**	80.08 (0.30) **(+2.11)**	78.43 (0.45) **(+17.32)**	79.43 (0.19) **(-0.47)**	73.69 (1.93) **(+1.55)**	78.75 (0.09) **(+1.66)**	76.24 (0.25) **(+16.38)**	79.42 (0.19) **(+1.10)**	76.12 (0.88) **(+11.75)**
Causilo	77.77 (0.24)	59.45 (0.49)	82.87 (0.14)	76.52 (0.25)	77.77 (0.16)	64.04 (0.52)	79.47 (0.18)	66.67 (0.42)	77.90 (0.09)	61.53 (0.25)	79.85 (0.04)	72.02 (0.25)	76.93 (0.05)	59.31 (0.16)	78.23 (0.06)	64.29 (0.22)
+ Ours	79.53 (0.33) **(+1.76)**	70.25 (2.34) **(+10.80)**	82.26 (0.21) **(-0.61)**	75.04 (0.40) **(-1.48)**	78.88 (0.07) **(+1.11)**	71.16 (1.34) **(+7.12)**	80.22 (0.20) **(+0.75)**	72.15 (1.36) **(+5.48)**	78.51 (1.18) **(+0.61)**	71.61 (0.62) **(+10.08)**	79.11 (0.79) **(-0.74)**	67.17 (1.57) **(-4.85)**	77.50 (1.35) **(+0.57)**	65.83 (0.58) **(+6.52)**	78.37 (1.11) **(+0.14)**	68.21 (0.92) **(+3.92)**
+ Ours*	79.83 (0.40) **(+2.06)**	76.43 (0.88) **(+16.98)**	81.47 (0.14) **(-1.40)**	79.31 (0.28) **(+2.79)**	79.15 (0.18) **(+1.38)**	75.89 (0.17) **(+11.85)**	80.15 (0.24) **(+0.68)**	77.21 (0.44) **(+10.54)**	80.03 (0.27) **(+2.13)**	78.44 (0.66) **(+16.91)**	79.39 (0.08) **(-0.46)**	75.21 (2.06) **(+3.19)**	78.61 (0.12) **(+1.68)**	76.58 (0.13) **(+17.27)**	79.35 (0.16) **(+1.12)**	76.74 (0.95) **(+12.45)**

AI USE STATEMENT

We used generative AI tools to assist with drafting and editing the manuscript and with implementation details. The authors reviewed all AI-assisted content and take responsibility for the final manuscript.