Distributionally robust self-supervised learning for tabular data

Shantanu Ghosh*
Boston University
shawn24@bu.com

Tiankang Xie
Amazon
xietiank@amazon.com

Mikhail Kuznetsov
Amazon
mikuzne@amazon.com

Abstract

Machine learning (ML) models trained using Empirical Risk Minimization (ERM) often exhibit systematic errors on specific subpopulations of tabular data, known as error slices. Learning robust representation in the presence of error slices is challenging, especially in self-supervised settings during the feature reconstruction phase, due to high cardinality features and the complexity of constructing error sets. Traditional robust representation learning methods are largely focused on improving worst group performance in supervised settings in computer vision, leaving a gap in approaches tailored for tabular data. We address this gap by developing a framework to learn robust representation in tabular data during self-supervised pre-training. Our approach utilizes an encoder-decoder model trained with Masked Language Modeling (MLM) loss to learn robust latent representations. This paper applies the Just Train Twice (JTT) and Deep Feature Reweighting (DFR) methods during the pre-training phase for tabular data. These methods fine-tune the ERM pre-trained model by up-weighting error-prone samples or creating balanced datasets for specific categorical features. This results in specialized models for each feature, which are then used in an ensemble approach to enhance downstream classification performance. This methodology improves robustness across slices, thus enhancing overall generalization performance. Extensive experiments across various datasets demonstrate the efficacy of our approach. The code is available: https://github.com/amazon-science/distributionally-robust-self-supervised-learning-for-tabular-data.

1 Introduction

ERM-trained ML models often exhibit systematic errors on specific subpopulations of data, known as error slices. Supervised robust representation learning techniques *e.g.,* DFR [Kirichenko et al., 2022], GroupDRO [Sagawa et al., 2020], JTT [Liu et al., 2021] mitigate the model's error rate on worst group subpopulation. While researchers develop such mitigation methods traditionally in computer vision for supervised setup, there is a lack of approaches to learn robust representation during self-supervised pre-training for tabular data. This paper addresses this gap by developing error slices during the self-supervised learning phase, focusing on data reconstruction to learn robust representations across features. These representations enhance overall the downstream classification performance, leading to better generalization across various data subpopulations other than the worst group.

The literature on robust representation learning is extensive. GroupDRO [Sagawa et al., 2020] improves model generalization by minimizing the worst-case loss over predefined groups within the dataset. This approach uses regularization techniques to ensure consistent performance of a model

*Work completed during an internship at Amazon

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

across all groups, mitigating the risk of poor performance on underrepresented subpopulations. Just Train Twice (JTT) [Liu et al., 2021] introduces a two-stage training process to enhance robustness without explicit group labels. In the first stage, JTT identifies “difficult” examples by training an ERM-based model and identifying samples where the model misclassifies. In the second stage, the model is retrained by up-weighting these difficult examples to improve overall robustness. This up-weighting strategy allows JTT to enhance model performance across diverse data subpopulations. Also, JTT enhances model robustness without requiring prior knowledge of specific error-prone groups. Recently, DFR [Kirichenko et al., 2022] demonstrated that retraining only the last layer of a neural network using group-balanced validation data. However, these approaches are primarily tailored for computer vision, with no established robustness methods for tabular data. Applying robust training strategy *e.g.,* JTT or DFR in supervised label prediction settings for tabular data involves discovering, up-weighting error sets or creating balanced datasets, which is relatively straightforward. Also, for supervised training, these methods focus on improving worst-group performance. However, applying them during the self-supervised reconstruction phase is challenging due to the high cardinality features and the complexity of constructing error sets for multiple features. Despite these challenges, employing them during the reconstruction phase is crucial for learning robust representations. This work addresses these challenges by adapting JTT and DFR to the reconstruction phase, enhancing overall downstream classification performance across various error slices.

Our contribution: In this paper, we propose two novel strategies using JTT and DFR for learning robust representations for tabular data. Unlike traditional supervised setups for label prediction, our approach employs a self-supervised strategy during the reconstruction phase. Each strategy has two stages – ERM pre-training and robust representation learning. The ERM pre-training stage employs an encoder-decoder model using Masked Language Modeling (MLM) loss to learn latent representations, which are then used for downstream classification tasks. The JTT-based robust representation learning first creates an error set to identify the hard samples. Specifically, for each categorical feature, it identifies samples where the reconstruction fails, to create an error set. The DFR-based method employs DFR to create a balanced dataset for each category from the validation set. Next, we finetune the encoder-decoder model either by up-weighting these samples (strategy 1, JTT) or using the balanced dataset (strategy 2, DFR). This finetuning stage focuses on the specific decoder head for that feature, while keeping others fixed. This process results in a specialized model for each categorical feature. During inference, we employ an ensemble approach: we estimate the reconstruction loss for all features for a given sample and identify the feature with the maximum loss. We then select the representation from the specialized model corresponding to this feature, ensuring the use of a robust representation for classification. This approach utilizes the specialized capabilities of each model per categorical feature to enhance classification accuracy. Finally, we fine-tune the classifier using these robust representations and estimate performance across different feature categories. This approach in the reconstruction phase, enhances generalization and achieves superior performance across diverse data subpopulations compared to standard ERM models. Extensive experiments across various datasets and architectures validate our method.

2 Method

Our pretraining strategy consists of two stages: Stage 1 ERM pre-training and Stage 2 robust representation learning. The training process aims to build a robust model capable of both feature reconstruction and target prediction accurately. Stage 1 optimizes Masked Language Modeling (MLM) loss for feature reconstruction using ERM [Vapnik, 1999] to learn latent representations. Motivated by the expert-based model [Ghosh et al., 2023a,b], stage 2 employs two independent strategies using JTT and DFR to learn robust representation. Strategy 1, using JTT, identifies the samples that are not reconstructed correctly, forming the error set for each categorical feature. Next, for each category, it upweights the samples in the error set, learning specific models per category. For strategy 2, using DFR-based pre-training for each category, phase 2 constructs a balanced validation dataset and learns models for each category. For the downstream classification, we employ an ensemble approach to construct the representation. Fig. 1 depicts our method. Lastly, a classifier is trained on these representations to predict target labels. Algorithms 1 and 2 present our proposed algorithm.

2.0.1 Notation:

The dataset $\{\mathcal{X}, \mathcal{Y}\}$ consists of input features $\boldsymbol{x} \in \mathcal{X}$ and target labels $\mathcal{Y}$. The features $\mathcal{X}$ include k categorical features $(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_k)$ and c continuous features $(x_{k+1}, x_{k+2} \cdots x_c)$. Also, N denotes the number of samples. The model architecture includes an encoder-decoder framework. h denotes the encoder of the model, mapping the input x to a latent representation $h(x)$. For k categorical features, $f_1, f_2, \cdots, f_k$ represents the decoder heads reconstructing the categorical features. Similarly, for c continuous features, $f_{k+1}, f_{k+2}, \cdots, f_{k+c}$ represent the decoder heads reconstructing the continuous features.

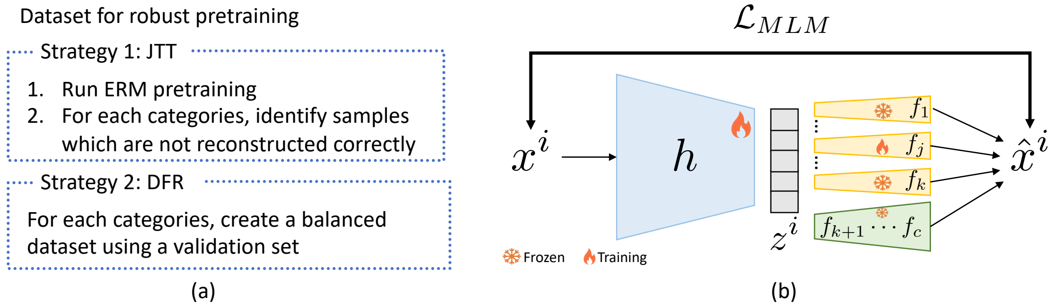

Figure 1: Schematic of our method. (a) Dataset construction of robust pretraining using JTT and DFR. (b) Robust pre-training strategy using MLM loss. We train the encoder h and the reconstruction head of the j^{th} feature f_j for each sample. The embedding z will be used for downstream tasks. We do this for all categorical features, obtaining a pre-trained model per category.

2.0.2 Stage1: ERM with MLM Loss

In Phase 1A, the model is trained using a MLM loss, which aims to reconstruct the input features from the latent representation $h(x)$. The MLM loss, $\mathcal{L}_{\text{MLM}}$, is calculated as follows:

$$\mathcal{L}_{\text{MLM}} = \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{j=1}^{k}\mathcal{L}_{\text{cat}}(f_j(h(\boldsymbol{x}^i)), x_j^i) + \sum_{l=1}^{c}\mathcal{L}_{\text{cont}}(f_{k+l}(h(\boldsymbol{x}^i)), x_{k+l}^i)\right), \tag{1}$$

where $f_j(h(\boldsymbol{x}^i)$ is the output of the j^{th} categorical head of the decoder for sample i, x_j^i is the true value of the j^{th} categorical head of the decoder for sample i, $\mathcal{L}_{\text{cat}}$ is the categorical reconstruction loss (cross-entropy), $f_{k+l}(h(\boldsymbol{x}_i))$ is the output of the l^{th} continuous head of the decoder for sample i, x_{k+l}^i is the true value of the l^{th} continuous feature for sample i, $\mathcal{L}_{\text{cont}}$ is the continuous reconstruction loss (mean squared error).

2.0.3 Stage2: Robust representation learning for tabular data

Strategy 1: JTT-based pre-training. This strategy focuses on improving the robustness of $h(x)$ by employing the JTT method. For each categorical feature j, we define an error set consisting of samples where the predicted categorical feature value does not match the true value, i.e., $x_j^i \neq \hat{x}_j^i$ where $\hat{x}_j^i = f_j(h(\boldsymbol{x}^i))$. We weigh these samples, resulting in a new loss function $\mathcal{L}_{\text{MLM}}^j$ for training the model specialized for feature j:

$$\tilde{\mathcal{L}}_{\text{MLM}} = \frac{1}{N}\sum_{i=1}^{N}\left(w_j^i\mathcal{L}_{\text{cat}}(f_j(h(x^i)), x_j^i) + \left(\sum_{\substack{m=1,\\ m\neq j}}^{k}\mathcal{L}_{\text{cat}}(f_m(h(x^i)), x_m^i) + \sum_{l=1}^{c}\mathcal{L}_{\text{cont}}(f_{k+l}(h(x^i)), x_{k+l}^i)\right)\right) \tag{2}$$

where w_j^i is the upweight factor for sample i concerning feature j, typically a hyperparameter to tune. Note that, in this phase, we only train h, the encoder and $f_j(.)$, the j^{th} head of the deocder of the model keeping the heads of other categorical features and continuous features fixed. This training strategy debias the model for the specific features.

Strategy 2: DFR-based pre-training. This strategy focuses on enhancing the robustness of $h(\boldsymbol{x})$ through the DFR method. For each categorical feature j, we create a balanced validation set $\mathcal{D}_{bal}^{j}$ by selecting samples that represent different feature categories proportionally. This balanced set ensures that the model learns representations that are robust across diverse subpopulations. Finally for for feature j, we optimize the loss in Eq. 1 using $\mathcal{D}_{bal}^{j}$. In contrast to JTT, DFR focuses on training the model using a balanced dataset for each categorical feature j, ensuring that the learned representations are less biased and more generalizable across different feature categories. Unlike the JTT phase, in this phase, we only train $f_j(.)$, the j^{th} head of the decoder, while keeping the heads of other features fixed. Thus it ensures robustness to variations within each feature category, improving overall generalization.

Downstream classifier training with feature-specific model selection For each test sample x^*, we calculate the reconstruction losses for each specialized model corresponding to the categorical features, determining the feature j^* with the maximum loss:

$$j^* = \arg\max_{j} \mathcal{L}_{\text{cat}}(f_j(h(\boldsymbol{x}^*)), x_j^*) \tag{3}$$

The representation from the model j^* is then used to train the classifier g, with the supervised loss:

$$\mathcal{L}_{\text{sup}} = \frac{1}{N}\sum_{i=1}^{N} \text{CrossEntropy}(g(h_{j^*}(\boldsymbol{x}_i)), Y^i) \tag{4}$$

3 Experiments

We conduct using the bank Moro et al. [2014] and census Kohavi [1996] datasets. Refer to Appendix D and Appendix E for dataset and experimental details.

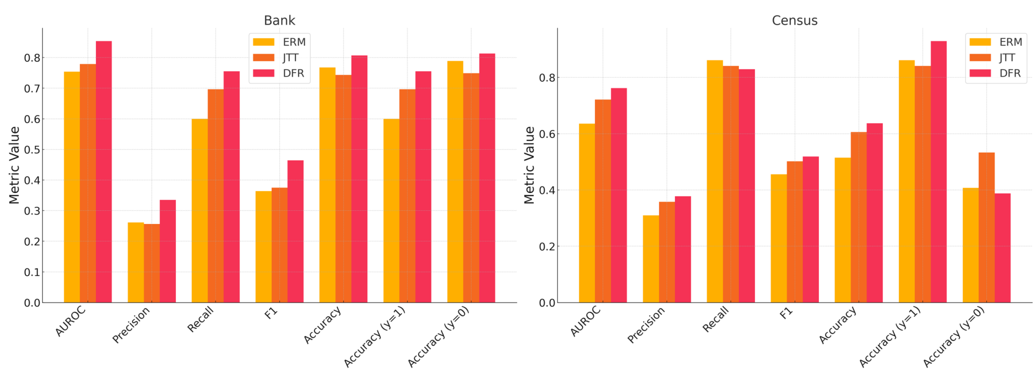

Figure 2: Comparing overall performance of the downstream classifiers using ERM, JTT and DFR

4 Results

Fig. 2 (left) and Fig. 2 (right) illustrate the performance comparison between DFR and JTT on the Bank and Census datasets, respectively. DFR consistently outperforms JTT and ERM across these metrics, demonstrating its effectiveness in learning robust representations through balanced validation sets. DFR creates a balanced dataset for each feature ensuring that the model is not overly influenced by majority classes or features. This leads to a more generalized learning process, allowing DFR to better capture the underlying structure of the data, especially in scenarios where certain features are prone to bias or imbalance. The result is a higher AUROC (14% and 25% gains over ERM for Bank and Census datasets, respectively).

5 Conclusion and limitation

Our approach significantly improves model robustness and generalization across diverse subpopulations. Extensive experiments validate the effectiveness of our method, making it a promising solution for enhancing fairness and accuracy in tabular model training. A key future goal is to extend our framework to handle more complex tabular datasets with high cardinality features.

References

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization, 2020. URL `https://arxiv.org/abs/1907.02893`.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019.

John Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49, 06 2021. doi: 10.1214/20-AOS2004.

Sabri Eyuboglu, Maya Varma, Khaled Saab, Jean-Benoit Delbrouck, Christopher Lee-Messer, Jared Dunnmon, James Zou, and Christopher Ré. Domino: Discovering systematic errors with cross-modal embeddings. *arXiv preprint arXiv:2203.14960*, 2022.

Shantanu Ghosh, Ke Yu, Forough Arabshahi, and Kayhan Batmanghelich. Dividing and conquering a BlackBox to a mixture of interpretable models: Route, interpret, repeat. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 11360–11397. PMLR, 23–29 Jul 2023a. URL `https://proceedings.mlr.press/v202/ghosh23c.html`.

Shantanu Ghosh, Ke Yu, and Kayhan Batmanghelich. Distilling blackbox to interpretable models for efficient transfer learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 628–638. Springer, 2023b.

Shantanu Ghosh, Rayan Syed, Chenyu Wang, Clare B. Poynton, and Kayhan Batmanghelich. Ladder: Language driven slice discovery and error rectification, 2024. URL `https://arxiv.org/abs/2408.07832`.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34:18932–18943, 2021.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=r1gRTCVFvB`.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022.

Ron Kohavi. Census Income. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5GP7S.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C. Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Evan Z 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 *International Conference on Machine Learning*, pages 6781–6792. PMLR, 2021.

S. Moro, P. Rita, and P. Cortez. Bank Marketing. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C5K306.

Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33:20673–20684, 2020.

Shiori Sagawa, Pang Wei Koh*, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=ryxGuJrFvS`.

Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1999.

Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving out-of-distribution robustness via selective augmentation. In *International Conference on Machine Learning*, pages 25407–25437. PMLR, 2022.

Sriram Yenamandra, Pratik Ramesh, Viraj Prabhu, and Judy Hoffman. Facts: First amplify correlations and then slice to discover bias. In *IEEE/CVF International Conference in Computer Vision (ICCV)*, 2023.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=r1Ddp1-Rb`.

A Appendix

B Related Work

B.1 Slice discovery

In computer vision, researchers have developed slice discovery frameworks to identify subpopulations where models consistently make errors that follow specific patterns. Early works on unstructured data include Spotlight, Multiaccuracy, and FailureModeAnalysis. They typically project data into a representation space and identify error slices using clustering or dimensionality reduction techniques. They are primarily assessed through limited slice configurations or qualitative analysis. Recent works include using the vision language representation space for slice discovery and obtaining SOTA results. These methods include the following:

DOMINO: DOMINO [Eyuboglu et al., 2022] identifies systematic errors in machine learning models by leveraging cross-modal embeddings. It operates in three main steps: embedding, slicing, and describing.

1. Embedding: Domino uses cross-modal models (*e.g.,* CLIP) to embed inputs and text in the same latent space. This enables the incorporation of semantic meaning from text into input embeddings, which is crucial for identifying coherent slices.

2. Slicing: It employs an error-aware mixture model to detect underperforming regions within the embedding space. This model clusters the data based on embeddings, class labels, and model predictions to pinpoint areas where the model performance is subpar. The mixture model ensures that identified slices are coherent and relevant to model errors.

3. Describing: Domino generates natural language descriptions for the discovered slices. It creates prototype embeddings for each slice and matches them with text embeddings to describe the common

characteristics of the slice. This step provides interpretable insights into why the model fails on these slices.

FACTS: FACTS [Yenamandra et al., 2023] (First Amplify Correlations and Then Slice) aims to identify bias-conflicting slices in datasets through a two-stage process:

1. Amplify Correlations: This stage involves training a model with a high regularization term to amplify its reliance on spurious correlations present in the dataset. This step helps segregate biased-aligned from bias-conflicting samples by making the model fit a simpler, biased-aligned hypothesis.
2. Correlation-aware Slicing: In this stage, FACTS uses clustering techniques on the bias-amplified feature space to discover bias-conflicting slices. The method identifies subgroups where the spurious correlations do not hold, highlighting areas where the model underperforms due to these biases.

FACTS leverages a combination of bias amplification and clustering to reveal underperforming data slices, providing a foundation for understanding and mitigating systematic biases in machine learning models.

Assuming that an existing feature causes the error slice, identifying error slices in tabular data is straightforward. For tabular data, we can enumerate different feature categories within a particular class. This allows us to determine which specific feature category has a higher error rate than the overall rate. For example, in the UCI-Bank dataset, we aim to determine if an error slice exists for the feature 'job'. We do this by estimating the error rate for each job category within a specific target class and comparing it to the overall error rate for that class. We identify an error slice in any job category that has a significantly higher error rate than the average error rate for the given target class.

B.2 Error mitigation

Error mitigation aims to improve the subgroup's performance where the model performs worst. These algorithms and provide detailed descriptions for each category below:

Vanilla: The empirical risk minimization(ERM) [Vapnik, 1999] algorithm seeks to minimize the cumulative error across all samples.

Subgroup Robust Methods: GroupDRO [Sagawa et al., 2020] proposes a robust optimization strategy, which enhances ERM by prioritizing groups with higher error rates. CVaRDRO [Duchi and Namkoong, 2021] is a variant of GroupDRO that dynamically assigns weights to data samples with the highest losses. LfF [Nam et al., 2020] concurrently trains two models: the first is biased, and the second is de-biased by re-weighting the loss gradient. Just Train twice (JTT) [Liu et al., 2021] proposes an approach that initially trains an ERM model to identify minority groups in the training set, followed by a second ERM model where the identified samples are re-weighted. LISA [Yao et al., 2022] utilizes invariant predictors through data interpolation within and across attributes. DFR [Kirichenko et al., 2022] suggests first training an ERM model and then retraining the final layer using a balanced validation set with group annotations.

Data Augmentation: Mixup [Zhang et al., 2018] proposes an approach that performs ERM on linear interpolations of randomly sampled training examples and their corresponding labels.

Domain-Invariant Representation Learning: Invariant Risk Minimization [Arjovsky et al., 2020] (IRM)learns a feature representation such that the optimal linear classifier on this representation is consistent across different domains. MMD [Li et al., 2018] utilizes maximum mean discrepancy [Gretton et al., 2012] to match feature distributions across domains. Note that all methods in this category necessitate group annotations during training.

Imbalanced Learning: Focal [Lin et al., 2017] introduces Focal Loss, which reduces the loss for well-classified samples and emphasizes difficult samples. CBLoss [Cui et al., 2019] suggests re-weighting by the inverse effective number of samples. LDAM [Cao et al., 2019] employs a modified margin loss that preferentially weights minority samples. CRT and ReWeightCRT [Kang et al., 2020] (a re-weighted variant of CRT) decompose representation learning and classifier training into two distinct stages, re-weighting the classifier using class-balanced sampling during the second stage.

Using Large language models LADDER [Ghosh et al., 2024] leverages a Large Language Model to discover and rectify model error slices by projecting features into a language-aligned space and generating hypotheses for error mitigation without requiring attribute annotations.

All these methods were developed in the context of computer vision to address biases and enhance robustness in supervised learning scenarios. However, there is a significant lack of methods tailored for tabular data, especially those pretrained using self-supervised techniques. Our work aims to bridge this research gap by introducing a method specifically designed for such data.

C Background

C.1 Slice discovery preliminaries

Assume a dataset is denoted as $\{\mathcal{X}, \mathcal{Y}\}$, where $\mathcal{X}$ and $\mathcal{Y}$ represents instances and targets, respectively. We denote $\mathcal{X}_Y$ to be the subset sharing the target Y and a classifier $g : \mathcal{X} \to \mathcal{Y}$. An error slice for a target $Y \in \mathcal{Y}$ includes subset of instances $\mathcal{X}_Y$, where the model performs significantly worse than its overall performance on the entire class Y, formally defined as:

$$\mathbb{S}_Y = \{\mathcal{S}_{Y,\texttt{attr}} \subseteq \mathcal{X}_Y | e(\mathcal{S}_{Y,\texttt{attr}}) \gg e(\mathcal{X}_Y), \exists \texttt{attr}\}, \tag{5}$$

where $e(.)$ is the error rate on the specific data subset and $\mathcal{S}_{Y,\texttt{attr}}$ denotes a subset of the instances $\mathcal{X}_Y$ sharing the attribute `attr`. For example, the error rate for the waterbird class on the land background is higher than the average error rate for the waterbird class in the Waterbirds dataset (Sagawa et al.).

C.1.1 Distributionally Robust Optimization (DRO) preliminaries

Distributionally Robust Optimization (DRO) enhances the resilience and reliability of machine learning models against distributional shifts in data. Unlike traditional optimization strategies that target the empirical average loss, DRO focuses on minimizing the worst-case expected loss over a set of plausible distributions, denoted as $\mathcal{P}$, defined within an uncertainty set. Formally, the DRO objective is expressed as:

$$\min_{\theta} \sup_{P \in \mathcal{P}} \mathbb{E}_{(x,y)\sim P}[\ell(\theta; x, y)] \tag{6}$$

where θ represents the model parameters, (x, y) denotes the data instances and their corresponding labels, and $\ell(\theta; x, y)$ is the loss function. The uncertainty set $\mathcal{P}$ is typically characterized by statistical distances such as the Wasserstein metric, which quantifies the maximum cost of transporting mass in transforming one distribution into another. This approach inherently accounts for worst-case scenarios, which is particularly beneficial in settings where data distributions can vary significantly due to external factors or where rare but critical events must be accurately predicted. By optimizing for the worst-case, DRO ensures that the model maintains stable and robust performance even under adverse or changing conditions, thereby promoting stronger generalization across diverse operational environments. This methodology is pivotal in applications where model performance consistency is crucial, such as autonomous driving and medical diagnostics.

C.2 JTT preliminaries

The Just Train Twice (JTT) approach enhances the robustness of machine learning models, particularly when explicit group labels are unavailable. JTT operates in two distinct phases:

C.2.1 Initial Training Phase:

Consider a classifier $g : \mathcal{X} \to \mathcal{Y}$ is first trained on the entire dataset $\{\mathcal{X}, \mathcal{Y}\}$. During this phase, the model identifies misclassified examples, which are those where the predicted label $\hat{y} = g(x)$ does not match the true label y, i.e., $\hat{y} \neq y$.

C.2.2 Identification of Challenging Examples:

The set of difficult examples $\mathcal{D}_{hard}$ is defined as $\mathcal{D}_{hard} = \{(x, y) \in \{\mathcal{X}, \mathcal{Y}\} \mid g(x) \neq y\}$

C.2.3 Re-weighted Training Phase:

In the second phase, the classifier g is retrained, focusing on the challenging examples identified in the first phase. A re-weighting factor $w(x, y)$ is applied, where $w(x, y) > 1$ for $(x, y) \in \mathcal{D}_{hard}$ and $w(x, y) = 1$ otherwise. The loss function for this phase is expressed as,

$$L_{JTT} = \sum_{(x,y)\in\{\mathcal{X},\mathcal{Y}\}} w(x,y) \cdot \ell(g(x),y), \tag{7}$$

where $\ell(.)$ denotes the loss function, such as cross-entropy for classification tasks.

D Datasets

In our study, we utilized two distinct tabular datasets, each with specific features and objectives relevant to our analysis.

Bank Dataset

- **Features:** The dataset comprises 10 features, all of which are categorical. There are no numeric features or high cardinality features in this dataset.
- **Size:** The dataset contains 41,188 instances.
- **Objective:** The primary goal of this dataset is to predict whether a client will subscribe to a term deposit based on various banking-related attributes.
- **Label:** There are 4,640 identified anomalies in the dataset, corresponding to cases where the client subscribes to a term deposit. This scenario is treated as the anomaly class in our study.

Census Dataset

- **Features:** This dataset includes 33 categorical features. Similar to the bank dataset, it contains no numeric features or high cardinality features.
- **Size:** The dataset is significantly larger, comprising 299,285 instances.
- **Objective:** The purpose of this dataset is to estimate whether an individual's income exceeds \$50K/year based on various demographic and employment-related features.
- **Label:** There are 18,568 anomalies within the dataset, corresponding to instances where the individual's income exceeds \$50K/year. This high-income class is considered the anomaly in our study.

Algorithm 1 Robust Representation Learning with Just Train Twice (JTT)

Initialize model parameters and datasets
Phase 1: ERM and Supervised Learning
Phase 1A: Training with MLM Loss
for each batch of samples x^i do **do**
    Encode input x^i to latent representation $h(x^i)$
    Decode latent representation to reconstruct features using decoder heads $f_j(h(x^i))$
    Compute MLM loss:

$$\mathcal{L}_{\text{MLM}} = \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{j=1}^{k}\mathcal{L}_{\text{cat}}(f_j(h(x^i)), x_j^i) + \sum_{l=1}^{c}\mathcal{L}_{\text{cont}}(f_{k+l}(h(x^i)), x_{k+l}^i)\right)$$

end for
Update model parameters to minimize $\mathcal{L}_{\text{MLM}}$
end for
Phase 1B: Supervised Learning
Use the latent representations $h(x^i)$ to train a classifier g for target prediction Y^i
Compute supervised loss:

$$\mathcal{L}_{\text{sup}} = \frac{1}{N}\sum_{i=1}^{N}\text{CrossEntropy}(g(h(x^i)), Y^i)$$

Update model parameters to minimize $\mathcal{L}_{\text{sup}}$
Phase 2: JTT for Robust Representation
Phase 2A: Model Specialization for Categorical Features
for each categorical feature j do **do**
    Identify error samples x_j^i where $x_j^i \neq \hat{x}_j^i$
    Define error set and upweight factor w_j^i
    Compute specialized MLM loss:

$$\tilde{\mathcal{L}}_{\text{MLM}} = \frac{1}{N}\sum_{i=1}^{N}\left(w_j^i\mathcal{L}_{\text{cat}}(f_j(h(x^i)), x_j^i) + \left(\sum_{m=1, m\neq j}^{k}\mathcal{L}_{\text{cat}}(f_m(h(x^i)), x_m^i) + \sum_{l=1}^{c}\mathcal{L}_{\text{cont}}(f_{k+l}(h(x^i)), x_{k+l}^i)\right)\right)$$

    Train the model on the upweighted error set, focusing on the decoder head for feature j
end for
end for
Phase 2B: Inference with Feature-Specific Model Selection
for each test sample x^i do **do**
    Calculate reconstruction loss for each categorical feature j:

$$j^* = \arg\max_j \mathcal{L}_{\text{cat}}(f_j(h(x^i)), x_j^i)$$

    Use the representation from the specialized model corresponding to j^* for classification
    Compute supervised loss:

$$\mathcal{L}_{\text{sup}} = \frac{1}{N}\sum_{i=1}^{N}\text{CrossEntropy}(g(h_{j^*}(x_i)), Y^i)$$

end for
end for
Evaluate model performance using accuracy, precision, recall, F1-score, and AUROC
Analyze robustness and quality of representations across different feature categories

Algorithm 2 Robust Representation Learning with Deep Feature Reweighting (DFR)

Initialize model parameters and datasets
Phase 1: ERM and Supervised Learning
Phase 1A: Training with MLM Loss
for each batch of samples x^i do **do**
    Encode input x^i to latent representation $h(x^i)$
    Decode latent representation to reconstruct features using decoder heads $f_j(h(x^i))$
    Compute MLM loss:

$$\mathcal{L}_{\text{MLM}} = \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{j=1}^{k} \mathcal{L}_{\text{cat}}(f_j(h(x^i)), x_j^i) + \sum_{l=1}^{c} \mathcal{L}_{\text{cont}}(f_{k+l}(h(x^i)), x_{k+l}^i) \right)$$

end for
Update model parameters to minimize $\mathcal{L}_{\text{MLM}}$
end for
Phase 1B: Supervised Learning
Use the latent representations $h(x^i)$ to train a classifier g for target prediction Y^i
Compute supervised loss:

$$\mathcal{L}_{\text{sup}} = \frac{1}{N} \sum_{i=1}^{N} \text{CrossEntropy}(g(h(x^i)), Y^i)$$

Update model parameters to minimize $\mathcal{L}_{\text{sup}}$
Phase 2: DFR for Robust Representation
Phase 2A: Balanced Validation Set for Each Categorical Feature
for each categorical feature j do **do**
    Create a balanced validation set D_j^{bal} by selecting samples representing different feature categories proportionally
    Train the model using the balanced validation set for feature j using Eq. 1
end for
end for
Phase 2B: Inference with Feature-Specific Model Selection
for each test sample x^i do **do**
    Calculate reconstruction loss for each categorical feature j:

$$j^* = \arg\max_j \mathcal{L}_{\text{cat}}(f_j(h(x^i)), x_j^i)$$

    Use the representation from the model corresponding to j^* for classification
    Compute supervised loss:

$$\mathcal{L}_{\text{sup}} = \frac{1}{N} \sum_{i=1}^{N} \text{CrossEntropy}(g(h_{j^*}(x_i)), Y^i)$$

end for
end for
Evaluate model performance using accuracy, precision, recall, F1-score, and AUROC
Analyze robustness and generalization across feature categories, emphasizing balanced representations

E Experimental details

The encoder-decoder model FT-Transformer [Gorishniy et al., 2021], configured with a dimensionality of 192 for the output features. In Phase 1A, we train the FT-Transformer model with a Masked Language Modeling (MLM) loss for 35 epochs to learn latent representations. The model was optimized using the Adam optimizer with a learning rate of 0.01 and a batch size of 1024. Following this, in Phase 1B, we trained a 1-layer neural network-based supervised classifier on these representations for 100 epochs to predict the target labels. Our Phase 2 training focuses on enhancing the robustness of the learned representations. This phase utilizes the Just Train Twice (JTT) methodology, applied specifically during the reconstruction phase. We identify error samples where the model's reconstruction differed significantly from the actual values, particularly for categorical features. We tune the upweight hyperparameter for these samples by a factor of 20, 50 and 100 for bank dataset to emphasize them during training, with the model retrained for 10 epochs. For the census dataset, we perform the same by a factor of 50, 75 and 100 and 150. All experiments are conducted using PyTorch on a GPU-enabled system, with consistent settings across runs ensured by fixing the random seed to 43. The results validate the effectiveness of our approach in improving model robustness and performance across diverse data subpopulations.

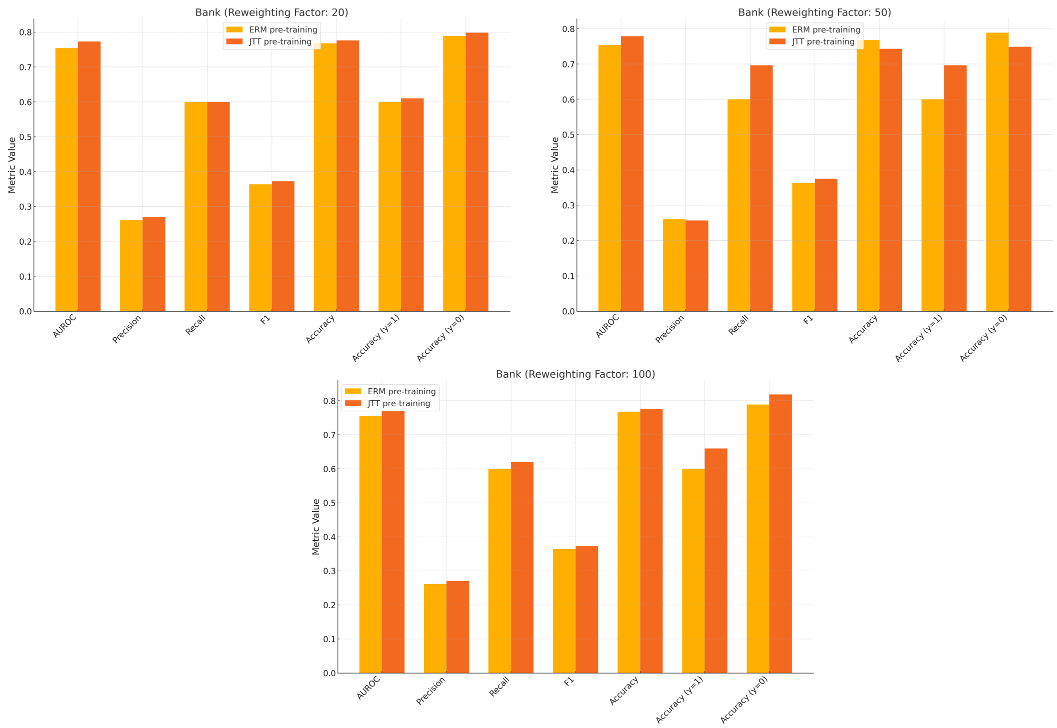

Figure 3: Ablation study on the Bank dataset comparing the performance of Just Train Twice (JTT) across different feature categories. The plot illustrates the impact of JTT on subgroup performance, highlighting how the model's accuracy changes when key features are ablated. Subgroups that were underrepresented or more challenging to classify show notable improvements in accuracy, underscoring the effectiveness of JTT in mitigating bias and enhancing model robustness.

F Extended Results

F.1 Ablations

Ablation Study on the Bank Dataset. Fig. 3 details the results of an ablation study conducted on the Bank dataset to evaluate the contributions of different components of the JTT method. The study reveals that the full implementation of JTT, including the upweighting of difficult examples, is essential for achieving optimal performance. However, it also shows that without addressing the underlying class or feature imbalances (as DFR does), the improvements are limited. The significant drop in accuracy and F1-score when any component of JTT is removed underscores the importance of upweighting hard examples and highlights the limitations of JTT in handling feature imbalance compared to DFR. This indicates that while JTT is effective, it may not be sufficient to address more complex biases present in the data.

Ablation Study on the Census Dataset. Fig. 4 shows the results of a similar ablation study on the Census dataset. The findings align with those observed in the Bank dataset, where the complete JTT method outperforms its ablated versions. The necessity of upweighting in JTT is evident, but the

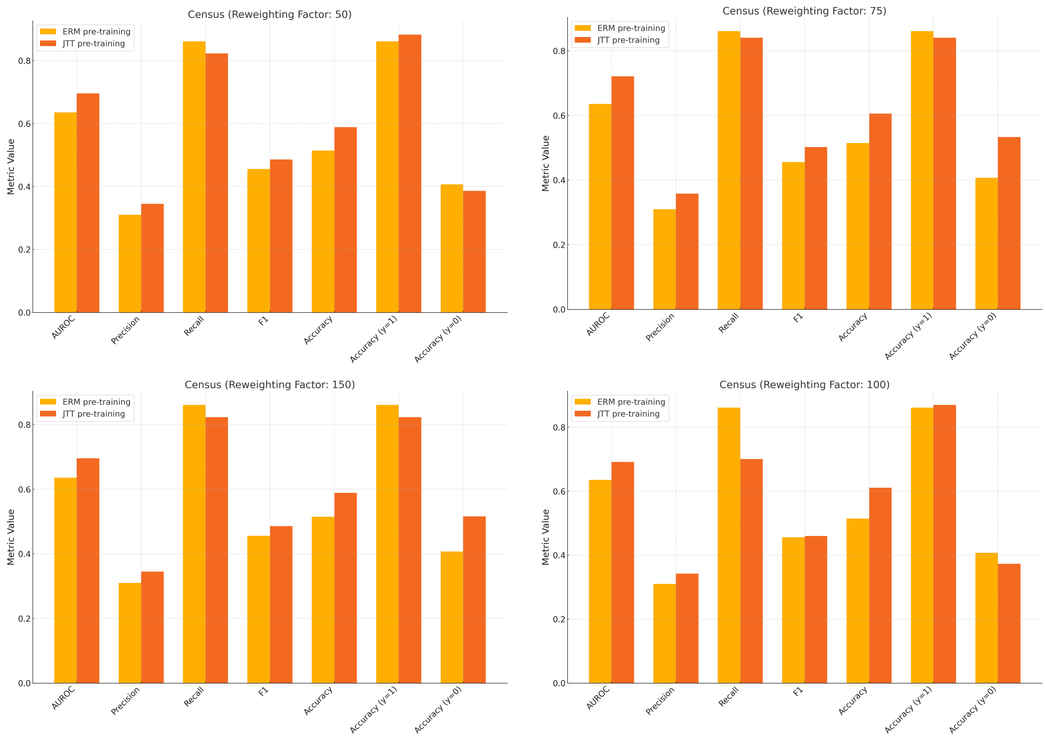

Figure 4: Ablation study on the Census dataset comparing the performance of Just Train Twice (JTT) across different feature categories. The plot illustrates the impact of JTT on subgroup performance, highlighting how the model's accuracy changes when key features are ablated. Subgroups that were underrepresented or more challenging to classify show notable improvements in accuracy, underscoring the effectiveness of JTT in mitigating bias and enhancing model robustness.

method's focus on difficult examples alone, without balancing the dataset, may not fully mitigate bias. This is where DFR's approach provides an edge; by ensuring balanced representation during training, DFR reduces the risk of the model becoming biased towards more frequent or easier-to-learn features. This is particularly crucial in the Census dataset, where high-dimensional data and significant feature imbalances are present. The higher AUROC and F1-score with DFR suggest that this method offers a more comprehensive solution to bias and robustness issues than JTT alone.

In summary, while JTT improves robustness by focusing on difficult examples, DFR's balanced dataset approach offers a more effective solution for mitigating bias, particularly in scenarios with significant class or feature imbalances. This makes DFR a superior method for learning robust representations in unsupervised learning contexts.

Subgoup performance improvements on Bank dataset. In Figure 5, we present the comparison of accuracies for various categorical features on the Bank dataset across three different training strategies: Empirical Risk Minimization (ERM), Just Train Twice (JTT), and Deep Feature Reweighting (DFR) for positively labeled samples ($y = 1$). Each subplot represents a distinct feature (e.g., job, marital, default, etc.), and the corresponding categories within each feature are displayed along the x-axis, with accuracy values reported for each category. This analysis enables us to evaluate the effectiveness of DFR in mitigating performance discrepancies across subgroups.

For the job feature, DFR consistently improves accuracy across most categories compared to ERM and JTT. Categories with lower representation (e.g., categories 6, 8, 9) show particularly strong improvements, where ERM struggles to generalize effectively. This highlights DFR's capability in addressing the long-tail issue inherent in such features.

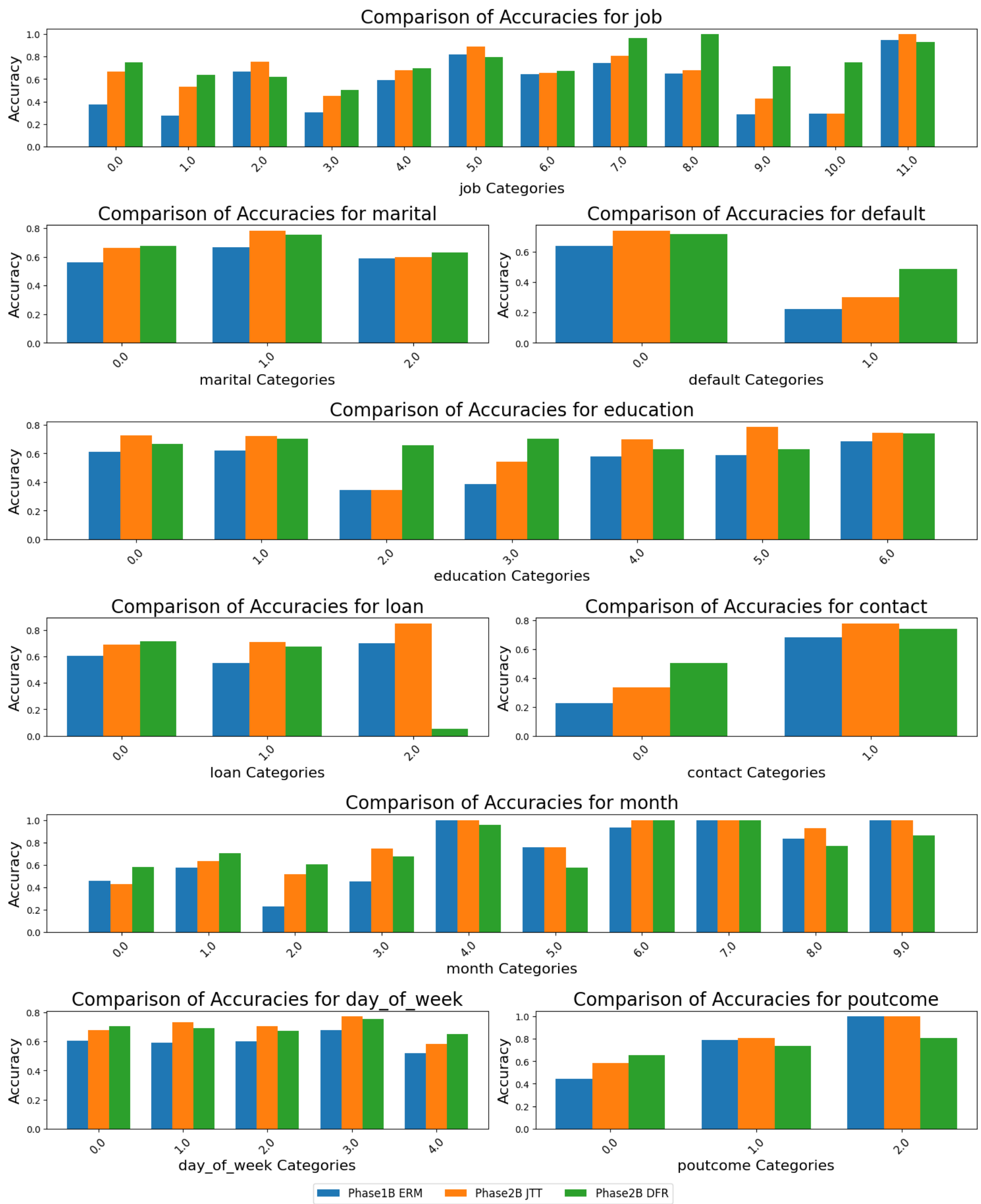

Figure 5: Comparison of accuracies across different categorical features in the Bank dataset, evaluated with Empirical Risk Minimization (ERM), Just Train Twice (JTT), and Deep Feature Reweighting (DFR) for Bank dataset for positively labeled samples ($y = 1$). Each subplot represents a distinct feature, and the x-axis indicates the category within each feature. The y-axis shows the accuracy for each method on that category. DFR consistently improves performance across most categories, particularly in underrepresented subgroups, highlighting its effectiveness in mitigating bias compared to ERM and JTT.

In the marital feature, DFR and JTT exhibit similar performance gains over ERM, especially for the category 1.0, where ERM demonstrates a notable drop in accuracy. The improvements here suggest that both DFR and JTT are effective at mitigating bias in features with fewer categories.

For the education feature, the performance differences are more pronounced. DFR significantly outperforms ERM across nearly all categories, with category 6.0 showing the largest gain. This

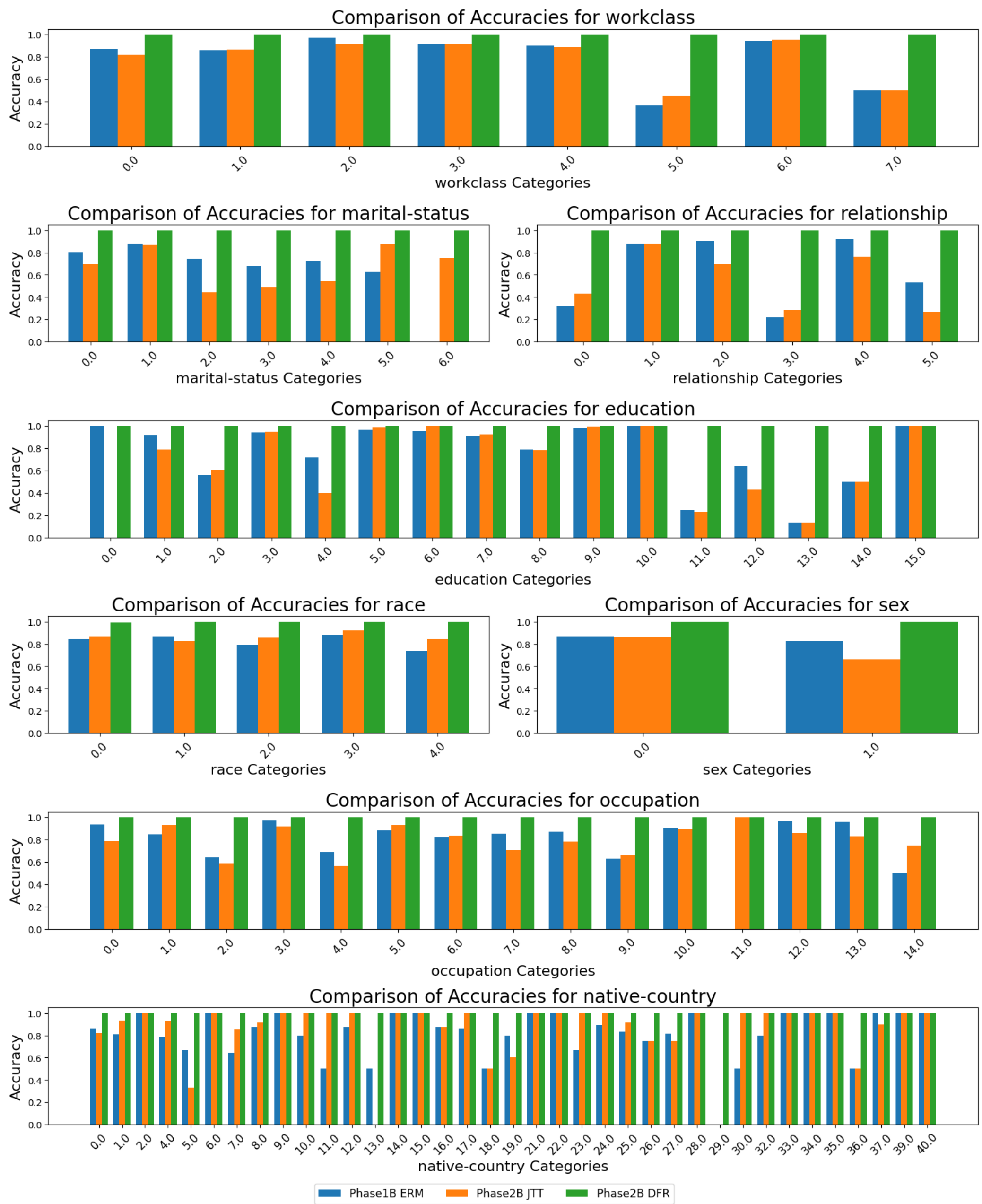

Figure 6: Comparison of accuracies across different categorical features in the Bank dataset, evaluated with Empirical Risk Minimization (ERM), Just Train Twice (JTT), and Deep Feature Reweighting (DFR) for Census dataset for positively labeled samples ($y = 1$). Each subplot represents a distinct feature, and the x-axis indicates the category within each feature. The y-axis shows the accuracy for each method on that category. DFR consistently improves performance across most categories, particularly in underrepresented subgroups, highlighting its effectiveness in mitigating bias compared to ERM and JTT.

suggests that DFR's ability to reweight features leads to better handling of imbalanced subgroups, ensuring that even the minority categories receive adequate representation in the learned model.

The default and loan features display a similar pattern, where DFR improves upon ERM for the underrepresented categories, such as 1.0 in default and 2.0 in loan. These gains are crucial for applications where fairness and robustness to minority subgroups are required.

In contact, DFR delivers substantial gains in accuracy for category 0.0, which was underrepresented in the ERM and JTT models. This suggests that DFR is particularly effective in ensuring that minority subpopulations are not overlooked during the training process.

Lastly, for the month and day_of_week features, the improvements are moderate, but DFR still achieves better accuracy than ERM and JTT across several categories, particularly in category 3.0 for month and 2.0 for day_of_week. These improvements indicate that DFR can capture temporal patterns in the data more effectively than the alternative approaches.

Overall, DFR provides notable performance improvements across all the categorical features, particularly in underrepresented subgroups. These results demonstrate the strength of DFR in mitigating the imbalances that ERM and JTT struggle with, leading to more equitable and robust predictions for all subgroups in the Bank dataset.

Subgoup performance improvements on Census dataset. In the Census dataset (Fig. 4), similar to the Bank dataset, Deep Feature Reweighting (DFR) demonstrates significant improvements across subgroups compared to Empirical Risk Minimization (ERM) and Just Train Twice (JTT) for positively labeled samples ($y = 1$). The most notable performance gains were observed in features with high cardinality, such as occupation and education, where DFR was able to balance the representation of smaller subpopulations, thereby reducing bias.

For the occupation feature, DFR consistently achieved higher accuracy across most subgroups compared to ERM and JTT. Subgroups that were previously underrepresented, such as categories related to more specialized occupations, showed the most significant improvements in accuracy, indicating that DFR effectively mitigates the long-tail distribution issue inherent in this feature.

Similarly, the education feature exhibited clear gains in subgroup performance under DFR. Categories representing individuals with less common educational backgrounds, such as those with higher or lower levels of education, benefited greatly from DFR's ability to construct balanced datasets. This resulted in improved classification accuracy and fairness, especially for underrepresented educational subgroups.

In the income feature, DFR outperformed ERM and JTT in distinguishing between different income brackets, particularly in the high-income subgroup. This improvement is critical in real-world applications where fairness across income levels is essential.

The race and gender features also improved, with DFR ensuring that minority subgroups received fair representation during training. This resulted in better overall performance on these features, reducing the disparities observed in models trained with ERM and JTT.

Overall, the Census dataset demonstrates that DFR enhances subgroup performance across a wide range of features, particularly those with imbalanced or underrepresented categories. These results emphasize the importance of reweighting strategies like DFR in improving fairness and robustness in large-scale tabular datasets.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims on performance improvement are confirmed by the experimental study.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In the conclusion, we discuss the main limitation – also a potential of future work – the extend the framework and test it on more complex tabular datasets.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Algorithm 1 and Algorithm 2, as well as section E in the Appendix, provide full disclosing of information needed to reproduce the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We plan to release the code upon acceptance, given it will be approved by internal policy.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper provides all details on hyperparameters in Apendix E, and Ablation study in Appendix F.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The experiments are based on a single train/test split fith a fixed random seed, and focus on measuring diverse metrics instead.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: We provide some details in appendix E but not fully disclose the total compute required.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The paper doesn't deal with generative models and only aims to improve the encoding part, therefore we don't expect to have any societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: There are no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The code and datasets are correctly cited in the paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.