EFFICIENT AND INTERPRETABLE TRANSFORMER FOR COUNTERFACTUAL FAIRNESS

Anonymous authors
Paper under double-blind review

ABSTRACT

The growing reliance on machine learning for decision-making in highly regulated industries has raised fundamental challenges in regulatory fairness and interpretability compliance. Although attention-based transformers provide a powerful framework for modeling complex feature dependencies, existing debiasing methods for tabular data either fail to guarantee counterfactual fairness or rely on restrictive causal assumptions that limit practical applicability. We propose Feature Correlation Transformer (FCorrTransformer), an efficient and attention-light architecture specifically designed for tabular learning, where the attention matrix offers direct interpretation as pairwise feature dependencies. Building on this structure, we introduce Counterfactual Attention Regularization (CAR), a regularization framework that enforces group-invariant representations of sensitive features at the attention level, thereby yielding counterfactually fair predictions. We further develop an optimized input augmentation strategy for efficient counterfactual attention learning with minimum computation resource. Empirical experiments on both imbalanced classification and regression benchmarks demonstrate that FCorrTransformer+CAR achieves strong counterfactual fairness guarantees with preserved predictive performance, offering a practical framework for responsible AI in regulatory-sensitive domains.

1 INTRODUCTION

The rapid adoption of machine learning (ML) has catalyzed transformative changes across many industries through data-driven decision making, while also raising significant governance and ethical concerns, particularly in heavily regulated sectors such as finance and insurance. For example, regulatory initiatives including *Protecting Consumers from Unfair Discrimination in Insurance Practices* (SB21-169)[1] proposed by the State of Colorado, the *NAIC Model Bulletin*[2] issued by the National Association of Insurance Commissioners (NAIC), and *Insurance Circular Letter No. 7*[3] from the New York State Department of Financial Services explicitly prohibit "unfair discrimination" in insurance practices. Although formal definitions of algorithmic fairness/bias are still evolving, the insurance industry already operates under strict regulatory constraints, where even subtle forms of bias in ML may lead to regulatory penalties. In addition, interpretability is a fundamental requirement for ML applications in insurance, not only for business operations but also to satisfy regulatory expectations, given the critical societal and financial roles insurers play. Consequently, there is a strong demand for interpretable ML models with rigorous fairness guarantees in financial domains such as insurance. In this work, we focus on counterfactual fairness (Kusner et al., 2017), which requires that insurers do not discriminate against individuals based on sensitive features, and that policyholders who are identical in all respects except for their sensitive features should receive identical treatment, providing a practical and actionable notion of fairness.

To achieve algorithmic fairness, a wide range of bias mitigation strategies, from simple preprocessing methods to more complex in-processing approaches, have been proposed for tabular learning

[1] https://doi.colorado.gov/for-consumers/sb21-169-protecting-consumers-from-unfair-discrimination-in-insurance-practices

[2] https://content.naic.org/sites/default/files/2023-12-4%252520Model%252520Bulletin_Adopted_0.pdf

[3] https://www.dfs.ny.gov/industry-guidance/circular-letters/cl2024-07

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.

(Calders & Verwer, 2010; Kamishima et al., 2012; Zafar et al., 2017). The task of bias mitigation can be conceptually understood as modifying feature dependencies by suppressing or reshaping dependencies of sensitive features in accordance with societal and regulatory objectives. The attention mechanism introduced by transformers (Vaswani et al., 2017) provides a natural framework for modeling feature dependencies, making it an appealing tool for identifying and mitigating unintended biased dependencies. While attention-based debiasing methods have been explored in language (Gaci et al., 2022; Haque et al., 2025) and vision (Qiang et al., 2025) tasks, relatively little work has addressed this problem in tabular settings. Moreover, most existing counterfactual fairness methods rely either on explicit causal models to characterize feature dependencies (Kusner et al., 2017; Rao et al., 2021; Robertson et al., 2025), or on post-attention processing (Rao et al., 2021; Du et al., 2021; Wang et al., 2023). Furthermore, enforcing fairness at the attention level alone does not generally guarantee counterfactually fair predictions under standard transformer architectures. These limitations highlight the need for assumption-light, attention-based bias mitigation frameworks that are both interpretable and capable of achieving counterfactual fairness in tabular settings.

We reformulate fair prediction as a problem of fair attention, which aligns with the notion of counterfactual fairness and enables more intuitive interpretation and regularization. To address the limitation that fair attention may still lead to discriminatory predictions under classical attention mechanisms, we propose a modified transformer architecture, termed *Feature Correlation Transformer* (FCorrTransformer), whose attention matrix offers a more direct interpretation as feature dependencies. Building on this architecture, we introduce a regularization-based method, *Counterfactual Attention Regularization* (CAR), which is applicable under both classification and regression settings, by enforcing fairness directly on the attention matrices and achieving strong counterfactual fairness in our empirical evaluations.

In summary, our main contributions are:

- We propose a modified transformer architecture, FCorrTransformer, which is computationally efficient and provides a interpretable attention mechanism for tabular data
- We introduce CAR, a regularization framework that enforces counterfactual fairness by directly constraining the attention matrix
- We validate the effectiveness of the proposed framework through experiments on both imbalanced classification and regression tasks

2 Related Work & Formulation

Attention-based Tabular Learning: Following the success of transformers in natural language processing (Vaswani et al., 2017), attention mechanisms have been adapted to tabular data through high-dimensional feature embeddings (Huang et al., 2020b; Gorishniy et al., 2021), inter-sample interactions (Somepalli et al., 2021), sequential feature selection (Arik & Pfister, 2021), cross-table transfer (Zhu et al., 2023), tabular foundation models (Hollmann et al., 2025), and embedding tabular as natural language representations (Hegselmann et al., 2023; Yan et al., 2024). Despite their empirical strength, these architectures typically rely on high-dimensional dense embeddings, which obscure explicit pairwise feature dependencies and allow biased representations to propagate implicitly across layers.

Attention-based Bias Mitigation: To mitigate algorithmic bias, existing techniques deploy adversarial objectives (Zhang et al., 2018), customized loss constraints (Qian et al., 2019), or group-invariant embeddings (Lu et al., 2020). Because bias corresponds to undesirable dependencies involving sensitive attributes, attention regularization has emerged as a promising tool, either by penalizing sensitive-token weights (Attanasio et al., 2022; Gaci et al., 2022; Haque et al., 2025), isolating biased components via ensembling (Wang et al., 2023), or applying adversarial attribution (Mehrabi et al., 2022; Qiang et al., 2025). However, existing approaches focus primarily on language and vision tasks and modify post-attention representations.

Problem Formulation: Let $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y}) = \{\boldsymbol{X}_n, y_n\}_{n=1}^N$ denote a dataset consisting of N independent observations. For each observation n, the data consists of two components: feature vectors $\boldsymbol{X}_n \in \mathcal{X}$ with p features and response feature $y_n \in \mathcal{Y}$. The feature matrix $\boldsymbol{X}$ can be decomposed into sensitive features $\boldsymbol{X}_s \in \mathcal{X}_s$ and non-sensitive features $\boldsymbol{X}_r \in \mathcal{X}_r$, which may be correlated with sensitive features $\boldsymbol{X}_s$. In this work, we assume that the sensitive features are categorical. In

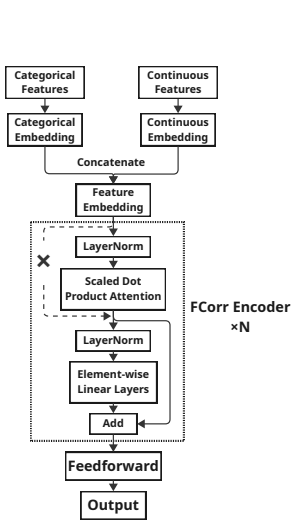
the continuous case, a binning strategy can be employed to construct approximated categorical features. Furthermore, for the simplicity of the formulation, we assume there exists only one sensitive feature, which can be extended to multi-sensitive feature scenarios if needed. Hence, for n-th observation, the sensitive feature $x_{s,n} \in \mathcal{S} = \{s_i\}_{i=1}^{C^s}$ where s_i denotes the i-th unique category of the sensitive feature, and C^s is number of unique categories of the sensitive feature. Finally, the supervised ML model $f : \boldsymbol{X} \rightarrow \boldsymbol{y}$ maps the combined features to the response feature, producing predictions $\hat{\boldsymbol{y}} = f(\boldsymbol{X})$. Under the formulation, a ML model f is considered counterfactually fair if $p(\hat{y}|\boldsymbol{X}_r = \boldsymbol{x}_r, X_s = s_i) = p(\hat{y}|\boldsymbol{X}_r = \boldsymbol{x}_r, X_s = s_j)$ for any input $\boldsymbol{x}_r$ and sensitive categories $s_i, s_j \in \mathcal{S}$ (Kusner et al., 2017). Please refer to Appendix A for the notations used in the paper.

3 METHOD

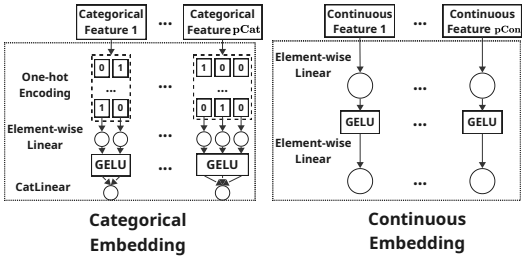
To enable counterfactual fairness learning via attention mechanisms, we propose a framework integrating FCorrTransformer, optimized for capturing complex feature dependencies in tabular data, with a CAR scheme designed to learn counterfactually fair representations.

3.1 FCORRTRANSFORMER: FEATURE CORRELATION TRANSFORMER

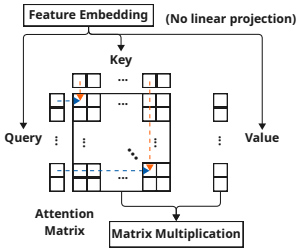
We design FCorrTransformer with the following considerations: 1) Since we aim to equate fair predictions with fair attention, classical transformer architecture must be modified to prevent biased information from bypassing the attention mechanism. 2) Unlike classical tabular transformers that utilize high-dimensional embeddings (Gorishniy et al., 2021; Somepalli et al., 2021; Zhu et al., 2023), we argue that raw feature values in tabular data already contain sufficient information for effective attention. Therefore, the attention mechanism in FCorrTransformer is designed to efficiently learn representations that remain close to the original inputs while explicitly capturing feature dependencies.



(a) Overall architecture



(b) Embedding architecture



(c) Attention architecture

Figure 1: Architecture design of FCorrTransformer

Overall Architecture: As illustrated in Figure 1a, the overall FCorrTransformer architecture follows an encoder-only structure similar to FT-Transformer. Both numerical and categorical features are first passed through corresponding one-dimensional embedding layers and then concatenated to form the feature embedding. In each FCorr encoder layer, layer normalization (LayerNorm) is applied before the dot-product attention and the linear transformations (Element-wise Linear layers), consistent with FT-Transformer, which has been shown to provide superior performance in our prior empirical analysis. Since the feature representations are one-dimensional, we adopt Element-Wise

Linear layers instead of fully connected layers to avoid introducing artificial feature interactions through the representation space. Furthermore, to prevent biased information from bypassing the attention mechanism, we remove the residual connections around the attention block while retaining residual connections in the Element-Wise Linear layers. Finally, a standard multi-layer feedforward network is appended after the encoder stack to produce the final predictions.

Embedding Architecture: Figure 1b illustrates the embedding design in FCorrTransformer. For categorical features, each feature is first one-hot encoded and then passed through an Element-Wise Linear layer followed by a GELU activation (Hendrycks & Gimpel, 2016). To ensure dimensional consistency with the original inputs, the projected category-wise activations for each categorical feature are subsequently aggregated through a linear layer, termed *CatLinear* layer, producing a single activation per categorical feature. The consistency of the embedding dimension facilitates direct control over the attention matrix and improves interpretability. For numerical features, two Element-Wise Linear layers with a GELU activation in between are used to project each feature independently. Consequently, the resulting feature embeddings preserve the original input dimensionality, and no feature dependency is introduced at the embedding stage.

Attention Architecture: We modify the standard scaled dot-product attention in FCorrTransformer, as illustrated in Figure 1c. Unlike the standard attention mechanism, which applies independent linear projections to obtain the query, key, and value matrices, FCorrTransformer removes these projections entirely. This design choice is motivated by the desire to prevent biased information from bypassing the attention matrix and by the assumption that the embedding layers already provide sufficient representations for attention. The first layer attention matrix for input $\boldsymbol{x}_n$ is defined as

$$\text{Attention}(\boldsymbol{x}_n) = \boldsymbol{A}(\boldsymbol{x}_n)^1 \boldsymbol{E}(\boldsymbol{x}_n) = \text{SoftMax}(\frac{\boldsymbol{E}(\boldsymbol{x}_n)\boldsymbol{E}(\boldsymbol{x}_n)^T}{\sqrt{p}})\boldsymbol{E}(\boldsymbol{x}_n) \tag{1}$$

where $\boldsymbol{E}(\boldsymbol{x}_n) \in \mathbb{R}^{p\times 1}$ denotes the feature embedding, $\boldsymbol{A}(\boldsymbol{x}_n)^1 \in \mathbb{R}^{p\times p}$ is the first-layer attention matrix, and p is the number of features. Since each feature embedding is one-dimensional, the dot-product attention reduces to computing pairwise scalar similarities between features, which are then normalized by the SoftMax operator. Furthermore, since multi-head attention can be difficult to interpret and we believe that a single attention head is sufficient to capture the essential dependencies among features in tabular settings, we adopt a single-head attention design throughout this work.

3.2 CAR: COUNTERFACTUAL ATTENTION REGULARIZATION

The architecture of the FCorrTransformer constrains the attention matrix to explicitly encode any biased feature dependency, thereby aligning fair predictions with fair attention matrices. To achieve such counterfactually fair attention, one of the intuitive solutions is counterfactual data augmentation (CDA), a technique established in prior literature (Kusner et al., 2017; Mutlu et al., 2022; Ma et al., 2023). By augmenting the dataset with counterfactual pairs, identical in all features except for the sensitive feature, we can isolate the specific influence of the sensitive feature within the resulting attention matrices. For any given sample $\boldsymbol{x}_n$ comprising non-sensitive features $\boldsymbol{x}_r$ and a sensitive feature x_s, the CAR term is defined as

$$\mathcal{L}_{CAR}(\boldsymbol{x}_n) = \frac{1}{C^s}\sum_{i=1}^{C^s} \|\boldsymbol{A}(\boldsymbol{x}_r, x_s)^1_{:,\sigma} - \boldsymbol{A}(\boldsymbol{x}_r, s_i)^1_{:,\sigma}\|_2^2 \tag{2}$$

where $\boldsymbol{A}(\boldsymbol{x}_r, x_s)^1$ denotes the first-layer attention matrix under the input $(\boldsymbol{x}_r, x_s)$, and σ is the position of the sensitive feature. This σ-th column vector represents the dependencies between the sensitive feature and all non-sensitive features. The CAR objective enforces a group-invariant representation of the sensitive feature in the attention space. By the construction of the CAR loss, we can theoretically show that, if the CAR loss is sufficiently small, the resulting model satisfies the definition of counterfactual fairness.

Theorem 1 (Translation from Fair Attention to Fair Prediction)**.** *Suppose the model f satisfies Assumption 1 (Post-attention Lipschitz Continuity) and 2 (Embedding Jacobian Condition). If the CAR loss defined in Equation 2 is bounded by $\mathcal{L}_{CAR}(\boldsymbol{x}_n) \leq \epsilon$, ($\epsilon > 0$, $\boldsymbol{x}_n = [\boldsymbol{x}_r, x_s]$), there exists such constant $U > 0$, such that for any two sensitive categories $s_i, s_j \in \mathcal{S}$, we have*

$$|f(\boldsymbol{x}_r, s_i) - f(\boldsymbol{x}_r, s_j)| \leq U\sqrt{\epsilon}$$

Consequently, as $\epsilon \to 0$, the model f achieves direct counterfactual fairness.

Proof. See Appendix B.1. □

The theorem establishes that, as the CAR regularization loss approaches zero, the prediction difference between counterfactual samples also converges to zero, thereby satisfying counterfactual fairness. This theoretical guarantee critically relies on both the architectural design of FCorrTransformer, the removal of the attention residual connection and linear projections, and the formulation of the CAR loss. In contrast, for standard transformer architectures, fair attention weights do not necessarily translate into counterfactually fair predictions.

However, this CDA-based approach introduces significant computational and memory overhead, particularly when dealing with high-dimensional sensitive features, as the framework requires C^s+1 forward passes for each batch. To mitigate this computational overhead, we adopt an optimization strategy inspired by AttenD (Gaci et al., 2022). During training, the input is augmented with all unique counterfactual sensitive categories, enabling the generation of all sensitive-related attention vectors within a single forward pass. However, naively expanded FCorrTransformer may learn distinct and biased embeddings for the original sensitive feature and its augmented counterfactual categories while still minimizing the misspecified regularization term, which is undesirable. Instead, our approach ensures that the sensitive feature's embeddings and linear projections are shared across all counterfactual categories. This design choice maintains the original parameter size while avoiding feature mismatch. Specifically, we introduce three specialized modules, *SenLayerNorm*, *SenElementWiseLinear*, and *SenCatLinear*. These three modules extract corresponding parameters of the sensitive feature from their non-counterfactual counterparts to construct a unified representation for the augmented counterfactual categories. Please refer to Appendix E for detailed CAR modification and the specialized modules. By expanding the feature space, the attention matrix is transformed from $\mathbb{R}^{p\times p}$ to $\mathbb{R}^{(p+C^s)\times(p+C^s)}$. The corresponding CAR term can be expressed as

$$\mathcal{L}_{CAR}(\boldsymbol{x}_n) = \frac{1}{C^s}\sum_{i=1}^{C^s} \|\boldsymbol{A}^{aug,1}_{:p,\sigma} - \boldsymbol{A}^{aug,1}_{:p,p+i}\|_2^2 \tag{3}$$

where $\boldsymbol{A}^{aug,1}_{:p,p+i}$ denotes the first p rows of the $(p+i)$-th column of the first-layer attention matrix with the augmented inputs. As demonstrated in Appendix E, this input augmentation strategy substantially improves training efficiency. Since CAR is unaffected by LayerNorm, Theorem 1 remains valid for the input augmentation variant. During inference, input augmentation is suppressed to preserve computational efficiency, reverting the attention matrix to its original $p \times p$ dimensions.

During training, the CAR-incorporated loss function yields the following objective:

$$\mathcal{L} = \mathcal{L}_{perf} + \lambda\mathcal{L}_{CAR} \tag{4}$$

where $\mathcal{L}_{perf}$ denotes the model performance loss, λ is regularization coefficient.[4] To maintain the structural integrity of the baseline architectures, we employ the CDA-based approach for all comparative models, while utilizing the optimized input augmentation method for the FCorrTransformer.

4 EXPERIMENTS

To evaluate the proposed framework, we first conduct experiments on a synthetic toy dataset in Subsection 4.1 to analyze the attention mechanism and CAR in FCorrTransformer. We then evaluate the framework on three datasets, focusing on highly imbalanced settings motivated by real-world applications such as insurance. We evaluate the performance of the proposed framework under both regression and classification tasks. Due to space constraints, we present one case study in Subsection 4.2 and defer the remaining two to Appendices G.2 and G.3. Across all three experiments, model hyperparameters are optimized using Optuna (Akiba et al., 2019), with 300 trials per configuration. We maximize AUROC for classification tasks and minimize MSE for regression tasks, unless otherwise specified. Experiments are conducted on NVIDIA H200 GPUs. Importantly, we'd like to acknowledge that the computational efficiency of FCorrTransformer enables training on consumer-grade hardware, making the proposed architecture practical for resource-constrained settings.

[4] In practice, we set the regularization coefficient as $\lambda = 10^{\lfloor \mathcal{L}^1_{perf}/\mathcal{L}^1_{CAR} \rfloor}$, where $\mathcal{L}^1_{perf}$ and $\mathcal{L}^1_{CAR}$ denotes the corresponding losses on the first batch sample and $\lfloor\cdot\rfloor$ is the floor function. The coefficient setting generally guarantees counterfactual fairness in our experiments.

4.1 SYNTHETIC DATA

We modify the synthetic data structure studied in Mehrabi et al. (2022), following

$$X_1 \sim \begin{cases} Ber(0.7), \text{if} X_2 = 1 \\ Ber(0.3), \text{if} X_2 = 0 \end{cases}, X_2 \sim Ber(0.5), X_3 \sim Ber(0.5), y \sim \begin{cases} Ber(0.8), \text{if} X_2 + X_3 = 2 \\ Ber(0.2), \text{o.w} \end{cases}$$

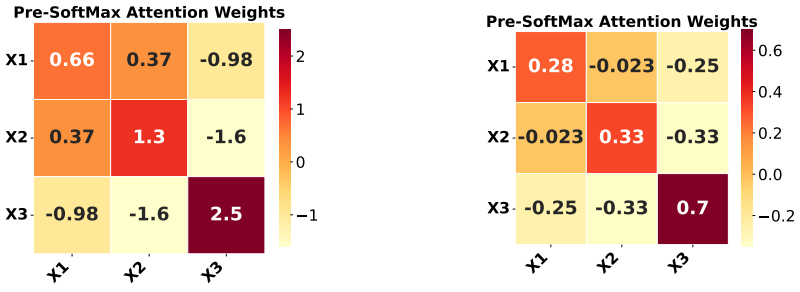
where X_2 and X_3 are the informative features, X_1 is a sensitive feature correlated with X_2, and Ber is Bernoulli distribution. Consequently, X_1 affects y indirectly through X_2, which corresponds to indirect discrimination in the algorithmic fairness literature. We train both FCorrTransformer and FCorrTransformer with CAR on this dataset. For counterfactual fairness evaluation, we adopt the Pairwise Comparison Metric (PCM) framework summarized in Czarnowska et al. (2021), which can be formulated as

$$PCM_{counterfactual}(\mathcal{D}) = \frac{1}{\zeta} \sum_{s_i \in \mathcal{S}} \sum_{\substack{s_j, s_k \in \mathcal{S} \\ s_j \neq s_k}} \mu(\phi(\mathcal{D}'_{s_i \to s_j}), \phi(\mathcal{D}'_{s_i \to s_k})) \tag{5}$$

where μ and ϕ denotes certain distance and score function to evaluate the fairness, ζ is the normalization factor, $\mathcal{D}'$ is the perturbed data given the input $\mathcal{D}$ by perturbing all the sensitive features, and the data partition $\mathcal{D}'_{s_i \to s_j}$ replaces its sensitive feature from s_i to perturbed category s_j. For the classification tasks, we adopt the Wasserstein distance along with several standard performance metrics as score functions, formulating Average Individual Fairness (AvgIF) (Huang et al., 2020a) and F1/AUROC/AUPRC Gap. Please refer to Appendix D for the formulation of the fairness metrics. Lower counterfactual fairness metrics indicate smaller differences in predictions and less bias across counterfactual samples, with a value of zero corresponding to counterfactual fairness. As shown in Table 1, evaluation on the synthetic dataset indicates that incorporating CAR enables the model to learn fair representations and effectively remove bias.

Table 1: Fairness metrics comparison on synthetic data

Model	AvgIF	F1 Gap	AUROC Gap	AUPRC Gap
FCorrTransformer	0.0085	**0.0000**	0.0058	0.0021
FCorrTransformer + CAR	**0.0000**	**0.0000**	**0.0000**	**0.0000**



(a) FCorrTransformer

(b) FCorrTransformer + CAR

Figure 2: Heatmap of pre-SoftMax attention weights on synthetic data

Figure 2 illustrates the pre-SoftMax attention matrix in FCorrTransformer and the effect of CAR. It is important to note that the post-SoftMax attention matrix can be difficult to interpret due to the additional row-wise normalization. In contrast, the pre-SoftMax attention matrix reflects the learned pairwise feature dependencies, and takes values over the entire real line. Lower pre-SoftMax values correspond to lower feature dependencies. In the standard FCorrTransformer (Figure 2a), the diagonal elements, interpretable as feature significance, exhibit the largest values. In particular, the significance of X_2 and X_3 is 1.3 and 2.5, respectively, given these features are the informative features. The off-diagonal elements capture pairwise feature dependencies. Notably, the interaction

between X_1 and X_2 has a value of 0.37, the largest among the off-diagonal entries, highlighting their strong dependency. In contrast, the interactions between X_1 and X_3, and between X_2 and X_3, are negative, indicating weak or negligible dependency, which is consistent with the data-generating design. With FCorrTransformer + CAR (Figure 2b), the attention matrix generally exhibits smaller pre-SoftMax values due to the effect of regularization. In particular, the unintended biased interaction between X_1 and X_2 is substantially suppressed. Furthermore, the interaction between X_1 and X_3 remains low, reflecting their independence. Overall, these results demonstrate that CAR effectively mitigates unintended biased dependencies in the learned feature relationships.

4.2 IMBALANCED CLASSIFICATION: BAF

For the imbalanced classification task, we use the Bank Account Fraud (BAF) dataset, a tabular dataset generated via a Generative Adversarial Network based on real-world fraud detection data (Jesus et al., 2022). We adopt the base variant of BAF[5], with approximately 1.1% labeled as fraudulent, reflecting the natural rarity of financial fraud in real-world settings. Following Jesus et al. (2022), we select *customer age* as the sensitive feature. This is a multi-class categorical feature representing age groups (e.g., 10s to 90s), with nine unique categories. As baseline models, we consider a standard feed-forward neural network (FFN), as well as attention-heavy tabular transformers, namely TabTransformer and FT-Transformer. In contrast, our proposed FCorrTransformer is an attention-light (FFN-heavy) architecture. We further apply CAR to both FT-Transformer and FCorrTransformer to evaluate their performance and counterfactual fairness. The definitions of performance and fairness metrics are provided in Appendix C and D. In addition to counterfactual fairness metrics, we report standard group fairness measures, including Demographic Parity Difference (DPD) and Equalized Odds (EqOdd). To minimize the impact of randomness, the mean and standard deviance of performance and fairness metrics across 10 independent trials are summarized in Tables 2 and 3. Mode details on experiment configurations, baseline results, and ablation study on the architecture design are summarized in Appendix F and G.1.

Table 2: Model performance comparison on BAF dataset

Subset	Model	Accuracy	F1 score	FPR	FNR	AUROC	AUPRC
Train	FFN	**0.9875** (0.0010)	0.1295 (0.0430)	**0.0024** (0.0015)	0.9127 (0.0373)	0.8669 (0.0029)	0.1325 (0.0033)
	TabTransformer	0.9814 (0.0025)	0.2014 (0.0074)	0.0100 (0.0029)	0.7880 (0.0362)	0.8698 (0.0011)	0.1350 (0.0028)
	FT-Transformer	0.9807 (0.0034)	**0.2410** (0.0069)	0.0114 (0.0041)	**0.7242** (0.0526)	**0.8967** (0.0009)	**0.1751** (0.0025)
	FT-Transformer + CAR	0.9807 (0.0044)	0.2353 (0.0101)	0.0112 (0.0051)	0.7326 (0.0645)	0.8950 (0.0015)	0.1712 (0.0041)
	FCorrTransformer	0.9808 (0.0044)	0.2219 (0.0097)	0.0109 (0.0051)	0.7535 (0.0619)	0.8906 (0.0013)	0.1568 (0.0037)
	FCorrTransformer + CAR	0.9811 (0.0058)	0.1926 (0.0618)	0.0103 (0.0069)	0.7823 (0.0966)	0.8863 (0.0006)	0.1463 (0.0036)
Test	FFN	**0.9878** (0.0012)	0.1228 (0.0408)	**0.0025** (0.0015)	0.9153 (0.0370)	0.8665 (0.0036)	0.1225 (0.0041)
	TabTransformer	0.9813 (0.0026)	0.1833 (0.0049)	0.0103 (0.0029)	0.8008 (0.0321)	0.8631 (0.0014)	0.1109 (0.0016)
	FT-Transformer	0.9809 (0.0034)	**0.2281** (0.0074)	0.0115 (0.0040)	**0.7313** (0.0529)	**0.8934** (0.0010)	**0.1581** (0.0027)
	FT-Transformer + CAR	0.9809 (0.0043)	0.2224 (0.0101)	0.0114 (0.0050)	0.7401 (0.0641)	0.8932 (0.0016)	0.1571 (0.0043)
	FCorrTransformer	0.9810 (0.0043)	0.2103 (0.0097)	0.0111 (0.0050)	0.7580 (0.0627)	0.8887 (0.0014)	0.1420 (0.0030)
	FCorrTransformer + CAR	0.9813 (0.0059)	0.1872 (0.0607)	0.0105 (0.0069)	0.7806 (0.0977)	0.8851 (0.0013)	0.1365 (0.0028)

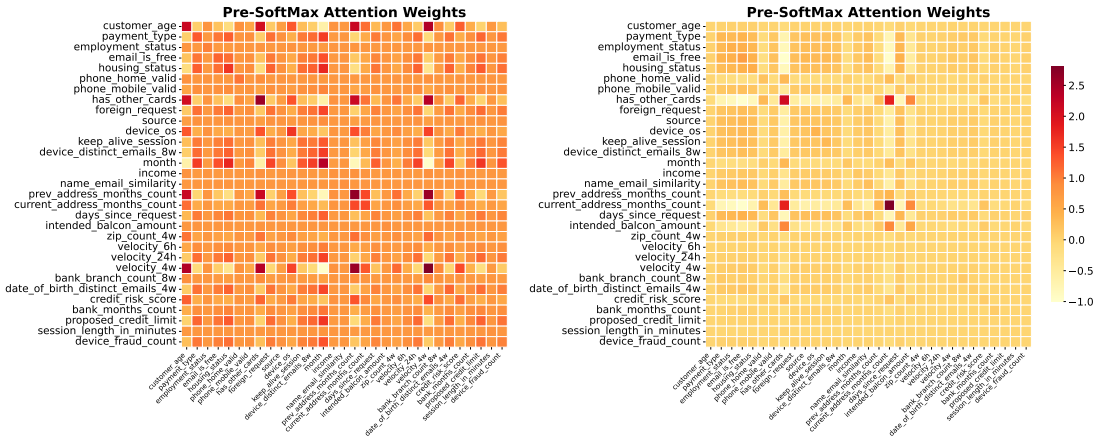
Table 2 shows that FT-Transformer and FCorrTransformer outperform FFN and TabTransformer in terms of F1 score, AUROC, and Area Under Precision Recall Curve (AUPRC). In contrast, metrics such as Accuracy, False Positive Rate (FPR), and False Negative Rate (FNR) are highly sensitive to the decision threshold, which is determined independently as discussed in Appendix G.1, and are therefore less informative in highly imbalanced classification settings. FCorrTransformer exhibits a noticeable performance gap compared to FT-Transformer, which is expected due to its simpler architectural design. As discussed in Appendix G.1, this gap can be partially mitigated by reintroducing

[5] https://www.kaggle.com/datasets/sgpjesus/bank-account-fraud-dataset-neurips-2022

Table 3: Fairness metrics comparison on BAF dataset

Subset	Model	DPD	EqOdd	F1 Gap	AUROC Gap	AUPRC Gap
Train	FFN	0.0585 (0.0246)	**0.2886** (0.1149)	0.7163 (0.1039)	0.0284 (0.0069)	0.0478 (0.0294)
	TabTransformer	0.1379 (0.0426)	0.5887 (0.1528)	0.5948 (0.0457)	0.0644 (0.0287)	0.1602 (0.0864)
	FT-Transformer	0.1352 (0.0425)	0.6244 (0.1300)	0.1745 (0.0442)	0.0055 (0.0019)	0.0082 (0.0024)
	FT-Transformer + CAR	0.0867 (0.0233)	0.4451 (0.0550)	0.1701 (0.0529)	0.0034 (0.0012)	0.0057 (0.0022)
	FCorrTransformer	0.0968 (0.0336)	0.4914 (0.0837)	0.3715 (0.1400)	0.0157 (0.0065)	0.0670 (0.0175)
	FCorrTransformer + CAR	**0.0541** (0.0358)	0.3585 (0.1823)	**0.0000** (0.0000)	**0.0000** (0.0000)	**0.0000** (0.0000)
Test	FFN	**0.0539** (0.0320)	**0.6286** (0.1143)	0.6244 (0.0639)	0.0447 (0.0107)	0.0578 (0.0180)
	TabTransformer	0.1525 (0.0680)	0.8286 (0.0571)	0.4242 (0.0551)	0.0455 (0.0253)	0.1396 (0.0573)
	FT-Transformer	0.1516 (0.0390)	0.8086 (0.0677)	0.1848 (0.0424)	0.0159 (0.0067)	0.0094 (0.0035)
	FT-Transformer + CAR	0.1214 (0.0454)	0.8238 (0.0604)	0.1768 (0.0546)	0.0120 (0.0086)	0.0075 (0.0033)
	FCorrTransformer	0.1241 (0.0425)	0.7429 (0.0857)	0.2614 (0.1038)	0.0134 (0.0079)	0.0742 (0.0297)
	FCorrTransformer + CAR	0.1064 (0.0529)	0.6590 (0.2358)	**0.0000** (0.0000)	**0.0000** (0.0000)	**0.0000** (0.0000)

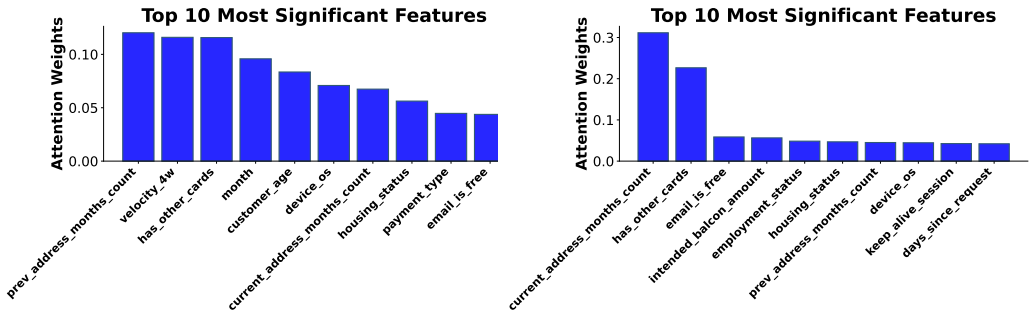
residual connections; however, doing so weakens the model's fairness guarantees. When CAR is applied, FT-Transformer does not suffer significant performance degradation and achieves comparable performance. In contrast, CAR introduces a more noticeable performance drop for FCorrTransformer. More importantly, the fairness results in Table 3 reveal substantial variation across models. While CAR generally reduces bias in FT-Transformer, significant residual bias remains. For FCorrTransformer, although group fairness is not guaranteed, the combination of FCorrTransformer + CAR achieves strong counterfactual fairness with only minimal performance loss. In real-world applications, this implies that decisions of standard transformers depend on the sensitive feature, even with properly regularized attention matrices. In contrast, FCorrTransformer ensures that predictions are invariant to the sensitive feature, preventing it from directly affecting the decisions. This property is particularly valuable in applications where strict fairness is required by regulatory standards.



(a) FCorrTransformer (b) FCorrTransformer + CAR

Figure 3: Heatmap of Pre-SoftMax Attention Weights on BAF Data

To better understand the mechanism of CAR, Figure 3 visualizes the average attention matrices over the training set. In contrast to the classical attention mechanism used in FT-Transformer (Figure 8a in Appendix G.1), whose attention weights lack clear statistical interpretation, the attention patterns



(a) FCorrTransformer

(b) FCorrTransformer + CAR

Figure 4: Top 10 feature significance on BAF data

in FCorrTransformer are more structured and closely resemble feature dependencies. As shown in Figure 3a, diagonal elements can be interpreted as the significance of individual features, while off-diagonal elements capture pairwise feature dependencies. For the purpose of visualization, the sensitive feature is placed in the first position in all experiments. The attention matrix indicates that the sensitive feature, customer age, receives high self-attention and exhibits strong dependencies with several informative features, enabling the model to exploit sensitive information. As discussed in Subsection 4.1, CAR suppresses unintended biased dependencies and reallocates attention toward fair features. This effect is clearly illustrated in Figure 3b, where attention to the sensitive feature is substantially suppressed. Figure 4 reports the top-10 most significant features identified by FCorrTransformer, which largely agree with the feature importance extracted from LightGBM (Figure 7 in Appendix G.1), suggesting the feasibility of interpreting attention weights in FCorrTransformer as feature dependency.

Finally, due to its linear architecture and absence of high-dimensional embeddings, FCorrTransformer is computationally efficient. FCorrTransformer has significantly lower parameter count and memory usage compared to FT-Transformer (Table 9 in Appendix G.1), enabling more efficient training and deployment. This efficiency stems from its attention-light design: the attention only consists 5.7% of parameters in FCorrTransformer, while it accounts for 99.9% of parameters in FT-Transformer.

5 Conclusion

In this work, we proposed a novel framework for achieving counterfactual fairness in tabular learning through the FCorrTransformer and CAR. By moving away from high-dimensional embeddings and complex multi-head projections, we demonstrated that an attention-light architectural design can provide direct and interpretable representations of feature dependencies while maintaining strong predictive performance. The proposed input augmentation strategy further enables efficient counterfactual learning. Empirical results on imbalanced financial and insurance datasets show that our framework effectively suppresses biased dependencies and centralizes on fair features. In contrast to baseline transformer architectures such as FT-Transformer, which exhibit persistent residual bias, FCorrTransformer+CAR achieves strong counterfactual fairness guarantees with minimal performance degradation, making it particularly suitable for regulatory-sensitive applications. Despite these promising results, our approach has limitations. In its current form, CAR primarily addresses fairness for categorical sensitive features and does not directly extend to continuous sensitive features such as income, where defining meaningful counterfactual interventions is more challenging. Addressing continuous and multi-dimensional sensitive features, as well as extending the framework to dynamic or sequential tabular settings, remains an important direction for future work. Overall, this study demonstrates that imposing architectural constraints together with explicit attention regularization offers a practical, interpretable pathway toward responsible ML in high-stakes societal domains.

AI USE STATEMENT

In this work, we used generative AI tools for *assisting in the writing of proofs*. We have not used generative AI tools for *generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results*. Additionally, we used generative AI tools for *editing the research paper to improve readability and cleaning up the implementation/experiment code (create or edit software code)*. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not raise any significant ethical concerns. We do not collect or expose personally identifiable information. The proposed methods are intended for research purposes and do not involve human subjects, interventions, or other potentially harmful experimental procedures.

REPRODUCIBILITY STATEMENT

We are currently cleaning and organizing the codebase and experimental scripts to facilitate reproducibility. Upon acceptance, we will publicly release the implementation, training and evaluation scripts, experimental configurations, and instructions for reproducing the reported results. The proof of the theoretical results is in Appendix B.

REFERENCES

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudik, John Langford, and Hanna Wallach. A Reductions Approach to Fair Classification. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 60–69. PMLR, July 2018.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Proceedings of the International Conference on Knowledge Discovery & Data Mining, pp. 2623–2631. Association for Computing Machinery, 7 2019. doi: 10.1145/3292500.3330701.

Sercan Ö Arik and Tomas Pfister. TabNet: Attentive Interpretable Tabular Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6679–6687, May 2021. doi: 10.1609/aaai.v35i8.16826.

Giuseppe Attanasio, Debora Nozza, Dirk Hovy, and Elena Baralis. Entropy-based Attention Regularization Frees Unintended Bias Mitigation from Lists. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 1105–1119, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.88.

Toon Calders and Sicco Verwer. Three naive Bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21(2):277–292, September 2010. ISSN 1573-756X. doi: 10.1007/s10618-010-0190-x.

Paula Czarnowska, Yogarshi Vyas, and Kashif Shah. Quantifying Social Biases in NLP: A Generalization and Empirical Comparison of Extrinsic Fairness Metrics. *Transactions of the Association for Computational Linguistics*, 9:1249–1267, November 2021. ISSN 2307-387X. doi: 10.1162/tacl_a_00425.

Mengnan Du, Subhabrata Mukherjee, Guanchu Wang, Ruixiang Tang, Ahmed Awadallah, and Xia Hu. Fairness via Representation Neutralization. In *Advances in Neural Information Processing Systems*, volume 34, pp. 12091–12103. Curran Associates, Inc., 2021.

Yacine Gaci, Boualem Benatallah, Fabio Casati, and Khalid Benabdeslem. Debiasing Pretrained Text Encoders by Paying Attention to Paying Attention. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9582–9602, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.651.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting Deep Learning Models for Tabular Data. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18932–18943. Curran Associates, Inc., 2021.

Farsheed Haque, Zhe Fu, Depeng Xu, Shuhan Yuan, and Xi Niu. Fine-tuning LLMs with Cross-Attention-based Weight Decay for Bias Mitigation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 15785–15798, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-emnlp.854.

Moritz Hardt, Eric Price, and Nathan Srebro. Equality of Opportunity in Supervised Learning. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. TabLLM: Few-shot Classification of Tabular Data with Large Language Models. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pp. 5549–5581. PMLR, April 2023.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2016.

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, January 2025. ISSN 1476-4687. doi: 10.1038/s41586-024-08328-6.

Po-Sen Huang, Huan Zhang, Ray Jiang, Robert Stanforth, Johannes Welbl, Jack Rae, Vishal Maini, Dani Yogatama, and Pushmeet Kohli. Reducing Sentiment Bias in Language Models via Counterfactual Evaluation. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 65–83, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.7.

Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. TabTransformer: Tabular Data Modeling Using Contextual Embeddings, December 2020b.

Sérgio Jesus, José Pombal, Duarte Alves, André F. Cruz, Pedro Saleiro, Rita P. Ribeiro, João Gama, and Pedro Bizarro. Turning the tables: Biased, imbalanced, dynamic tabular datasets for ML evaluation. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pp. 33563–33575, Red Hook, NY, USA, November 2022. Curran Associates Inc.

Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-Aware Classifier with Prejudice Remover Regularizer. In Peter A. Flach, Tijl De Bie, and Nello Cristianini (eds.), *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, pp. 35–50, Berlin, Heidelberg, 2012. Springer. doi: 10.1007/978-3-642-33486-3_3.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual Fairness. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. Gender Bias in Neural Natural Language Processing. In Vivek Nigam, Tajana Ban Kirigin, Carolyn Talcott, Joshua Guttman, Stepan Kuznetsov, Boon Thau Loo, and Mitsuhiro Okada (eds.), *Logic, Language, and Security: Essays Dedicated to Andre Scedrov on the Occasion of His 65th Birthday*, pp. 189–202. Springer International Publishing, Cham, 2020. doi: 10.1007/978-3-030-62077-6_14.

Jing Ma, Ruocheng Guo, Aidong Zhang, and Jundong Li. Learning for Counterfactual Fairness from Observational Data. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, pp. 1620–1630, New York, NY, USA, August 2023. Association for Computing Machinery. doi: 10.1145/3580305.3599408.

Ninareh Mehrabi, Umang Gupta, Fred Morstatter, Greg Ver Steeg, and Aram Galstyan. Attributing Fair Decisions with Attention Interventions. In Apurv Verma, Yada Pruksachatkun, Kai-Wei Chang, Aram Galstyan, Jwala Dhamala, and Yang Trista Cao (eds.), *Proceedings of the 2nd Workshop on Trustworthy Natural Language Processing (TrustNLP 2022)*, pp. 12–25, Seattle, U.S.A., July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.trustnlp-1.2.

Ece Çiğdem Mutlu, View Profile, Niloofar Yousefi, View Profile, Ozlem Ozmen Garibay, and View Profile. Contrastive Counterfactual Fairness in Algorithmic Decision-Making. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, ACM Conferences, pp. 499–507, New York, NY, United States, July 2022. Association for Computing Machinery. doi: 10.1145/3514094.3534143.

Yusu Qian, Urwa Muaz, Ben Zhang, and Jae Won Hyun. Reducing Gender Bias in Word-Level Language Models with a Gender-Equalizing Loss Function. In Fernando Alva-Manchego, Eunsol Choi, and Daniel Khashabi (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pp. 223–228, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-2031.

Yao Qiang, Chengyin Li, Prashant Khanduri, and Dongxiao Zhu. Fairness-Aware Vision Transformer via Debiased Self-Attention. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 358–376, Cham, 2025. Springer Nature Switzerland. doi: 10.1007/978-3-031-72913-3_20.

Zhiyu Quan, Changyue Hu, Panyi Dong, and Emiliano A. Valdez. Improving Business Insurance Loss Models by Leveraging InsurTech Innovation. *North American Actuarial Journal*, 29(2): 247–274, April 2025. ISSN 1092-0277. doi: 10.1080/10920277.2024.2400648.

Yongming Rao, Guangyi Chen, Jiwen Lu, and Jie Zhou. Counterfactual Attention Learning for Fine-Grained Visual Categorization and Re-Identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1025–1034, 2021.

Jake Robertson, Noah Hollmann, Samuel Müller, Noor Awad, and Frank Hutter. FairPFN: A Tabular Foundation Model for Causal Fairness, June 2025.

Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C. Bayan Bruss, and Tom Goldstein. SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training, June 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Fei Wang, James Y. Huang, Tianyi Yan, Wenxuan Zhou, and Muhao Chen. Robust Natural Language Understanding with Residual Attention Debiasing. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 504–519, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.32.

Jiahuan Yan, Bo Zheng, Hongxia Xu, Yiheng Zhu, Danny Z. Chen, Jimeng Sun, Jian Wu, and Jintai Chen. Making Pre-trained Language Models Great on Tabular Prediction, March 2024.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P. Gummadi. Fairness Constraints: Mechanisms for Fair Classification. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 962–970. PMLR, April 2017.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating Unwanted Biases with Adversarial Learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, pp. 335–340, New York, NY, USA, December 2018. Association for Computing Machinery. doi: 10.1145/3278721.3278779.

Bingzhao Zhu, Xingjian Shi, Nick Erickson, Mu Li, George Karypis, and Mahsa Shoaran. XTab: Cross-table pretraining for tabular transformers. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *ICML'23*, pp. 43181–43204, Honolulu, Hawaii, USA, July 2023. JMLR.org.

A NOTATIONS

Table 4: Notations used in the formulation

Notation	Description
$\mathcal{D}$	a dataset
$(\boldsymbol{X}, \boldsymbol{y})$	a pair of feature matrix $\boldsymbol{X}$ and response vector $\boldsymbol{y}$
N	number of observations
$\mathcal{X}$/$\mathcal{Y}$	feature and response space
$\boldsymbol{X}_s$	sensitive feature matrix
$\mathcal{X}_s$	sensitive feature space
$\boldsymbol{X}_r$	non-sensitive feature matrix
$\mathcal{X}_r$	non-sensitive feature space
p	number of features in feature matrix $\boldsymbol{X}$
s_i	sensitive demographic group i
$\mathcal{S}$	sensitive demographic group space
C^s	Number of categories for (discrete) sensitive feature S
f	a ML model
$\hat{\boldsymbol{y}}$	model predictions
$\boldsymbol{E}$	feature embedding
$\boldsymbol{A}$	attention matrix
$\mathcal{L}_{CAR}$	CAR loss function
σ	index of sensitive feature in feature matrix
$\mathcal{L}$	overall loss function
$\mathcal{L}_{perf}$	performance loss function
λ	regularization coefficient for CAR
μ	distance function for fairness evaluation
ϕ	score function for fairness evaluation
ζ	normalization factor for generalized fairness metrics
$\mathcal{D}'$	a perturbed data
$\boldsymbol{h}$	post-attention representation
e	simplified feature embedding
g	attention mapping
L	Lipschitz continuity
U	convergence bound

B PROOF

By the construction of FCorrTransformer, the model f can be decomposed into

$$f = f^{post} \circ \text{Attention} \circ f^{pre}$$

where f^{pre} and f^{post} denote the pre-attention layers and post-attention prediction heads respectively, and Attention is the attention mechanism following Equation 1. One thing to note is that different from the standard tabular transformer architectures, there exists no residual connection for the attention mechanism, and all information passes through the attention mechanism.

Assumption 1 (Post-attention Lipschitz Continuity)**.** *There exists a constant $L^{post} > 0$, such that the post-attention prediction head f^{post} of the model f, with post-attention representations $\boldsymbol{h}_1, \boldsymbol{h}_2 \in \mathbb{R}^{p \times p}$ and response $\boldsymbol{y} \in \mathcal{Y}$, satisfies*

$$|\mathcal{L}(f^{post}(\boldsymbol{h}_1), \boldsymbol{y}) - \mathcal{L}(f^{post}(\boldsymbol{h}_2), \boldsymbol{y})| \leq L^{post} \|\boldsymbol{h}_1 - \boldsymbol{h}_2\|$$

Assumption 2 (Embedding Jacobian Condition)**.** *The feature embedding vector $\boldsymbol{E}(\boldsymbol{x}_n)$, ($\boldsymbol{x}_n = [\boldsymbol{x}_r, x_s]$), is defined within a compact set $\mathcal{E} \subset \mathbb{R}^p$. Denote $e(x_s) = \boldsymbol{E}(\boldsymbol{x}_n)_\sigma \in \mathbb{R}$ as the embedding of the sensitive feature. There exists such attention mapping $g : \mathbb{R} \to \mathbb{R}^p$ so that the attention weights can be expressed as*

$$\boldsymbol{A}(x_s)_{:,\sigma} = g \circ e(x_s)$$

and the L^2 norm of its Jacobian is strictly bounded such that $\|g' \circ e(x_s)\|_2 \geq \xi > 0$.

The bound condition is generally satisfied in practice because the partial derivative of the post-SoftMax attention weights, $\boldsymbol{A}(x_s)_{:,\sigma}$, with respect to the sensitive feature embedding $e(x_s)$ vanishes only when the corresponding embedding is zero. This condition is rarely encountered in practice, since at least some features are expected to be informative, while layer normalization further discourages feature embeddings from collapsing to zero.

B.1 Proof of Theorem 1

Given the model decomposition, the difference in model predictions can be attributed to post-attention prediction head f^{post} and post-attention representation $\boldsymbol{h}$ accordingly, where the post-attention representation can be further decomposed into attention weights $\boldsymbol{A}$ and embeddings $\boldsymbol{E}$ given the formulation of attention mechanism in Equation 1. For the simplicity of formulation, we denote $\boldsymbol{A}(s_i)$, $\boldsymbol{E}(s_i)$, and $\boldsymbol{h}(s_i)$ as the first-layer attention matrix, the embedding, and the post-attention representation with non-sensitive features $\boldsymbol{x}_r$ and sensitive category s_i, respectively.

Regarding the difference in the attention matrix, given the condition of CAR loss,

$$\mathcal{L}_{CAR}(\boldsymbol{x}_n) = \frac{1}{C^s}\sum_{i=1}^{C^s} \|\boldsymbol{A}(x_s)_{:,\sigma} - \boldsymbol{A}(s_i)_{:,\sigma}\|_2^2 \leq \epsilon$$

where $\boldsymbol{x}_n = [\boldsymbol{x}_r, x_s]$, we have $\|\boldsymbol{A}(x_s)_{:,\sigma} - \boldsymbol{A}(s_i)_{:,\sigma}\|_2^2 \leq C^s\epsilon$ for $\forall s_i \in \mathcal{S}$. Thus, by triangle inequality,

$$\|\boldsymbol{A}(s_i)_{:,\sigma} - \boldsymbol{A}(s_j)_{:,\sigma}\|_2 \leq \|\boldsymbol{A}(x_s)_{:,\sigma} - \boldsymbol{A}(s_i)_{:,\sigma}\|_2 + \|\boldsymbol{A}(x_s)_{:,\sigma} - \boldsymbol{A}(s_j)_{:,\sigma}\|_2 \leq 2\sqrt{C^s\epsilon}$$

Regarding the difference in the embedding, given that the permutation is only performed on sensitive feature, $\boldsymbol{E}(s_i)_k = \boldsymbol{E}(s_j)_k$, for $\forall k \neq \sigma$, where σ is the index of sensitive feature. Thus,

$$\|\boldsymbol{E}(s_i) - \boldsymbol{E}(s_j)\|_2 = |e(s_i) - e(s_j)|$$

If we write the attention weights as attention mapping $\boldsymbol{A}(s_i)_{:,\sigma} = g \circ e(s_i)$ and the mapping follows Assumption 2, by inverse function theorem, we know g^{-1} exists and

$$|e(s_i) - e(s_j)| \leq \frac{1}{\xi}\|\boldsymbol{A}(s_i)_{:,\sigma} - \boldsymbol{A}(s_j)_{:,\sigma}\|_2 \leq \frac{2\sqrt{C^s\epsilon}}{\xi}$$

Thus, $\|e(s_i) - e(s_j)\|_2 \leq \dfrac{2\sqrt{C^s\epsilon}}{\xi}$. Furthermore, given the formulation of attention mechanism in Equation 1, the post-attention representation $\boldsymbol{h}(s_i)$ is a continuously differentiable mapping of embedding $\boldsymbol{E}(s_i)$ in a compact set (Assumption 2) since the K/Q/V linear projections are removed. Thus, the post-attention representation $\boldsymbol{h}(s_i)$ is Lipschitz continuous with some constant L^{attn} such that

$$\|\boldsymbol{h}(s_i) - \boldsymbol{h}(s_j)\|_2 \leq L^{attn}\|e(s_i) - e(s_j)\|_2 \leq \frac{2L^{attn}\sqrt{C^s\epsilon}}{\xi}$$

Finally, given the Assumption 1,

$$|f(\boldsymbol{x}_r, s_i) - f(\boldsymbol{x}_r, s_j)| \leq L^{post}\|\boldsymbol{h}(s_i) - \boldsymbol{h}(s_j)\|_2 \leq \frac{2L^{post}L^{attn}\sqrt{C^s\epsilon}}{\xi}$$

Let $U = \dfrac{2L^{post}L^{attn}\sqrt{C^s}}{\xi}$, we complete the proof $|f(\boldsymbol{x}_r, s_i) - f(\boldsymbol{x}_r, s_j)| \leq U\sqrt{\epsilon}$.

C Performance Evaluation Metrics

1. Accuracy Score

$$Accuracy(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}_{\{y_n = \hat{y}_n\}}$$

2. F1 Score

$$F1(\boldsymbol{y}, \hat{\boldsymbol{y}}) = 2\frac{Pr(\boldsymbol{y}, \hat{\boldsymbol{y}})Re(\boldsymbol{y}, \hat{\boldsymbol{y}})}{Pr(\boldsymbol{y}, \hat{\boldsymbol{y}}) + Re(\boldsymbol{y}, \hat{\boldsymbol{y}}}$$

with precision score $Pr(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{TP}{TP+FP}$, recall score $Re(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{TP}{TP+FN}$ where true Positive $TP = \sum_{n=1}^{N} \mathbb{1}_{\{y_n=1\}}\mathbb{1}_{\{\hat{y}_n=1\}}$, false positive $FP = \sum_{n=1}^{N} \mathbb{1}_{\{y_n=0\}}\mathbb{1}_{\{\hat{y}_n=1\}}$, and false negative $FN = \sum_{n=1}^{N} \mathbb{1}_{\{y_n=1\}}\mathbb{1}_{\{\hat{y}_n=0\}}$.

3. False Positive Rate (FPR)

$$FPR(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{\sum_{n=1}^{N} \mathbb{1}_{\{y_n=0\}}\mathbb{1}_{\{\hat{y}_n=1\}}}{\sum_{n=1}^{N} \mathbb{1}_{\{y_n=0\}}}$$

4. False Negative Rate (FNR)

$$FNR(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{\sum_{n=1}^{N} \mathbb{1}_{\{y_n=1\}}\mathbb{1}_{\{\hat{y}_n=0\}}}{\sum_{n=1}^{N} \mathbb{1}_{\{y_n=1\}}}$$

5. Area Under Precision-Recall Curve (AUPRC)

$$AUPRC(\boldsymbol{y}, \hat{\boldsymbol{y}}^p) = \sum_i (Re^i(\boldsymbol{y}, \hat{\boldsymbol{y}}^p) - Re^{i-1}(\boldsymbol{y}, \hat{\boldsymbol{y}}p))Pr^i(\boldsymbol{y}, \hat{\boldsymbol{y}}^p)$$

where $\hat{\boldsymbol{y}}^p$ is the predicted probabilities for the positive category, Re_r^i and Pr_r^i denotes the recall and precision score with i-th thresholds, respectively.

6. Area Under the Receiver Operating Characteristic Curve (AUROC)

$$AUROC(\boldsymbol{y}, \hat{\boldsymbol{y}}^p) = \frac{\sum_{m:y_m=1}\sum_{n:y_n=0} \mathbb{1}_{\{\hat{y}_m^p > \hat{y}_n^p\}}}{\sum_{n=1}^{N} \mathbb{1}_{\{y_n=1\}} \sum_{n=1}^{N} \mathbb{1}_{\{y_n=0\}}}$$

7. Gini index

$$Gini(\boldsymbol{y}, \hat{\boldsymbol{y}}) = 1 - \frac{2}{N-1}\left(N - \frac{\sum_{n=1}^{N} ny_{[n]}}{\sum_{n=1}^{N} ny_{[n]}}\right)$$

where $y_{[n]}$ is n-th value of $\boldsymbol{y}$ based on ranks of $\hat{\boldsymbol{y}}$.

8. Percentage Error (PE)

$$PE(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{\sum_{n=1}^{N}(\hat{y}_n - y_n)}{\sum_{n=1}^{N} y_n}$$

9. Root Mean Squared Error (RMSE)

$$RMSE(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\hat{y}_n - y_n)^2}$$

10. Mean Absolute Error (MAE)

$$MAE(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{N}\sum_{n=1}^{N}|\hat{y}_n - y_n|$$

D FAIRNESS EVALUATION METRICS

1. Demographic Parity Difference (DPD) Agarwal et al. (2018)

$$DPD(f, \mathcal{D}) = \max_{s_j, s_k} |\mathbb{E}(\hat{\boldsymbol{y}}|\boldsymbol{X}_s = s_j) - \mathbb{E}(\hat{\boldsymbol{y}}|\boldsymbol{X}_s = s_k)|$$

2. Equalized Odds Difference (EqOdd) (Hardt et al., 2016)

$$EqOdd(f, \mathcal{D}) = \max_{s_j, s_k} \max_{y \in \{0,1\}} |P(\hat{\boldsymbol{y}} = 1|\boldsymbol{X}_s = s_j, \boldsymbol{y} = y) - P(\hat{\boldsymbol{y}} = 1|\boldsymbol{X}_s = s_k, \boldsymbol{y} = y)|$$

3. F1 Gap

$$F1_Gap(f, \mathcal{D}) = \frac{1}{C^s(C^s - 1)} \sum_{s_i \in \mathcal{S}} \sum_{\substack{s_j, s_k \in \mathcal{S} \\ s_j \neq s_k}} |F1(\boldsymbol{y}'_{s_i \to s_j}, \hat{\boldsymbol{y}}'_{s_i \to s_j}) - F1(\boldsymbol{y}'_{s_i \to s_k}, \hat{\boldsymbol{y}}'_{s_i \to s_k})|$$

where $\hat{\boldsymbol{y}}'_{s_i \to s_j} = f(\mathcal{D}'_{s_i \to s_j})$ and $\hat{\boldsymbol{y}}'_{s_i \to s_k} = f(\mathcal{D}'_{s_i \to s_k})$ are the predictions of the corresponding subsets in the perturbed dataset.

4. AUROC Gap

$$AUROC_Gap(f, \mathcal{D}) = \frac{1}{C^s(C^s - 1)} \sum_{s_i \in \mathcal{S}} \sum_{\substack{s_j, s_k \in \mathcal{S} \\ s_j \neq s_k}} |AUROC(\boldsymbol{y}'_{s_i \to s_j}, \hat{\boldsymbol{y}}'_{s_i \to s_j}) - AUROC(\boldsymbol{y}'_{s_i \to s_k}, \hat{\boldsymbol{y}}'_{s_i \to s_k})|$$

5. AUPRC Gap

$$AUPRC_Gap(f, \mathcal{D}) = \frac{1}{C^s(C^s - 1)} \sum_{s_i \in \mathcal{S}} \sum_{\substack{s_j, s_k \in \mathcal{S} \\ s_j \neq s_k}} |AUPRC(\boldsymbol{y}'_{s_i \to s_j}, \hat{\boldsymbol{y}}'_{s_i \to s_j}) - AUPRC(\boldsymbol{y}'_{s_i \to s_k}, \hat{\boldsymbol{y}}'_{s_i \to s_k})|$$

6. RMSE Gap

$$RMSE_Gap(f, \mathcal{D}) = \frac{1}{C^s(C^s - 1)} \sum_{s_i \in \mathcal{S}} \sum_{\substack{s_j, s_k \in \mathcal{S} \\ s_j \neq s_k}} |RMSE(\boldsymbol{y}'_{s_i \to s_j}, \hat{\boldsymbol{y}}'_{s_i \to s_j}) - RMSE(\boldsymbol{y}'_{s_i \to s_k}, \hat{\boldsymbol{y}}'_{s_i \to s_k})|$$

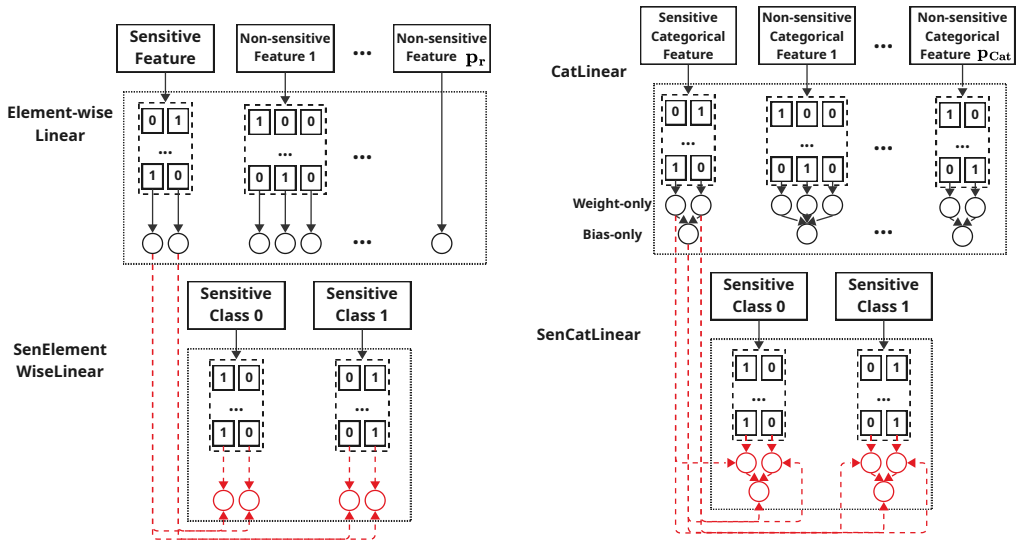
7. MAE Gap

$$MAE_Gap(f, \mathcal{D}) = \frac{1}{C^s(C^s - 1)} \sum_{s_i \in \mathcal{S}} \sum_{\substack{s_j, s_k \in \mathcal{S} \\ s_j \neq s_k}} |MAE(\boldsymbol{y}'_{s_i \to s_j}, \hat{\boldsymbol{y}}'_{s_i \to s_j}) - MAE(\boldsymbol{y}'_{s_i \to s_k}, \hat{\boldsymbol{y}}'_{s_i \to s_k})|$$

E DETAILS ON CAR MODIFICATIONS

The primary objective of input augmentation-based CAR is to derive all sensitive feature attentions within a single forward pass, thereby mitigating the computational redundancy inherent in CDA-based methods while maintaining high fidelity to the original CDA process.

However, a significant challenge arises in the standard FCorrTransformer: if the model dimension is naively expanded according to the number of counterfactual sensitive categories (i.e., from p features to $p + C^s$ features), the original sensitive feature and its augmented counterfactual categories are treated as distinct features with separate embedding parameters. Such a configuration is undesirable, as regularization applied to the naively expanded attention matrix no longer enforces fair feature dependencies across counterfactual realizations, potentially allowing biased information to bypass the attention mechanism via independent parameter paths. Thus, we introduce three specialized modules, *SenLayerNorm*, *SenElementWiseLinear*, and *SenCatLinear*, which adapt their non-counterfactual counterparts to share weights across the augmented feature space.



(a) SenElementWiseLinear (b) SenCatLinear

Figure 5: Illustration of SenElementWiseLinear and SenCatLinear on binary sensitive feature

As illustrated in Figure 5, during training, SenElementWiseLinear and SenCatLinear identify the indices of sensitive features and extract the corresponding weight and bias parameters from the standard Element-Wise Linear and CatLinear layers. These parameters and gradients are shared between the standard and specialized modules. For the first categorical embedding layer and the CatLinear layers, the indices correspond to one-hot encoded category indices, while for all other layers, the indices directly match the sensitive feature positions in the input. For SenLayerNorm, instead of normalizing over the augmented input space with counterfactual categories, we compute the mean and standard deviation only over the original input. The corresponding weight and bias parameters of the sensitive features are then extracted and applied to the augmented sensitive categories, ensuring consistency between input augmentation and CDA. Importantly, these sensitive layers reuse existing parameters and therefore introduce no additional learnable parameters.

For the attention mechanism and SoftMax with CAR, the dot-product can naturally handle expanded inputs since no linear projection is introduced. We retain the scale of $\sqrt{p}$ in the dot-product operation. The SoftMax is applied over the augmented space rather than the original feature space. In our prior experiments, SoftMax operation over the original feature space leads to attention weight explosion, especially in regression settings. As a result, attention weights under input augmentation are not strictly identical to those with CDA, but they converge to similar distributions after a few training iterations and at inference. After the FCorr encoder blocks, we discard the activations of the augmented sensitive categories and retain only the original feature activations as input to the prediction head.

Intuitively, this design reduces training time, as most of the computational cost in CDA arises from repeated attention computations over non-sensitive features. In our experiments discussed in Section 4, input augmentation reduces the average training time per epoch from 21.46 seconds to 19.50 seconds for classification and from 52.98 seconds to 14.40 seconds for regression under identical settings. The magnitude of this reduction is closely related to the dimensionality of the input and the cardinality of sensitive features. The classification task has low feature dimensionality, while the regression task has much higher dimensionality, leading to a more significant speedup. Overall, input augmentation provides a more efficient way to estimate variations in dependencies between sensitive and non-sensitive features. During inference, since input augmentation is disabled, this mechanism introduces no additional computational overhead.

While our formulation and experiments focus on the single sensitive feature setting, the framework naturally extends to multiple sensitive features by augmenting all combinations of sensitive cate-

gories and applying the same parameter-sharing strategy. Our implementation fully supports the multi-sensitive-feature case.

F Impact of Embedding Dimension on FCorrTransformer

To further justify the motivation for the one-dimensional feature embedding structure introduced in Subsection 3.1, we conduct additional empirical experiments to investigate the impact of embedding dimension on model performance and to evaluate whether one-dimensional embeddings are sufficient for transformer-based tabular learning. Specifically, we vary the embedding dimension of FCorrTransformer in the BAF experiment described in Subsection 4.2 from 1 to 32 and examine the same predictive performance and fairness metrics. To mitigate the effects of randomness in neural network training, all experiments are repeated over 10 independent trials.

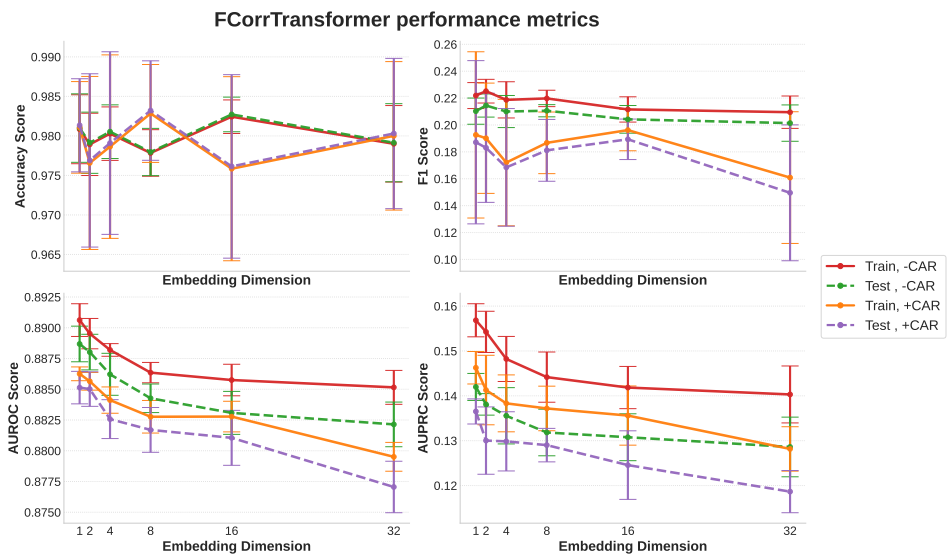
Figure 6 summarizes the performance and fairness metrics across different embedding dimensions, while Tables 5 and 6 provide the detailed numerical results. For predictive performance, given the highly imbalanced nature of the BAF dataset, accuracy and F1 score are sensitive to the choice of decision threshold, whereas AUROC and AUPRC provide more robust assessments of model performance. As shown in Figure 6a, increasing the embedding dimension generally leads to a deterioration in FCorrTransformer performance, regardless of whether CAR regularization is applied. These results provide empirical support for the use of one-dimensional embeddings in FCorrTransformer and suggest that high-dimensional feature embeddings, which are commonly adopted in transformer architectures, may be unnecessary for tabular learning tasks. The fairness results, presented in Figure 6b, exhibit no clear trend with respect to embedding dimension. This observation is expected, as predictive performance and fairness are not directly linked. Nevertheless, CAR consistently reduces direct discrimination across all embedding dimension settings, demonstrating that the effectiveness of the proposed fairness regularization is largely independent of the embedding dimension.

G Details on Experiments

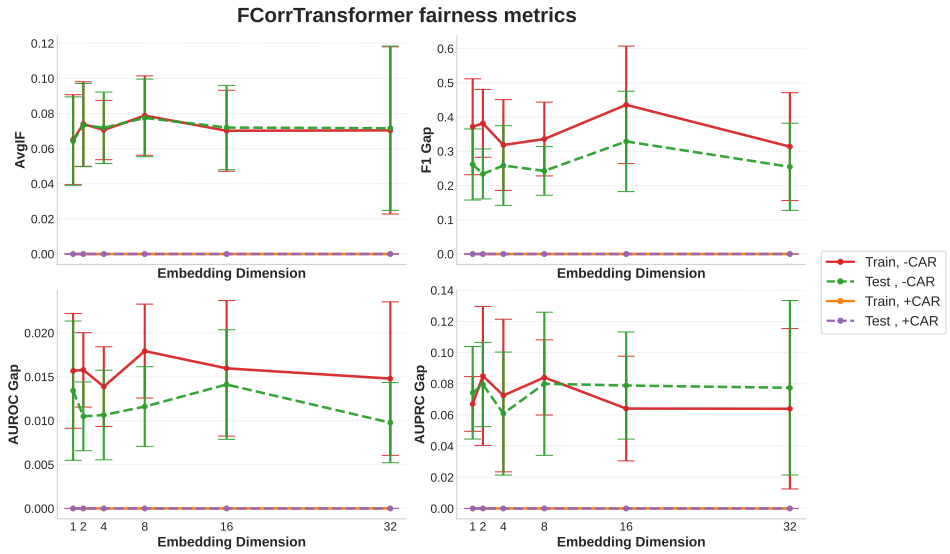
G.1 Imbalanced Classification: BAF

In the BAF experiment, to establish an independent decision threshold suitable for imbalanced classification, since the default threshold of 0.5 is not appropriate, and to provide a strong performance baseline, we train a LightGBM model (Ke et al., 2017) with default hyperparameter settings. The decision threshold is selected by maximizing the training F1 score of the LightGBM model, resulting in a threshold of 0.12. This partially explains the superior performance of LightGBM in terms of Accuracy, FPR, and FNR, as these metrics are highly sensitive to the choice of decision threshold in imbalanced classification settings. The performance of the additional baselines is summarized in Table 7.

To assess whether the performance gap and improvement in counterfactual fairness of FCorrTransformer is caused by its simplified architecture, we conduct an ablation study with the baselines reported in Table 7. *FCorrTransformer + Res* and *FCorrTransformer + Res + CAR* reintroduce the residual connection in the attention mechanism (dashed connection in Figure 1a), with the latter additionally applying CAR. *FCorrTransformer + Rem* removes the sensitive features during training and restores them during evaluation. Finally, *FCorrTransformer + PreR* and *FT-Transformer + PreR* replace CAR with prediction-level regularization applied to the entire model. Please refer to Appendix F for the impact of embedding dimension. The results show that reintroducing the residual connection improves predictive performance but substantially weakens the effect of CAR and the counterfactual fairness guarantee, as observed for *FCorrTransformer + Res + CAR*. The residual connection allows information to bypass the attention redistribution mechanism, undermining the architectural constraint that enables CAR to regulate counterfactual predictions. Naive feature removal also proves inadequate. As shown by *FCorrTransformer + Rem*, it can substantially degrade predictive performance while failing to eliminate bias through indirect discrimination, as correlated non-sensitive features may still encode information about the sensitive attribute. Finally, prediction-level regularization is less effective than CAR. The FCorrTransformer-based model exhibits a more pronounced degradation in predictive performance, while the FT-Transformer-based model collapses to predicting the negative class for all samples.



(a) Model performance



(b) Fairness metrics

Figure 6: Impact of embedding dimension on BAF dataset

For interpretability comparison with the proposed framework, we visualize the top-10 feature importance scores obtained from LightGBM's built-in feature importance function in Figure 7. Among the 31 features, five features (in the top-10 list) overlap with those identified by FCorrTransformer in Figures 4a and 4b. This suggests that although FCorrTransformer learns a slightly different internal representation from LightGBM, the two models are largely consistent in identifying the most informative features. One thing to note about the performance of TabTransformer in Table 2 is that, due to its architectural design, the model size cannot be reduced to be comparable with the other models. Therefore, we adopt a minimal TabTransformer configuration for this dataset. However, despite its relatively large number of parameters, TabTransformer does not achieve better performance in our experiments.

The CAR regularization centralizes attention on a few critical entries, such as the self-attention of *current_address_months_count* and *has_other_cards*, as well as the dependencies between these two features, suggesting that they provide relatively fair predictive signals. In contrast, the atten-

Table 5: Impact of embedding dimension on model performance comparison on BAF dataset

Subset	Dimension	Model	Accuracy	F1 score	FPR	FNR	AUROC	AUPRC
Train	1	FCorrTransformer	0.9808 (0.0044)	0.2219 (0.0097)	0.0109 (0.0051)	0.7535 (0.0619)	0.8906 (0.0013)	0.1568 (0.0037)
		FCorrTransformer + CAR	0.9811 (0.0058)	0.1926 (0.0618)	0.0103 (0.0069)	0.7823 (0.0966)	0.8863 (0.0006)	0.1463 (0.0036)
	2	FCorrTransformer	0.9789 (0.0039)	0.2251 (0.0088)	0.0131 (0.0046)	0.7243 (0.0561)	0.8895 (0.0012)	0.1543 (0.0046)
		FCorrTransformer + CAR	0.9766 (0.0109)	0.1901 (0.0410)	0.0153 (0.0125)	0.7415 (0.1321)	0.8856 (0.0007)	0.1413 (0.0077)
	4	FCorrTransformer	0.9803 (0.0034)	0.2187 (0.0135)	0.0115 (0.0040)	0.7501 (0.0528)	0.8882 (0.0005)	0.1482 (0.0050)
		FCorrTransformer + CAR	0.9786 (0.0116)	0.1720 (0.0470)	0.0128 (0.0133)	0.7823 (0.1424)	0.8841 (0.0011)	0.1383 (0.0063)
	8	FCorrTransformer	0.9778 (0.0029)	0.2198 (0.0061)	0.0143 (0.0034)	0.7199 (0.0388)	0.8864 (0.0008)	0.1442 (0.0056)
		FCorrTransformer + CAR	0.9828 (0.0062)	0.1867 (0.0229)	0.0081 (0.0072)	0.8175 (0.0843)	0.8828 (0.0013)	0.1372 (0.0050)
	16	FCorrTransformer	0.9824 (0.0021)	0.2116 (0.0094)	0.0089 (0.0025)	0.7874 (0.0353)	0.8858 (0.0013)	0.1418 (0.0047)
		FCorrTransformer + CAR	0.9758 (0.0116)	0.1962 (0.0154)	0.0161 (0.0131)	0.7367 (0.1171)	0.8828 (0.0012)	0.1356 (0.0066)
	32	FCorrTransformer	0.9790 (0.0048)	0.2095 (0.0120)	0.0128 (0.0057)	0.7466 (0.0725)	0.8852 (0.0014)	0.1403 (0.0063)
		FCorrTransformer + CAR	0.9800 (0.0094)	0.1609 (0.0490)	0.0111 (0.0109)	0.8057 (0.1310)	0.8795 (0.0012)	0.1282 (0.0050)
Test	1	FCorrTransformer	0.9810 (0.0043)	0.2103 (0.0097)	0.0111 (0.0050)	0.7580 (0.0627)	0.8887 (0.0014)	0.1420 (0.0030)
		FCorrTransformer + CAR	0.9813 (0.0059)	0.1872 (0.0607)	0.0105 (0.0069)	0.7806 (0.0977)	0.8851 (0.0013)	0.1365 (0.0028)
	2	FCorrTransformer	0.9791 (0.0039)	0.2145 (0.0086)	0.0133 (0.0045)	0.7285 (0.0563)	0.8880 (0.0014)	0.1381 (0.0024)
		FCorrTransformer + CAR	0.9769 (0.0110)	0.1830 (0.0406)	0.0154 (0.0124)	0.7444 (0.1274)	0.8850 (0.0014)	0.1300 (0.0075)
	4	FCorrTransformer	0.9805 (0.0034)	0.2100 (0.0119)	0.0117 (0.0040)	0.7524 (0.0522)	0.8862 (0.0017)	0.1355 (0.0063)
		FCorrTransformer + CAR	0.9791 (0.0116)	0.1685 (0.0438)	0.0128 (0.0131)	0.7828 (0.1381)	0.8826 (0.0016)	0.1298 (0.0066)
	8	FCorrTransformer	0.9780 (0.0030)	0.2106 (0.0045)	0.0146 (0.0035)	0.7208 (0.0406)	0.8843 (0.0012)	0.1318 (0.0052)
		FCorrTransformer + CAR	0.9832 (0.0063)	0.1811 (0.0230)	0.0082 (0.0072)	0.8185 (0.0835)	0.8817 (0.0018)	0.1290 (0.0037)
	16	FCorrTransformer	0.9827 (0.0022)	0.2042 (0.0103)	0.0091 (0.0026)	0.7884 (0.0364)	0.8831 (0.0017)	0.1308 (0.0052)
		FCorrTransformer + CAR	0.9761 (0.0116)	0.1893 (0.0151)	0.0162 (0.0129)	0.7386 (0.1130)	0.8811 (0.0022)	0.1246 (0.0076)
	32	FCorrTransformer	0.9791 (0.0049)	0.2014 (0.0135)	0.0131 (0.0058)	0.7466 (0.0737)	0.8821 (0.0018)	0.1286 (0.0066)
		FCorrTransformer + CAR	0.9803 (0.0095)	0.1495 (0.0505)	0.0113 (0.0110)	0.8109 (0.1337)	0.8771 (0.0021)	0.1186 (0.0047)

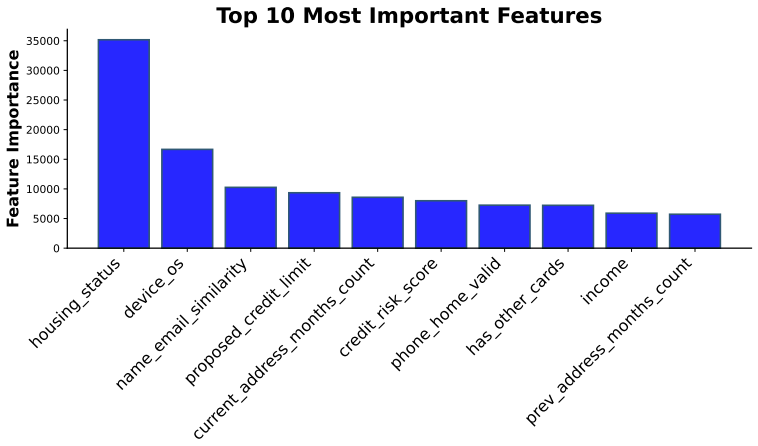


Figure 7: Top 10 feature importance in LightGBM on BAF data

Table 6: Impact of embedding dimensionality on fairness metrics comparison on BAF dataset

Subset	Dimension	Model	DPD	EqOdd	F1 Gap	AUROC Gap	AUPRC Gap
Train	1	FCorrTransformer	0.0968 (0.0336)	0.4914 (0.0837)	0.3715 (0.1400)	0.0157 (0.0065)	0.0670 (0.0175)
		FCorrTransformer + CAR	0.0541 (0.0358)	0.3585 (0.1823)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
	2	FCorrTransformer	0.1114 (0.0378)	0.5668 (0.1836)	0.3816 (0.0990)	0.0158 (0.0042)	0.0850 (0.0446)
		FCorrTransformer + CAR	0.0782 (0.0602)	0.4346 (0.2581)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
	4	FCorrTransformer	0.1005 (0.0285)	0.4537 (0.0816)	0.3182 (0.1326)	0.0139 (0.0045)	0.0724 (0.0490)
		FCorrTransformer + CAR	0.0697 (0.0726)	0.3437 (0.2252)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
	8	FCorrTransformer	0.1223 (0.0324)	0.5237 (0.0759)	0.3357 (0.1078)	0.0179 (0.0053)	0.0841 (0.0241)
		FCorrTransformer + CAR	0.0414 (0.0273)	0.2707 (0.0960)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
	16	FCorrTransformer	0.0813 (0.0228)	0.4385 (0.0671)	0.4356 (0.1715)	0.0160 (0.0077)	0.0641 (0.0336)
		FCorrTransformer + CAR	0.0657 (0.0465)	0.3494 (0.1663)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
	32	FCorrTransformer	0.0996 (0.0511)	0.5320 (2431)	0.3135 (0.1575)	0.0148 (0.0087)	0.0639 (0.0514)
		FCorrTransformer + CAR	0.0412 (0.0371)	0.2164 (0.1336)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
Test	1	FCorrTransformer	0.1241 (0.0425)	0.7429 (0.0857)	0.2614 (0.1038)	0.0134 (0.0079)	0.0742 (0.0297)
		FCorrTransformer + CAR	0.1064 (0.0529)	0.6590 (0.2358)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
	2	FCorrTransformer	0.1499 (0.0403)	0.7429 (0.0857)	0.2338 (0.0729)	0.0105 (0.0039)	0.0795 (0.0270)
		FCorrTransformer + CAR	0.1135 (0.0517)	0.7000 (0.2065)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
	4	FCorrTransformer	0.1342 (0.0442)	0.8000 (0.0700)	0.2583 (0.1163)	0.0106 (0.0051)	0.0609 (0.0395)
		FCorrTransformer + CAR	0.1118 (0.0785)	0.6571 (0.1938)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
	8	FCorrTransformer	0.1349 (0.0411)	0.8000 (0.0948)	0.2427 (0.0710)	0.0116 (0.0045)	0.0800 (0.0459)
		FCorrTransformer + CAR	0.1253 (0.0434)	0.6571 (0.1309)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
	16	FCorrTransformer	0.1172 (0.0466)	0.7857 (0.1152)	0.3287 (0.1465)	0.0141 (0.0062)	0.0789 (0.0344)
		FCorrTransformer + CAR	0.1191 (0.0600)	0.7000 (0.1000)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
	32	FCorrTransformer	0.1403 (0.0422)	0.7857 (0.0958)	0.2546 (0.1272)	0.0098 (0.0046)	0.0774 (0.0560)
		FCorrTransformer + CAR	0.1009 (0.0680)	0.5337 (0.2538)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)

tion on significant features such as *customer age*, *velocity_4w*, and *prev_address_months_count* is suppressed, as these features exhibit statistical dependencies with biased information. Overall, the attention mechanism in FCorrTransformer provides an interpretable perspective on how CAR reshapes feature dependencies during training and how such dependencies contribute to the model's decision-making.

For the attention matrix in FT-Transformer, as illustrated in Figure 8a, it exhibits a much denser attention distribution compared to FCorrTransformer. In contrast to FCorrTransformer, the statistical interpretation of the attention matrix in FT-Transformer is less clear. However, given the noticeable variation in attention weights across columns, we aggregate the attention weights by column (instead of using diagonal elements) and treat them as feature significance scores for FT-Transformer, as shown in Figure 8b. The resulting feature significance matches five of the features identified by LightGBM and shares most of the remaining features with FCorrTransformer. This suggests that FT-Transformer can still identify significant features, but its attention weights are less directly interpretable and provide limited insight into feature dependencies.

Table 7: More model baseline performance comparison

Subset	Model	Accuracy	F1 score	FPR	FNR	AUROC	AUPRC
Train	FCorrTransformer	0.9808 (0.0044)	0.2219 (0.0097)	0.0109 (0.0051)	0.7535 (0.0619)	0.8906 (0.0013)	0.1568 (0.0037)
	FCorrTransformer + Res	0.9746 (0.0030)	0.2299 (0.0025)	0.0182 (0.0035)	0.6607 (0.0390)	0.8934 (0.0012)	0.1591 (0.0030)
	FCorrTransformer + Res + CAR	0.9783 (0.0072)	0.2171 (0.0113)	0.0137 (0.0081)	0.7308 (0.0804)	0.8885 (0.0006)	0.1496 (0.0059)
	FCorrTransformer + Rem	0.9809 (0.0056)	0.2049 (0.0328)	0.0106 (0.0064)	0.7749 (0.0737)	0.8841 (0.0071)	0.1433 (0.0084)
	FCorrTransformer + PreR	0.9809 (0.0047)	0.1879 (0.0669)	0.0105 (0.0058)	0.7818 (0.0974)	0.8485 (0.1097)	0.1293 (0.0403)
	FT-Transformer + PreR	0.9888 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)	1.0000 (0.0000)	0.6357 (0.1079)	0.0271 (0.0210)
Test	FCorrTransformer	0.9810 (0.0043)	0.2103 (0.0097)	0.0111 (0.0050)	0.7580 (0.0627)	0.8887 (0.0014)	0.1420 (0.0030)
	FCorrTransformer + Res	0.9750 (0.0030)	0.2203 (0.0030)	0.0182 (0.0034)	0.6656 (0.0386)	0.8917 (0.0008)	0.1453 (0.0029)
	FCorrTransformer + Res + CAR	0.9785 (0.0072)	0.2095 (0.0103)	0.0139 (0.0081)	0.7322 (0.0763)	0.8875 (0.0011)	0.1383 (0.0054)
	FCorrTransformer + Rem	0.9812 (0.0057)	0.2000 (0.0316)	0.0108 (0.0065)	0.7732 (0.0731)	0.8825 (0.0069)	0.1333 (0.0074)
	FCorrTransformer + PreR	0.9812 (0.0048)	0.1828 (0.0655)	0.0107 (0.0059)	0.7801 (0.0991)	0.8479 (0.1086)	0.1190 (0.0363)
	FT-Transformer + PreR	0.9895 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)	1.0000 (0.0000)	0.6370 (0.1064)	0.0260 (0.0201)

Table 8: More model baseline fairness comparison

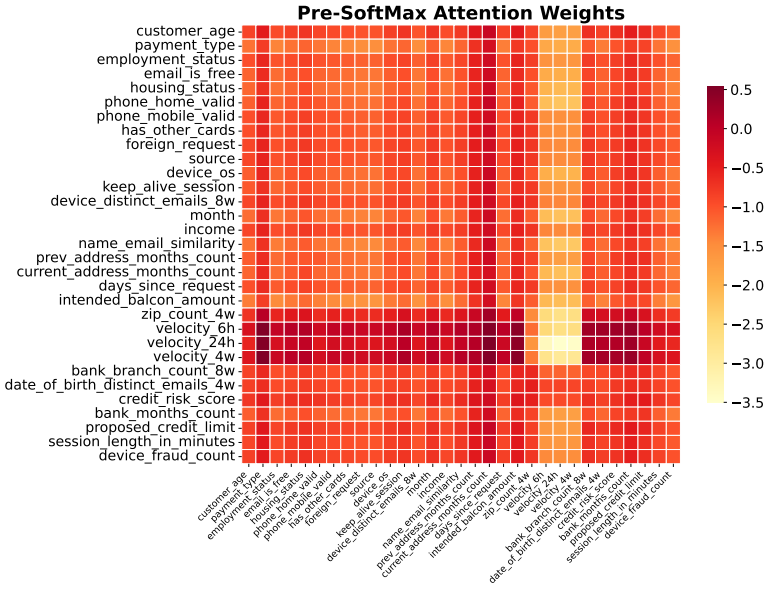
Subset	Model	DPD	EqOdd	F1 Gap	AUROC Gap	AUPRC Gap
Train	FCorrTransformer + Res	0.1155 (0.0374)	0.5279 (0.1499)	0.3012 (0.0606)	0.0090 (0.0037)	0.0588 (0.0426)
	FCorrTransformer + Res + CAR	0.0986 (0.0443)	0.4827 (0.1413)	0.2902 (0.1381)	0.0066 (0.0036)	0.0601 (0.0436)
	FCorrTransformer + Rem	0.0518 (0.0313)	0.2840 (0.0885)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
	FCorrTransformer + PreR	0.0492 (0.0240)	0.2809 (0.1444)	0.0000 (0.0000)	0.0019 (0.0057)	0.0011 (0.0034)
	FT-Transformer + PreR	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)	0.0399 (0.0439)	0.0036 (0.0050)
Test	FCorrTransformer + Res	0.1451 (0.0310)	0.7857 (0.0958)	0.1975 (0.0490)	0.0080 (0.0029)	0.0607 (0.0369)
	FCorrTransformer + Res + CAR	0.1155 (0.0317)	0.8000 (0.0700)	0.1726 (0.0582)	0.0043 (0.0021)	0.0523 (0.0291)
	FCorrTransformer + Rem	0.1081 (0.0448)	0.7229 (0.1353)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
	FCorrTransformer + PreR	0.1138 (0.0696)	0.6714 (0.2395)	0.0000 (0.0000)	0.0030 (0.0090)	0.0008 (0.0024)
	FT-Transformer + PreR	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)	0.0627 (0.0734)	0.0111 (0.0188)

Table 9: Complexity of each model on BAF dataset

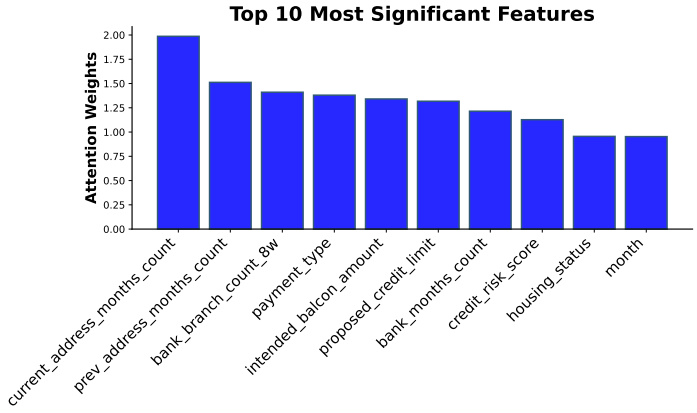
Model	Number of parameters	GPU Memory Usage/MB (batch of 64)	
		Parameter size	Forward/backward pass size
FFN	74,897	0.30	0.40
TabTransformer	126,998	0.51	5.63
FT-Transformer	94,981	0.37	67.88
FCorrTransformer	11,064	0.04	0.25

G.2 IMBALANCED REGRESSION: INSURTECH

We further evaluate the proposed framework on an imbalanced regression task using the InsurTech dataset introduced by Quan et al. (2025). This dataset consists of real-world Business Owner's Policy (BOP) insurance data enriched with InsurTech features extracted from online sources. We focus on the liability (LIAB) coverage line, which includes 235,636 insurance policies and 581 features in the train/test split. The response feature is the claim amount incurred during the policy period. Please refer Quan et al. (2025) for more details on the dataset. Consistent with the characteristics of



(a) Feature attention weights



(b) Top 10 feature significance

Figure 8: Attention weights and top 10 features in FT-Transformer on BAF data

business insurance, the data exhibit extreme imbalance: only 1,870 policies (approximately 0.79%) report non-zero claims, and the positive claim amounts are highly skewed, with a mean of 27,507.90 and a standard deviation of 33,244.72. We select business credit score as the sensitive feature, which is a multi-class categorical feature with seven categories (A+, A, B+, B, C+, C, and missing). The models are optimized to minimize Mean Squared Error (MSE). The performance and fairness metrics are summarized in Table 10 and 11 respectively. Given the high dimensionality of the InsurTech dataset, training the FT-Transformer + CAR model requires over 20 GPU hours. Thus, we report only the results of the optimal model.

For the InsurTech experiment, we report the same performance metrics used in Quan et al. (2025), which capture prediction ranking, portfolio-level performance, and individual prediction accuracy, key criteria for practical decision-making in insurance applications. Note that we make two minor modifications to the dataset compared to the original setup. First, since neural networks cannot inherently handle missing values, while LightGBM can, we apply simple imputation using the mean for continuous features and the mode for categorical features. While this treatment is not optimal,

Table 10: Model performance comparison on InsurTech dataset

Subset	Model	Gini	PE	RMSE	MAE
Train	FT-Transformer	0.52	-0.03	3,927.13	453.69
	FT-Transformer + CAR	0.50	**0.01**	3,928.28	444.24
	FCorrTransformer	**0.57**	**0.01**	**3,915.64**	**441.94**
	FCorrTransformer + CAR	0.54	**-0.01**	3,918.32	445.86
Test	FT-Transformer	0.46	-0.32	3,311.66	407.21
	FT-Transformer + CAR	0.46	**-0.26**	3,311.97	398.05
	FCorrTransformer	**0.54**	**-0.26**	**3,306.11**	**396.06**
	FCorrTransformer + CAR	0.50	-0.28	3,310.45	398.99

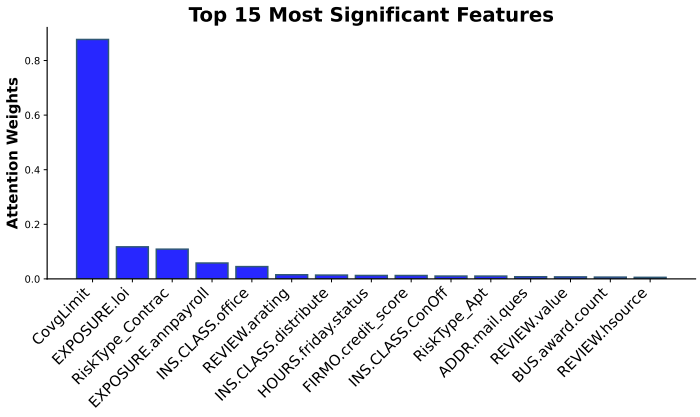
Table 11: Fairness metrics comparison on InsurTech dataset

Subset	Model	DPD	AvgIF	RMSE Gap	MAE Gap
Train	FT-Transformer	135.98	187.15	7.22	182.27
	FT-Transformer + CAR	52.19	55.91	1.21	55.25
	FCorrTransformer	75.82	90.74	4.11	88.46
	FCorrTransformer + CAR	**41.48**	**0.00**	**0.00**	**0.00**
Test	FT-Transformer	140.32	187.49	32.63	181.33
	FT-Transformer + CAR	**53.03**	55.97	9.38	55.39
	FCorrTransformer	113.62	92.76	19.23	90.51
	FCorrTransformer + CAR	59.47	**0.00**	**0.00**	**0.00**

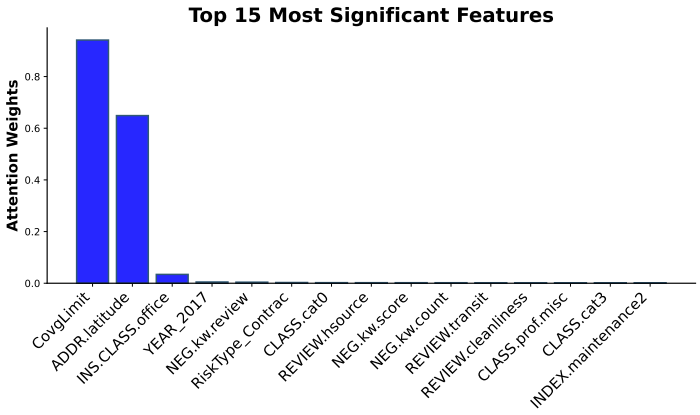
it is a simplified approach strictly for comparative purposes for neural network-based approaches. Second, as all multi-class categorical features are binarized in the original dataset (e.g., separate indicators for credit scores A+, A, etc.), we regroup the credit score indicators into a single nine-class categorical feature. Table 10 summarizes the performance of all evaluated models. In our experiments, straightforward FFN collapses to a constant prediction (202.4047) despite extensive optimization. While this result achieves absolute fairness, it lacks practical utility for real-world insurance applications, indicating that simple linear architectures struggle to capture signal within complex, imbalanced insurance datasets. Furthermore, fairness evaluations in Table 11 reveal that CAR significantly enhances both group and counterfactual fairness for both models. Notably, only the FCorrTransformer+CAR provides the robust counterfactual fairness guarantees to meet stringent regulatory demands.

Due to the high dimensionality (581 features) of the InsurTech dataset, visualizing the full attention matrices is not informative for human interpretation. Instead, we extract the diagonal elements of the attention matrix as feature significance scores from FCorrTransformer, as shown in Figure 9. Consistent with LightGBM, *CovgLimit* (i.e., coverage limit of the purchased insurance policy) is identified as the most influential feature, along with several features previously reported in Quan et al. (2025). However, the overall feature significance profiles differ. In the original study, territorial risk featurees (i.e., *TERRITORY.xx*) provided by insurers are considered highly significant, whereas in FCorrTransformer (Figure 9a), firm-provided risk characterizations (i.e., *INS.CLASS.xxx*) emerge as significant features. These features are strongly correlated with business credit score and are therefore suppressed by CAR (Figure 9b). As a result, attention is centered toward InsurTech-engineered features, such as negative keyword information (i.e., *NEG.kw.xxx*), InsurTech classification factors (*CLASS.xxx*), and engineered indices (*INDEX.xxx*). This suggests that InsurTech features, which are more personalized and fine-grained, provide relatively fairer predictive signals for insurance pricing compared to traditional firm-defined risk characteristics.

The complexity analysis, summarized in Table 12, shows that FCorrTransformer contains roughly half the number of parameters of FT-Transformer, reflecting its attention-light design philosophy. On the InsurTech dataset, the FFN accounts for 10,145 out of 21,767 parameters (46.6%) in FCorrTransformer, whereas it contributes less than 0.1% of the parameters in FT-Transformer. This architectural design, combined with the high dimensionality of the InsurTech dataset, allows FCorrTransformer to use only 4.4% of the GPU memory during training (on the same batch size), achiev-



(a) FCorrTransformer



(b) FCorrTransformer + CAR

Figure 9: Top 15 feature significance on InsurTech dataset

Table 12: Complexity of each model on InsurTech dataset

Model	Number of parameters	GPU Memory Usage/MB (batch of 64)	
		Parameter size	Forward/backward pass size
FFN	383,313	1.53	0.43
FT-Transformer	55,317	0.22	661.12
FCorrTransformer	21,767	0.09	2.91

ing more efficient training/deployment, comparable predictive performance while providing a strong counterfactual fairness guarantee.

G.3 Classification: Adult

We further evaluate the proposed framework on the well-studied Adult (Census Income) dataset from the UCI Machine Learning Repository[6]. The model's predictive performance and fairness metrics are summarized in Tables 13 and 14, respectively. The results indicate that FCorrTransformer achieves predictive performance comparable to FT-Transformer, while simultaneously improving counterfactual fairness through the CAR regularization. These findings further strengthen the feasibility and practical applicability of the proposed framework.

[6] https://archive.ics.uci.edu/dataset/20/census+income

Table 13: Model performance comparison on Adult dataset

Subset	Model	Accuracy	F1 score	FPR	FNR	AUROC	AUPRC
Train	FT-Transformer	**0.8599** (0.0018)	**0.6849** (0.0088)	0.0682 (0.0073)	**0.3670** (0.0231)	**0.9177** (0.0018)	0.7775 (0.0045)
	FT-Transformer + CAR	0.8597 (0.0022)	0.6841 (0.0115)	0.0679 (0.0078)	0.3685 (0.0276)	0.9174 (0.0017)	0.7772 (0.0043)
	FCorrTransformer	0.8550 (0.0031)	0.6637 (0.0198)	0.0630 (0.0162)	0.4033 (0.0492)	0.9122 (0.0031)	**0.7833** (0.0110)
	FCorrTransformer + CAR	0.8440 (0.0127)	0.5984 (0.0889)	**0.0475** (0.0245)	0.4978 (0.1177)	0.9067 (0.0042)	0.7671 (0.0164)
Test	FT-Transformer	0.8462 (0.0016)	0.6470 (0.0091)	0.0768 (0.0077)	**0.4027** (0.0234)	0.8987 (0.0017)	0.7339 (0.0041)
	FT-Transformer + CAR	0.8461 (0.0017)	0.6461 (0.0126)	0.0764 (0.0078)	0.4045 (0.0278)	0.8988 (0.0013)	0.7337 (0.0048)
	FCorrTransformer	**0.8538** (0.0038)	**0.6545** (0.0214)	0.0642 (0.0157)	0.4112 (0.0495)	**0.9072** (0.0031)	**0.7703** (0.0116)
	FCorrTransformer + CAR	0.8447 (0.0127)	0.5910 (0.0921)	**0.0472** (0.0244)	0.5048 (0.1191)	0.9032 (0.0041)	0.7584 (0.0144)

Table 14: Fairness metrics comparison on Adult dataset

Subset	Model	DPD	EqOdd	F1 Gap	AUROC Gap	AUPRC Gap
Train	FT-Transformer	0.1882 (0.0141)	0.3177 (0.0716)	0.3437 (0.0899)	0.0666 (0.0242)	0.2241 (0.0547)
	FT-Transformer + CAR	0.1855 (0.0059)	0.3494 (0.0602)	0.3263 (0.0808)	0.0394 (0.0112)	0.1507 (0.0375)
	FCorrTransformer	0.1690 (0.0230)	0.2065 (0.0582)	0.1163 (0.0599)	0.0072 (0.0052)	0.0223 (0.0198)
	FCorrTransformer + CAR	**0.1437** (0.0378)	**0.1647** (0.0547)	**0.0012** (0.0028)	**0.0000** (0.0001)	**0.0001** (0.0001)
Test	FT-Transformer	0.2030 (0.0212)	0.4451 (0.0632)	0.4322 (0.0419)	0.0556 (0.0218)	0.1876 (0.0349)
	FT-Transformer + CAR	0.1984 (0.0174)	0.4000 (0.0664)	0.3810 (0.0718)	0.0298 (0.0133)	0.1171 (0.0329)
	FCorrTransformer	0.1711 (0.0260)	0.2767 (0.0871)	0.1651 (0.0843)	0.0077 (0.0049)	0.0299 (0.0204)
	FCorrTransformer + CAR	**0.1412** (0.0333)	**0.1822** (0.0685)	**0.0026** (0.0077)	**0.0000** (0.0000)	**0.0001** (0.0001)