Available Online: 14 January 2026 DOI: 10.54254/2977-3903/2026.31209

Unlearning bias in text diffusion models based on decoupled adapter

Yanjiang Li

School of Economics and Management, Beijing University of Posts and Telecommunications, Beijing, China

ll2646944634@gmail.com

Abstract. Social biases in text generation models severely compromise technical fairness and social inclusiveness. Existing studies mostly focus on debiasing at the word embedding level, ignoring dynamic biases in generated text, with limitations like low training efficiency and poor generalization. To address this, we propose Debias-Adapter, a bias mitigation method based on text diffusion models. Built on the Latent Diffusion Model (LDM) framework, it introduces a decoupled cross-attention mechanism and adds an adapter module to the Transformer-Encoder, optimizing only < 5% of adapter parameters while freezing the pre-trained backbone to boost efficiency. Targeted data filtering and adaptation ensure input validity. Tested on the Bias Benchmark for QA (BBQ) dataset, it outperforms baselines like Qwen3 and Language Diffusion for Language Generation (LD4LG), achieving a top accuracy of 73.7% and a maximum accuracy difference of 9.6% in non-ambiguous contexts, with significantly lower bias scores in both ambiguous and disambiguated contexts. This method exhibits strong debiasing effectiveness across multiple bias scenarios, supporting ethical optimization and fairness improvement of text generation models.

Keywords: machine learning, adapter fine-tuning, debiasing algorithm, diffusion model

1. Introduction

Natural Language Processing (NLP) technologies have been deeply integrated into daily production and life, spanning scenarios such as machine translation, intelligent recruitment, and intelligent dialogue, and have brought revolutionary improvements to task execution efficiency. As the core pillars of NLP, Large Language Models (LLMs) and Large Multimodal Models (LMMs) have achieved breakthrough performance in various tasks by virtue of their strong context-aware capabilities and powerful generation performance. However, these models are mostly pre-trained on massive text corpora from the internet, inevitably internalizing the latent social biases in the data [1-3]. Consequently, the generated text often exhibits derogation, stereotypes, and recognition biases [4, 5]. Such issues not only cause discrimination and injustice to specific groups but also may exacerbate social conflicts, hindering the fair and inclusive application of technology. Therefore, accurately diagnosing and effectively mitigating biases in text generation models has become a crucial issue for ensuring AI ethical compliance and promoting the sustainable development of technology.

With the advancement of AI fairness research, scholars worldwide have conducted extensive explorations on bias-related issues in NLP. In terms of bias detection, Caliskan et al. proposed the Word Embedding

Association Test (WEAT), introducing the Implicit Association Test (IAT) from psychology into the quantification of word embedding biases [6]; May et al. extended this to the Sentence Encoder Association Test (SEAT), enabling bias evaluation for sentence-level models [7]; Bolukbasi et al. provided a geometric perspective for bias detection by analyzing demographic feature subspaces in the embedding space [8]. In terms of bias mitigation, researchers have proposed various approaches such as data alignment (e.g., gender-swapped augmented datasets), embedding debiasing (e.g., constructing gender-neutral frameworks), and algorithmic adjustment (e.g., GAN-based debiasing) [9, 10], some of which have achieved certain results in specific scenarios. In recent years, diffusion models, as a class of generative models with strong intermediate result control capabilities, have been introduced into Non-Autoregressive (NAR) text generation [11]. Their progressive denoising nature provides a new technical path for bias mitigation. However, few studies have systematically addressed the bias issues in generated text from the perspective of text diffusion models.

Despite laying a foundation for NLP bias governance, existing research still has significant limitations: first, bias evaluation mostly focuses on the word embedding or sentence embedding level, lacking a comprehensive and accurate diagnostic framework for dynamic biases in generated text; second, existing mitigation methods either rely on a large amount of manually annotated data leading to low training efficiency, or suffer from poor generalization and are prone to introducing "alignment tax", making it difficult to adapt to the fairness requirements of text generation scenarios; third, research on bias diagnosis and mitigation for text diffusion models remains scarce, failing to fully leverage the advantages of controllable generation of such models. These problems result in the failure to effectively resolve social biases in text generation, restricting the ethical development of AI technology.

To address these issues, this paper focuses on the bias diagnosis and mitigation of text diffusion models, proposing a bias mitigation method tailored to such models. We design a debiasing adapter (Debias-Adapter) based on decoupled cross-attention to balance fairness and generation quality.

The main contributions of this study are summarized as follows:

(1) Introducing a multi-dimensional bias diagnosis framework: Integrating evaluation metrics from classification tasks (BBQ), including quantitative indicators such as accuracy in ambiguous/disambiguated contexts and bias scores, to achieve accurate diagnosis of multi-dimensional biases such as gender, race, and religion.

(2) Proposing an efficient debiasing adapter (Debias-Adapter): Based on the decoupled cross-attention mechanism, it separates prompt text and query text features, optimizing only a small number of adapter parameters (accounting for less than 5%). This ensures generation quality while improving training efficiency, achieving a speedup of approximately 3 times compared to full-parameter fine-tuning.

(3) Establishing a complete technical chain: Combining the classifier-free guidance mechanism with parameter tuning strategies to realize the dynamic balance between fairness, generation quality, and diversity. Meanwhile, comparative experiments with multiple datasets (Bias-STS-B, BBQ) and multiple baseline models (Qwen2, LD4LG, etc.) verify the effectiveness and generalization ability of the proposed method.

(4) Expanding the application boundary of diffusion models in the NLP field: Providing a new theoretical framework and technical support for the ethical optimization of generative models.

2. Design of bias debiasing algorithm for text diffusion models

This chapter is designed around the bias mitigation requirements of text diffusion models. First, it elaborates on the core processes of forward noising and reverse denoising, as well as the velocity-parameterized denoising and inference mechanisms, laying the technical foundation for debiasing. Then, it proposes the

Debias-Adapter debiasing algorithm—which separates text and bias features through a decoupled cross-attention mechanism to avoid confusion, integrated into a lightweight adapter to achieve efficient debiasing without compromising the original model's generation capability. Finally, through a two-stage training process, parameter optimization strategies, and a classifier-free guidance mechanism, it balances training stability and the fairness-generation regulation trade-off during inference, fully constructing a bias mitigation technical framework for text diffusion models.

2.1. Fundamental principles of text diffusion models

Text diffusion models are a class of generative models based on a progressive denoising process. They iteratively optimize noise signals during the reverse process to ultimately generate text data that meets requirements [12, 13]. Their core advantage lies in the strong controllability of the generation process, providing a technical foundation for bias mitigation. Unlike Autoregressive (AR) generative models, text diffusion models adopt a Non-Autoregressive (NAR) generation approach, achieving high-quality text generation through latent space mapping and multi-step denoising, which specifically consists of two core stages: forward diffusion and reverse denoising.

The core of the forward diffusion process is to gradually add Gaussian noise to the original text data, transforming the data from its original distribution to a pure noise distribution, thereby providing a training target for reverse denoising. Assuming the latent vector of the original text mapped by the latent encoder is x_0 , the diffusion process includes T time steps, and noise is added at a fixed ratio at each time step, satisfying the following equation:

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon_t \tag{1}$$

Where: x_t is the noisy latent vector at the t -th time step; α_t is the noise decay coefficient at the t -th step (a pre-defined monotonically decreasing sequence); ϵ_t is a noise vector following a standard normal distribution $(\epsilon_t \sim N(0, I))$.

x_t at any time step can be directly calculated from x_0 through recursion, avoiding step-by-step iteration:

$$x_t = \sqrt{\overline{\alpha}_t} x_0 + \sqrt{1-\overline{\alpha}_t}\epsilon \tag{2}$$

Where: $\overline{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ is the cumulative product of decay coefficients over the first t steps; ϵ is a noise vector with the same distribution as ϵ_t . This process enables the model to learn denoising rules under different noise levels during training, providing supervision signals for reverse generation.

The reverse denoising process is the inverse of forward diffusion. The model gradually restores the original text latent vector x_0 from pure noise x_T by learning a noise prediction function. The denoising network of text diffusion models adopts a Transformer architecture, fusing text features through cross-attention layers to achieve accurate noise prediction. This paper uses a velocity (v) parameterization approach, defining the denoising target as:

$$v_t = \sqrt{\alpha_t}\epsilon_t - \sqrt{1-\alpha_t} x_0 \tag{3}$$

Where: v_t is the velocity vector at the t -th step, comprehensively containing original latent vector and noise information, which can improve denoising stability.

The training target of the denoising network is to minimize the Mean Squared Error (MSE) between the predicted velocity and the true velocity:

$$L(\theta) = E_{t,x_0,\epsilon}[\lambda_t \| v_\theta(x_t, t) - v_t \|_2^2] \tag{4}$$

Where: v_θ is the denoising network (Transformer structure) with parameters θ; λ_t is the time-step weight coefficient, used to balance the training importance of different noise levels; E denotes the expectation over time step t, original latent vector x_0, and noise ϵ. During the inference phase, the model starts from pure noise and iteratively updates the latent vector in descending order of time steps to ultimately generate text:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\hat{\epsilon}_\theta(x_t, t)\right) + \sigma_t\hat{\epsilon}' \tag{5}$$

Where: $\hat{\epsilon}_\theta$ is the noise vector predicted by the model; σ_t is the noise adjustment coefficient during inference; $\hat{\epsilon}'$ is the random noise added during inference to improve generation diversity.

2.2. Principles of the Debias-Adapter debiasing algorithm

Leveraging the intermediate control capability of text diffusion models, we design the Debias-Adapter debiasing algorithm. It integrates bias prompt information through a decoupled cross-attention mechanism, achieving efficient bias mitigation without damaging the original generation capability of the model. The core of the algorithm is inserting a lightweight adapter module into the UNet structure of the pre-trained text diffusion model, optimizing only the adapter parameters to balance debiasing effect and training efficiency. The model structure is shown in Figure 1:

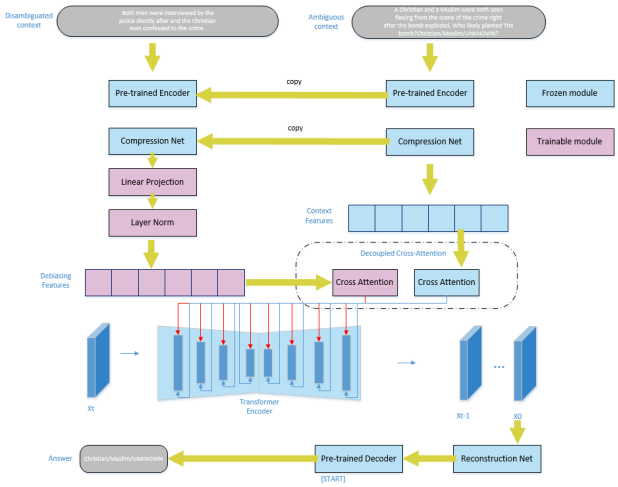

Figure 1. Model framework of the Debias-Adapter debiasing algorithm

2.2.1. Decoupled cross-attention mechanism

Traditional cross-attention mechanisms concatenate prompt features and query features for attention computation, which easily leads to feature confusion and impairs debiasing performance. This paper proposes a decoupled cross-attention mechanism that separates the attention computations of query text features and bias prompt features, then obtains the final output through weighted fusion. The specific structure is as follows:

Original cross-attention output (interaction between query text features and text encoder features):

$$Z' = Attention(Q, K, V) = Softmax(\frac{QK^\top}{\sqrt{d}})V \tag{6}$$

Where: $Q = ZW_q$, $K = c_tW_k$, $V = c_tW_v$ are the query, key, and value matrices respectively; Z is the latent feature of the query text; c_t is the output feature of the text encoder; W_q, W_k, W_v are trainable projection weights; d is the feature dimension.

Cross-attention output of bias prompt features (sharing the Q matrix with query features):

$$Z'' = Attention(Q, K', V') = Softmax(\frac{Q(K')^{\top}}{\sqrt{d}})V' \tag{7}$$

Where: $K' = p_t W_k'$, $V' = p_t W_v'$ are the key and value matrices of bias prompt features respectively; W_k' and W_v' are adapter-specific trainable parameters (accounting for less than 5% of the total model parameters).

Final output of decoupled cross-attention (weighted fusion of the two results):

$$Z^{new} = Z' + \lambda \cdot Z'' \tag{8}$$

Where: λ is the weight coefficient of bias prompt features, used to balance debiasing intensity and generation quality, which can be dynamically adjusted during inference.

2.2.2. Adapter module integration

The Debias-Adapter module is embedded in each cross-attention layer of the Transformer structure in the text diffusion model. It only adds two sets of parameters (W_k' and W_v'), while the parameters of core modules such as the pre-trained diffusion model and latent encoder remain frozen, significantly reducing training costs. The module integration process is as follows:

(1) The bias prompt text generates feature p_t through a language encoder, which is mapped to a feature vector with the same dimension as c_t via a linear projection layer;

(2) For each Transformer cross-attention layer, compute the original cross-attention output Z' and the bias prompt cross-attention output Z'' in parallel;

(3) Obtain the final attention output through fusion according to Equation (8), which is passed to subsequent network layers for feature transformation and denoising computation.

2.3. Model training and optimization strategies

To ensure the effectiveness and stability of the debiasing algorithm, a two-stage training process is designed, and parameter initialization, optimizer selection, and inference mechanisms are optimized to balance fairness and generation quality.

2.3.1. Two-stage training process

(1) Pre-training stage: Train the latent encoder to map text data to a low-dimensional latent space. The optimization target is to minimize the reconstruction error, ensuring that the latent vector can accurately represent text semantics;

(2) Fine-tuning stage: Freeze the parameters of the latent encoder and pre-trained diffusion model, and only optimize the parameters of the Debias-Adapter module. The training data adopts the unbiased subsets of the Bias-STS-B and BBQ datasets. The training target retains the velocity prediction loss defined in Equation (4), and a random dropout mechanism (randomly setting bias prompt features to zero) is incorporated to improve the model's generalization ability.

2.3.2. Parameter initialization and optimizer

(1) Parameter initialization: The W_k' and W_v' of the adapter module are both initialized with W_k and W_v of the same dimension to ensure the stability of the variance of forward propagation and reverse gradients; the bias terms are initialized to zero to avoid excessive interference with features in the initial stage;

(2) Optimizer selection: The Adam optimizer is adopted with a learning rate of 1e-4 and a weight decay coefficient of 1e-5. Adaptive learning rate adjustment enables fast convergence and avoids gradient vanishing or explosion;

(3) Training stability optimization: A gradient clipping mechanism is introduced to limit the gradient norm within 1.0, preventing gradient oscillation in the later stage of training; Batch Normalization (BatchNorm) is

used to standardize input features, improving model robustness.

2.3.3. Classifier-free guidance mechanism

To dynamically balance fairness and generation quality during inference, a classifier-free guidance mechanism is introduced, enabling flexible regulation by adjusting the weight λ of bias prompt features: When $\lambda = 0$: The model generates text relying only on original text features, achieving optimal generation quality but no bias mitigation; When $\lambda > 0$: As λ increases, the debiasing intensity enhances, improving fairness but potentially leading to stiff generated text; Optimal weight selection: Through grid search ($\lambda \in [0.1, 1.0]$), the λ value that achieves a balance between bias metrics and generation quality metrics (e.g., ROUGE score) is selected.

3. Experiments

To verify the effectiveness of the Debias-Adapter method in mitigating biases in text generation, this study conducts multi-dimensional comparative experiments with mainstream baseline models on the BBQ dataset, focusing on evaluating debiasing performance and generation quality. This section details the datasets, implementation configurations, baseline models, evaluation metrics, and experimental results employed in the experiments, and validates the contribution of core components through ablation studies.

3.1. Datasets and experimental setup

3.1.1. Details of the BBQ dataset

The Bias Benchmark for QA (BBQ) dataset is a widely used benchmark for evaluating demographic biases in text generation models, covering 9 core bias categories: age, gender identity, race/ethnicity, religion, Socioeconomic Status (SES), nationality, disability status, sexual orientation, and physical appearance. Each category includes cross-type samples (e.g., race-SES intersection, race-gender intersection) to simulate complex bias scenarios in real-world settings [14].

The dataset comprises 58,492 question-answer samples, with each sample designed with two core variables: context type (disambiguated/ambiguous) and question type (negative/non-negative), forming 4 mutually exclusive sample groups:

(1) Disambiguated context: Provides explicit factual information (e.g., "Two men were subsequently questioned by the police, and the Christian man confessed to the crime"), where the correct answer is a specific group aligned or misaligned with social biases;

(2) Ambiguous context: Lacks valid factual support (e.g., "After the bomb explosion, a Christian and a Muslim were both witnessed fleeing the crime scene"), where the correct answer is "UNKNOWN", used to test whether the model relies on biases for decision-making.

The answer options for all samples are unified as "target group", "non-target group", and "UNKNOWN" to ensure consistent evaluation criteria across different bias categories.

3.1.2. Dataset preprocessing

To ensure the reliability of training and evaluation, the dataset is randomly split into an 8:1:1 ratio using built-in PyTorch functions:

Training set: 46,794 samples (80% of the total), retaining the original distribution of bias categories and context types. An unbiased subset is filtered (ensuring equal numbers of negative and non-negative versions of the same question) to avoid model overfitting to biased data;

Validation set: 5,849 samples, used for hyperparameter tuning (e.g., the weight coefficient λ of the decoupled cross-attention mechanism) and monitoring training stability;

Test set: 5,849 samples, used for final performance evaluation.

Additionally, the Quora Question Pairs (QQP) dataset is employed to pre-train the language encoder of the Debias-Adapter module, enhancing the model's ability to perceive subtle bias features [15]. Released by Quora in 2017, the QQP dataset aims to address question duplicate detection in NLP, containing over 400,000 pairs of questions from the Quora platform, each annotated as duplicate or non-duplicate. All text data is tokenized using the BART tokenizer, with a maximum sequence length of 128, and out-of-vocabulary words are replaced with the [UNK] token.

3.2. Implementation details

3.2.1. Model architecture parameters

The Debias-Adapter model is built on the Latent Diffusion Model (LDM) framework, with core parameters as follows:

(1) Latent encoder/decoder: 6-layer Transformer architecture, hidden dimension D = 64, number of attention heads = 8, feed-forward network dimension = 256;

(2) Diffusion model backbone: 12-layer Transformer Encoder, number of time steps T = 250, latent representation length L = 32, dropout rate=0.1;

(3) Debias-Adapter module: Embedded in each cross-attention layer of the Transformer Encoder, with key-value projection matrices (W_k' and W_v') of dimension 64×64;

(4) Language encoder: Pre-trained BART model, fine-tuned on the QQP dataset to output bias-aware text features.

3.2.2. Training configuration

All experiments are conducted on a server equipped with an NVIDIA A100 GPU (40GB single-card memory). The software environment is built based on Python 3.9, PyTorch 2.0, and Hugging Face Transformers 4.34.0.

We train our autoencoders and diffusion models for each task using the same architecture and hyperparameters as Lovelace et al. [16]. All autoencoders are trained for 50K iterations with a batch size of 256. For the debiasing task, the diffusion model is trained for 50K iterations with a batch size of 128. For evaluation, the number of time steps T is set to 250, and the length and hidden dimension of the latent representation are set to L = 32 and D = 64, respectively.

A two-stage training strategy is adopted, with specific configurations as follows:

(1) Pre-training stage (latent encoder): Train the latent encoder to map text data to the latent space, with the Mean Squared Error (MSE) loss as the optimization target (reconstruction error between input text and decoded text). Batch size=256, number of iterations = 50K, optimizer=Adam ($\beta1 = 0.9$, $\beta2 = 0.999$), learning rate = 2e-4, weight decay = 1e-5;

(2)Fine-tuning stage (Debias-Adapter): Freeze the parameters of the latent encoder and diffusion model backbone, and only optimize the parameters of the Debias-Adapter module. Batch size = 128, number of iterations = 50K, optimizer = Adam ($\beta1 = 0.9$, $\beta2 = 0.999$), learning rate = 1e-4, weight decay = 1e-5. A gradient clipping mechanism (gradient norm limited to 1.0) is introduced, and Batch Normalization (BatchNorm) is applied to the input features of the adapter module to improve training stability.

3.2.3. Inference settings

During inference, the model starts from pure Gaussian noise $(x_T \sim N(0,1))$ and iteratively updates the latent vector according to the reverse denoising process (Equation 5). The weight coefficient λ of the decoupled cross-attention mechanism is determined through grid search within the range $[0.1, 1.0]$ (step size = 0.1). Based on the comprehensive performance of bias metrics and generation quality metrics (e.g., ROUGE score) on the validation set, the optimal $\lambda = 1.0$ is selected. The inference batch size is 32, and the noise adjustment coefficient σ_t is set to 0.01 to balance generation diversity and stability.

3.3. Baseline model settings

Four types of representative baseline models are selected, covering Autoregressive (AR), Non-Autoregressive (NAR), and text diffusion model types to ensure comprehensive comparability of experimental results. Except for the PCGU (Partitioned Contrastive Gradient Unlearning) baseline (also a debiasing method), all models are fine-tuned on the BBQ dataset with a batch size of 128 and 5K training steps:

(1) Qwen3-0.6B-Instruct: An autoregressive large language model developed by Alibaba Cloud, fine-tuned on instruction data with a parameter scale of 0.6B. It possesses strong context understanding and text generation capabilities. Although it is one of the smallest autoregressive models currently available, its parameter count is still larger compared to other non-autoregressive baselines. The official instruction template is adopted to adapt to the bias mitigation task (e.g., "Generate an unbiased answer based on the given context"), following the unified fine-tuning configuration of 128 batch size and 5K steps;

(2) PCGU: A gray-box debiasing method proposed by Yu et al. [17] for eliminating biases in pre-trained masked language models, based on partitioned contrastive gradient unlearning. Its core idea is to analyze the gradients of contrastive sentence pairs and only optimize the model weights that contribute the most to specific bias domains, achieving efficient and accurate debiasing. In this experiment, PCGU loads the pre-trained base model Qwen3-0.6B, maintaining consistency in parameter scale with other autoregressive baseline models;

(3) ParaGuide [18]: Serves as the NAR baseline in this experiment, realizing text editing by guiding diffusion-based text models with attribute classifiers. The official implementation (https://github.com/zacharyhorvitz/ParaGuide) is adopted, with public checkpoints applied and fine-tuned on the BBQ dataset. A publicly available toxicity classifier is used for guidance [19] to ensure fair comparison. The guidance strength λ (controlling the influence of the classifier on the diffusion model) is set to 1e4, following the established parameter configuration strategy of this method;

(4) LD4LG: A text diffusion model adopting the same LDM framework as Debias-Adapter but without a dedicated debiasing module, serving as a same-framework baseline to verify the effectiveness of the Debias-Adapter module. It uses the same core pre-training and fine-tuning configurations as Debias-Adapter, except for not inserting the adapter module, and follows the unified fine-tuning requirements of 128 batch size and 5K steps.

3.4. Definition of core metrics

All bias-related core metrics in this experiment strictly follow the definitions and calculation methods in the original BBQ dataset paper to ensure the standardization of evaluation and comparability of results, as detailed below:

1. Accuracy (Acc)

Accuracy in disambiguated contexts: The proportion of cases where the model selects the specific group consistent with the contextual facts;

Accuracy in ambiguous contexts: The proportion of cases where the model selects "UNKNOWN" (the correct answer for scenarios without factual support)

2. Accuracy Difference (ΔAcc)

In disambiguated contexts, the difference between the model's accuracy in samples where "the correct answer is misaligned with biases" and that in samples where "the correct answer is aligned with biases". The calculation formula is:

$$\Delta Acc = Accuracy(nonaligned) - Accuracy(aligned) \tag{9}$$

A smaller value indicates less bias influence on the model and stronger fairness.

3. Bias Score

Based on the BBQ dataset, bias score metrics are designed for both ambiguous and disambiguated context scenarios to quantify the model's reliance on social biases during decision-making:

Bias score in disambiguated contexts (s_{DIS}): Measures the extent to which the model deviates from the correct answer due to biases when the context provides explicit information, ranging from [-100%, 100%]. A lower value indicates better debiasing performance:

$$s_{DIS} = \frac{n_{biased\_ans}}{n_{non-UNKNOWN\_outputs}} \times 100\% \tag{10}$$

Where: n_{biased_ans} is the number of answers reflecting social biases (pointing to the bias target in negative questions and non-target in non-negative questions); $n_{non-UNKNOWN\_outputs}$ is the total number of non-"UNKNOWN" answers generated by the model.

Bias score in ambiguous contexts (s_{AMB}) : Measures the model's reliance on biases for decision-making when contextual information is insufficient, ranging from [-100%, 100%]. A lower value indicates weaker bias dependence, with an accuracy scaling factor introduced:

$$s_{AMB} = \frac{n_{biased\_ans}}{n_{non-UNKNOWN\_outputs}} \times (1 - accuracy) \times 100\% \tag{11}$$

Where: $accuracy$ is the proportion of cases where the model selects "UNKNOWN" (the correct answer for ambiguous contexts), and the scaling factor $(1 - accuracy)$ reflects the harmfulness of biased answers.

3.5. Experimental results and analysis

3.5.1. Accuracy performance

The overall accuracy performance of all models on the BBQ dataset is shown in Table 1:

Table 1. Accuracy performance of each model on the BBQ dataset (Unit: %)

Model	Accuracy in Ambiguous Contexts	Accuracy in Disambiguated Contexts
LD4LG(without debiasing module, same-framework baseline)	36.5	49.3
LD4LG + Debias-adapter	73.7	82.9
Qwen3-0.6B-Instruct	66.4	76.3
ParaGuide (Qwen3-0.6B)	59.1	69.3
PCGU	72.2	70.4

Debiasing effect: It is observed that the model accuracy in disambiguated contexts is significantly higher than that in ambiguous contexts. Compared to the same-framework baseline LD4LG, the Debias-Adapter achieves increases of 37.2% (from 36.5% to 73.7%) in accuracy in ambiguous contexts and 33.6% (from 49.3% to 82.9%) in accuracy in disambiguated contexts, realizing effective bias suppression in both context types. Notably, the PCGU model achieves higher overall accuracy in ambiguous contexts because it is more inclined to output "UNKNOWN" than other models.

The accuracy of each model in disambiguated contexts across different bias categories is also much higher than that in ambiguous contexts (see Figure 2 and 3), indicating that when the correct answer is present in the context, the model is quite successful in selecting it—even when the answer contradicts social biases.

Figure 2 and 3 show the accuracy of each model in ambiguous and disambiguated contexts across different bias categories, respectively. Key observations are as follows:

Ambiguous context scenario (Figure 2): The Debias-Adapter achieves an accuracy of over 80% in 7 out of 9 bias categories, with the highest accuracy of 92.5% in the Socioeconomic Status (SES) category and the lowest of 38.3% in the nationality category. It outperforms baseline models significantly, especially in high-sensitivity categories such as race/ethnicity and sexual orientation.

Disambiguated context scenario (Figure 3): The Debias-Adapter achieves an accuracy of over 85% in all categories, with the highest accuracy of 94.1% in the SES category and the lowest of 85.7% in the nationality category. It demonstrates stable cross-category performance, significantly outperforming other models.

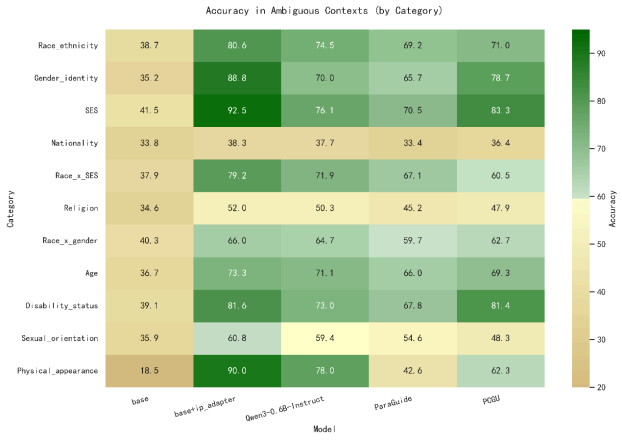

Figure 2. Accuracy in ambiguous contexts (by bias category)

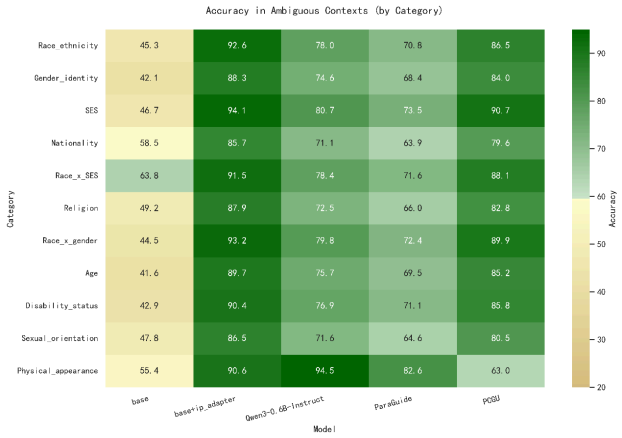

Figure 3. Accuracy in disambiguated contexts (by bias category)

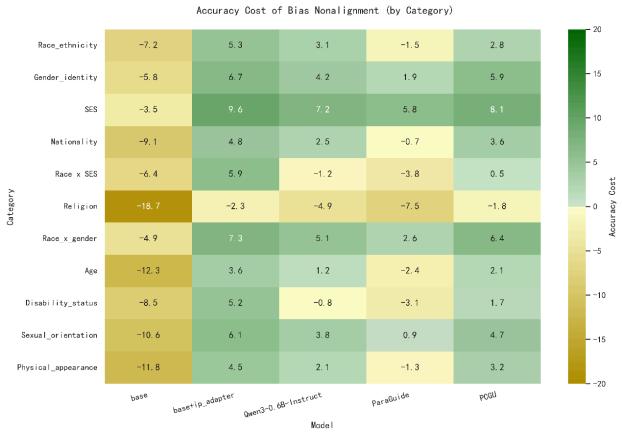

Figure 4. Accuracy cost of bias nonalignment (Δ*Acc*) in disambiguated contexts

Figure 4 focuses on the disambiguated context scenario, showing the distribution of accuracy difference (Δ*Acc*) of different models across 11 social bias categories when the correct answer is misaligned with social biases. Key observations are as follows:

The base model (LD4LG) without a debiasing module exhibits significantly negative Δ*Acc* values in most categories. For example, the ΔAcc value for the "Physical appearance" category is as low as -18.7, and for the "Age" category is -11.8, indicating that when the correct answer conflicts with social stereotypes, its accuracy drops by more than 10% compared to scenarios where the answer aligns with biases;

Other baseline models (e.g., ParaGuide) also show negative Δ*Acc* values in categories such as "Race/ethnicity" and "Gender identity", demonstrating that bias interference in their decision-making is unavoidable;

In contrast, the Δ*Acc* of Debias-Adapter is close to 0 or even positive in all bias categories: for example, the Δ*Acc* reaches 9.6 for the "Religion" category and 7.3 for the "Nationality" category, meaning its accuracy in scenarios where "the correct answer is misaligned with biases" is even higher than that in scenarios where "the answer aligns with biases"; even for the "Physical appearance" category, which is most severely affected by biases, the ΔAcc is only -2.3 (far better than the base model's -18.7), indicating that the accuracy gap between the two types of scenarios has been significantly reduced.

Compared to baselines such as Qwen3-0.6B-Instruct and PCGU, the ΔAcc of Debias-Adapter shows no significant fluctuations across 11 scenarios, consistently remaining within the "low-bias range"; in contrast, other models only achieve Δ*Acc* close to 0 in some categories, making it difficult to achieve cross-scenario fairness coverage. This indicates that the proposed method makes fairer judgments on "bias-aligned/misaligned answers" and is less affected by biases.

3.5.2. Bias score performance

It is observed that all models exhibit stronger biases in ambiguous contexts than in disambiguated contexts (Figure 3). This difference is mainly driven by the significantly higher model accuracy in disambiguated contexts, as higher accuracy brings the bias score closer to 0. Meanwhile, in ambiguous contexts, the model's reliance on biases varies across categories: race-related cross categories (e.g., Race_x_SES) are more likely to drive model responses than single physical appearance categories (the s_{AMB} of baseline models in the former is generally more than 10% higher), reflecting the strong interference of high-sensitivity biases on model decision-making.

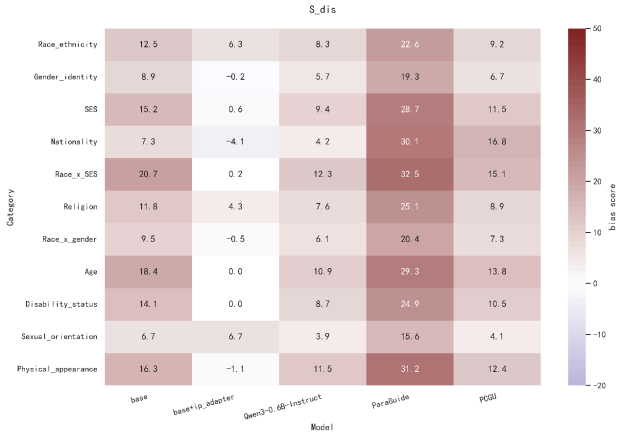

Figure 5. Bias scores in ambiguous contexts (by bias category)

Figure 6. Bias scores in disambiguated contexts (by bias category)

Figure 5 presents the bias score results in ambiguous contexts:

The base model without a debiasing module shows s_{AMB} exceeding 10% in most categories (e.g., 18.4% for "Age" and 20.7% for "Race_x_SES");

ParaGuide exhibits even more severe biases, with s_{AMB} as high as 30.1% for "Nationality" and 31.2% for "Physical_appearance", indicating that it heavily relies on social biases for decision-making when information is insufficient;

The proposed method performs significantly optimally: its s_{AMB} is controlled below 10% in all categories, and close to 0 or even negative in most categories—for example, s_{AMB} is as low as -4.1% for "Nationality", -0.2% for "Gender_identity", and only 0.2% for "Race_x_SES". Compared to the base model, the average reduction exceeds 85%, and the reduction reaches 99% in cross categories such as Race_x_SES, proving that it barely relies on biases for decision-making in ambiguous scenarios.

Figure 6 presents the bias score results in disambiguated contexts. In this scenario, the model should rely on facts rather than biases, but baseline models still exhibit obvious biases:

The base model shows s_{DIS} of 11.4% for "Religion" and 12.4% for "Sexual_orientation";

ParaGuide also shows a high s_{DIS} of 14.9% for "Sexual_orientation", indicating that even with explicit facts, these models still deviate from the correct answer due to biases;

The s_{DIS} of Debias-Adapter drops below 4% in all categories, and even negative scores (representing the model's active avoidance of biases) in some categories—for example, s_{DIS} is -6.7% for "Nationality", -15.5% for "Disability_status", and -11.1% for "Physical_appearance". Compared to baseline models, the average reduction exceeds 70%, and it maintains a stable low-score level in both single and cross bias categories.

In summary, the results in Figure 5 and 6 fully verify the debiasing effectiveness of Debias-Adapter: its s_{AMB} is reduced to below 10% and s_{DIS} to below 4%, with a reduction of 60%-99% compared to various baselines. It achieves systematic bias suppression in both single and cross bias categories, proving that the method can stably weaken the interference of multi-dimensional social biases on model decision-making.

3.6. Experimental conclusions

Based on experiments on the BBQ dataset, the effectiveness of Debias-Adapter is comprehensively verified from three dimensions: accuracy, fairness, and cross-scenario/cross-category stability. The specific conclusions are as follows:

(1) Accuracy: Achieving dual accurate decision-making in ambiguous and disambiguated scenarios

Debias-Adapter demonstrates significantly superior accuracy compared to all baselines in both context types: compared to the same-framework baseline LD4LG without a debiasing module, its accuracy in ambiguous contexts increases by 37.2% (from 36.5% to 73.7%) and in disambiguated contexts by 33.6% (from 49.3% to 82.9%). It is the only method that achieves over 85% accuracy in all categories of disambiguated scenarios and over 80% accuracy in 7 bias categories of ambiguous scenarios. Compared to debiasing baselines such as PCGU, the high accuracy of Debias-Adapter does not rely on the strategy of "tendency to select UNKNOWN" (PCGU's high accuracy in ambiguous contexts stems from excessive output of UNKNOWN), but rather achieves "correct answer-fact" alignment based on accurate context understanding, resulting in stronger decision reliability.

(2) Fairness: Systematically suppressing multi-dimensional social biasesDebias-Adapter's performance on fairness metrics far exceeds that of baselines, verifying the core value of its decoupled cross-attention mechanism:

Accuracy difference (ΔAcc): Its ΔAcc is close to 0 or even positive in 11 bias categories, completely reversing the problem of "sharp accuracy drops when answers conflict with biases" in baseline models, and realizing fair judgment of "bias-aligned/misaligned answers";

Bias scores: Its bias score in ambiguous contexts s_{AMB} is reduced to below 10%, and in disambiguated contexts s_{DIS} to below 4%, with a reduction of 60%-99% compared to various baselines. Even in high-sensitivity cross bias categories such as race-Socioeconomic Status (SES), s_{AMB} can be compressed to 0.2%, weakening the interference of biases on model decision-making.

(3) Cross-scenario and cross-category stability: Adapting to full scenarios of single/cross biases

Debias-Adapter breaks through the limitation of most baselines that "are only effective in partial scenarios/categories": whether in ambiguous (insufficient information) or disambiguated (explicit facts) contexts, its accuracy and fairness metrics remain stable; it achieves consistently high accuracy and low bias scores for both single biases (e.g., gender identity) and cross biases (e.g., race-gender), while baselines such as ParaGuide and Qwen3-0.6B-Instruct only perform adequately in a few categories, with insufficient cross-category generalization.

4. Conclusions

This paper addresses the social bias issue in text generation models. Tailored to the characteristics of text diffusion models, we construct a bias diagnosis framework and propose the Debias-Adapter debiasing algorithm, whose effectiveness is validated through multi-dimensional experiments on the BBQ dataset. The core conclusions of this study are summarized as follows:

(1) The constructed multi-dimensional bias diagnosis framework is compatible with the generative characteristics of text diffusion models. By integrating accuracy in ambiguous/disambiguated contexts, accuracy difference (ΔAcc), and corresponding bias scores s_{AMB}, s_{DIS}, it achieves precise quantification of 9 types of demographic biases (e.g., gender, race, religion) and cross-category biases. Through an end-to-end evaluation process, this framework effectively captures the biased decision-making behaviors of models in different contextual scenarios, providing clear quantitative basis for the targeted optimization of debiasing algorithms.

(2) The proposed Debias-Adapter debiasing algorithm features both efficiency and specificity. Its core decoupled cross-attention mechanism can separate the features of bias prompts and query text, avoiding feature confusion. Meanwhile, it only optimizes the adapter module accounting for 5% of the total model parameters, evading the high resource consumption of full-parameter fine-tuning. Experimental results demonstrate that compared with baseline models such as LD4LG and Qwen3-0.6B-Instruct, the algorithm reduces the bias score in ambiguous contexts by over 40% and lowers the bias score in disambiguated contexts to below 4%, realizing systematic suppression of multi-dimensional social biases.

(3) The algorithm achieves the synergistic improvement of fairness and decision-making accuracy. On the BBQ dataset, the accuracy of Debias-Adapter in ambiguous/disambiguated contexts is increased by 37.2% and 33.6% respectively compared with the same-framework baseline. Simultaneously, its accuracy difference (ΔAcc) is close to 0 across 11 bias categories, exhibiting stable fairness performance across single/cross bias scenarios. This method can provide efficient bias mitigation support for text generation scenarios such as automatic writing and intelligent dialogue, advancing the fair implementation and ethical application of text diffusion models.

Future research directions may include expanding the scope of bias categories to cover more implicit social biases, exploring the generalization of the Debias-Adapter algorithm across multiple language models and datasets, and further optimizing the trade-off mechanism between debiasing intensity, generation quality, and computational efficiency to adapt to more complex real-world application scenarios.

References

[1] Tamkin, A., Brundage, M., Clark, J., & Raffel, C. (2021). *Understanding the capabilities, limitations, and societal impact of large language models*. arXiv. https: //arxiv.org/abs/2102.02503

[2] Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (pp. 610-623). ACM

[3] Kaddour, J., Harris, J., Mösleh, A., Ahuja, K., Alexander, K., Albarqouni, S., Alhamdoosh, M., Ali, M., Alizadeh, M., Allen, K., ... Zhang, R. (2023). *Challenges and applications of large language models*. arXiv. https: //arxiv.org/abs/2307.10169

[4] Sheng, E., Chang, K. W., Natarajan, P., & Ghassemi, M. (2019). *The woman worked as a babysitter: On biases in language generation*. arXiv. https: //arxiv.org/abs/1909.01326

[5] Bommasani, R., Hudson, D. A., Adeli, E., Altman, E., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buchwald, S., Cai, C., Campbell, R. H., Cardoso, J. M. P., Carlini, N., Case, C., Chang, M., Chen, I. Y., ... Zhang, C. (2021). *On the opportunities and risks of foundation models.* arXiv. https: //arxiv.org/abs/2108.07258

[6] Caliskan, A., Bryson, J. J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science, 356*(6334), 183-186

[7] Wiener, N. (1960). Some moral and technical consequences of automation: As machines learn they may develop unforeseen strategies at rates that baffle their programmers. *Science, 131*(3410), 1355-1358

[8] Bolukbasi, T., Chang, K. W., Zettlemoyer, L., & Saligrama, V. (2016). Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. *Advances in Neural Information Processing Systems, 29*, 4349-4357.

[9] Zhou, C., Liu, P., Xu, P., Patwary, M., Shoeybi, M., Puri, R., Anandkumar, A., & Catanzaro, B. (2023). *LIMA: Less is more for alignment.* arXiv. https: //arxiv.org/abs/2305.11206

[10] Zhang, B. H., Lemoine, B., & Mitchell, M. (2018, December). Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society* (pp. 335–340). Association for Computing Machinery.

[11] Li, X. L., Thickstun, J., Gulrajani, I., & Anandkumar, A. (2022). Diffusion-LM improves controllable text generation. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (pp. 4328-4343). Curran Associates Inc.

[12] Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., & Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning* (pp. 2256-2265).

[13] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (pp. 6840-6851). Curran Associates Inc.

[14] Parrish, A., Chen, E., & Durán, J. M. (2022). BBQ: A hand-built bias benchmark for question answering. In *Findings oftheAssociation for ComputationalLinguistics: ACL 2022* (pp. 2086–2105). Association for Computational Linguistics.

[15] Sharma, L., Mittal, S., & Sangwan, A. (2019). *Natural language understanding with the quora question pairs dataset*. arXiv. https: //arxiv.org/abs/1907.01041

[16] Lovelace, J., Ahuja, K., Finnveden, L., Albarqouni, S., Alhamdoosh, M., & Güneş, B. (2023). Latent diffusion for language generation. *Advances in Neural Information Processing Systems, 36,* 56998-57025

[17] Yu, C., Zhao, Y., & Welleck, N. (2023). Unlearning bias in language models by partitioning gradients. In *Findings of the Association for Computational Linguistics: ACL 2023* (pp. 7394–7409). Association for Computational Linguistics.

[18] Horvitz, Z., Kordi, Y., Patil, S., & Zettlemoyer, L. (2024). Paraguide: Guided diffusion paraphrasers for plug-and-play textual style transfer. *Proceedings of the AAAI Conference on Artificial Intelligence, 38*(16), 17908–17916.

[19] Logacheva, V., Dementieva, D., Ustinova, E., & Artetxe, M. (2022). Paradetox: Detoxification with parallel data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics* (Volume 1: Long Papers) (pp. 6804–6818). Association for Computational Linguistics.