Leveraging Diffusion and Flow Matching Models for Demographic Bias Mitigation of Facial Attribute Classifiers

Sreeraj Ramachandran[1] and Ajita Rattani[1]

[1]Affiliation not available

December 14, 2023

Abstract

Published research highlights the presence of demographic bias in automated facial attribute classification algorithms, notably im-pacting women and individuals with darker skin tones. Proposed bias mitigation techniques are not generalizable, need demographic annotations , are application-specific, and often obtain fairness by reducing overall accuracy. In response to these challenges, this paper proposes a novel bias mitigation technique that systematically integrates diffusion and flow-matching models with a base classifier with minimal additional computational overhead. These generative models are chosen for their extreme success in capturing diverse data distributions and their inherent stochasticity. Our proposed approach augments the base classifier's accuracy across all demographic subgroups with enhanced fairness. Further, the stochastic nature of these generative models is harnessed to quantify prediction uncertainty, allowing for test-time rejection, which further enhances fairness. Additionally, novel solvers are proposed to significantly reduce the computational overhead of generative model inference. An exhaustive evaluation carried out on facial attribute annotated datasets substantiates the efficacy of our approach in enhancing the accuracy and fairness of facial attribute classifiers by 0.5% - 3% and 0.5% - 5% across datasets over SOTA mitigation techniques. Thus, obtaining state-of-the-art performance. Further, our proposal does not need a demographically annotated training set and is generalizable to any downstream classification task.

Leveraging Diffusion and Flow Matching Models for Demographic Bias Mitigation of Facial Attribute Classifiers

Sreeraj Ramachandran and Ajita Rattani

Abstract—Published research highlights the presence of demographic bias in automated facial attribute classification algorithms, notably impacting women and individuals with darker skin tones. Proposed bias mitigation techniques are not generalizable, need demographic annotations, are application-specific, and often obtain fairness by reducing overall accuracy.

In response to these challenges, this paper proposes a novel bias mitigation technique that systematically integrates diffusion and flow-matching models with a base classifier with minimal additional computational overhead. These generative models are chosen for their extreme success in capturing diverse data distributions and their inherent stochasticity. Our proposed approach augments the base classifier's accuracy across all demographic sub-groups with enhanced fairness. Further, the stochastic nature of these generative models is harnessed to quantify prediction uncertainty, allowing for test-time rejection, which further enhances fairness. Additionally, novel solvers are proposed to significantly reduce the computational overhead of generative model inference.

An exhaustive evaluation carried out on facial attribute annotated datasets substantiates the efficacy of our approach in enhancing the accuracy and fairness of facial attribute classifiers by $0.5\% - 3\%$ and $0.5\% - 5\%$ across datasets over SOTA mitigation techniques. Thus, obtaining state-of-the-art performance. Further, our proposal does not need a demographically annotated training set and is generalizable to any downstream classification task.

Index Terms—Diffusion Models, Fairness in AI, Flow Matching Models, Facial Attribute Classifier

1 Introduction

With the increasing reliance on Artificial Intelligence (AI) for decision-making in high-impact situations such as risk assessment in criminal justice, patient diagnosis in healthcare, and credit scoring in financial lending, it is imperative that such systems do not exhibit discrimination [1]. However, recent research has raised several fairness concerns about these systems, with researchers finding significant accuracy disparities (bias) across demographic groups. Fairness is the absence of prejudice or favoritism toward an individual or a group based on their inherent or acquired characteristics. Thus, an unfair (biased) algorithm is one whose decisions are skewed toward a particular group of people. The facial-analysis-based algorithms are at the center stage of this discussion [2], [3].

Automated facial analysis-based algorithms encompass face detection, face recognition, and facial attribute classification (including gender-, race-, age classification, and BMI prediction). Numerous existing studies investigating the fairness of facial analysis-based algorithms confirm the performance disparities of these algorithms for people of color (such as African-Americans) and females [2], [4];

These facial-analysis-based algorithms are deeply integrated into various sectors, such as surveillance and border control, retail and entertainment, healthcare, and education. Given their far-reaching impact, the need to deploy accurate and unbiased facial analysis-based systems becomes not just essential but urgent. Thus, bias in these systems emerges as a significant societal issue that warrants immediate redress, particularly for the large-scale deployment of fair and trustworthy facial-analysis-based algorithms across demographics.

Along this direction, several bias mitigation techniques have been proposed by the vision community for these facial-analysis-based algorithms. Established bias mitigation techniques utilize regularization [5], attention mechanism [6], adversarial debiasing [7], [8], GAN-based oversampling [9], [10], multi-task classification [11], and network pruning [12]. Most of these techniques predominantly fall into the category of **in-processing techniques** that introduce fairness-related constraints during model training. However, these existing in-processing techniques often require demographically annotated training sets, are limited in their generalizability, and are computationally expensive. Furthermore, these techniques often sacrifice overall classification accuracy by diminishing the performance of the best-performing demographic group in pursuit of improved fairness, making them *Pareto inefficient* with respect to group accuracy [7]. **Importantly**, these existing bias mitigation techniques focus predominantly on improving accuracy or fairness but overlook the crucial aspect of capturing the full data distribution and uncertainty estimation of individual sample predictions. This is a significant *gap*, as estimating uncertainty is pivotal for risk mitigation and the model's trustworthiness.

Diffusion and Normalizing Flows: Diffusion models [13] belong to a specialized category of generative mod-

Sreeraj Ramachandran is with the School of Computing, Wichita State University, Kansas, USA. email: sxramachandran2@shockers.wichita.edu
Ajita Rattani is with the Dept. of Computer Science and Engineering, University of North Texas at Denton, Texas, USA. email: ajita.rattani@unt.edu

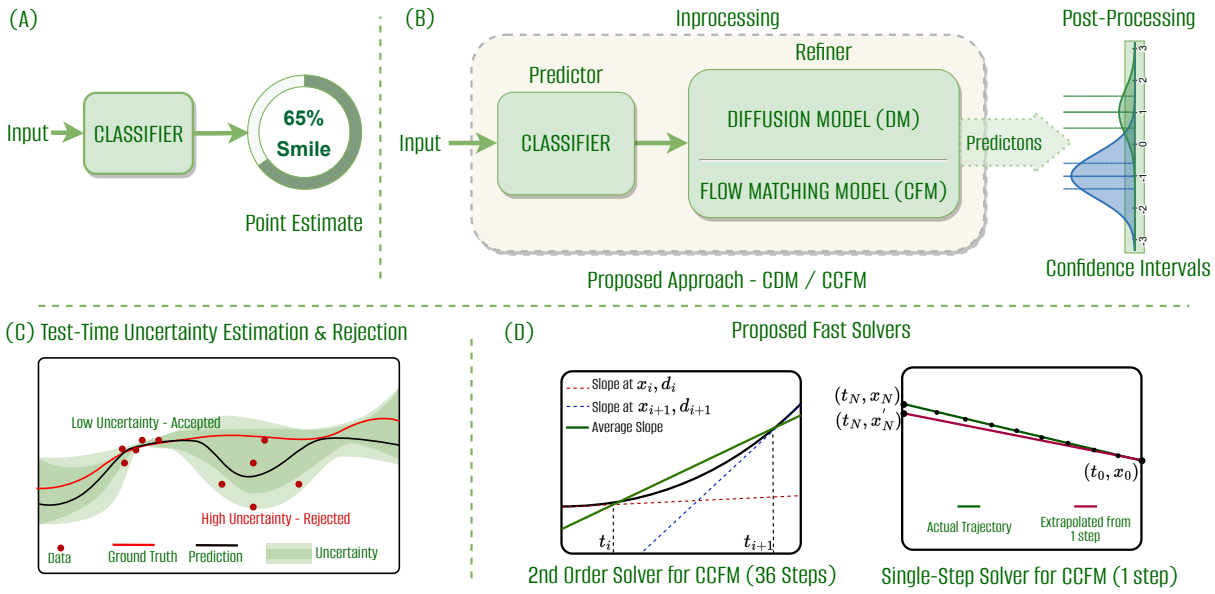


Fig. 1. Overview of the Proposed Method: (A) Traditional classifiers obtain point estimates for probability scores. (B) **Proposed Approach**: We jointly train a generative model and a classifier to function as a predictor-refiner system that captures the full data distribution during the in-processing stage. During evaluation, we can make multiple predictions on a single sample via the diffusion process to estimate confidence intervals in the post-processing stage. (C) These estimated confidence intervals enable the identification and rejection of samples with low or high uncertainty. (D) **Proposed Fast Solvers**: We introduce rapid second-order solvers using higher-order derivatives specifically for our CCFM model, significantly speeding up inference. An even faster single-step solver approximates the output through extrapolation, rendering the inference time for CCFM virtually negligible.

els designed to simulate a diffusion process, wherein a simple initial distribution gradually transforms through a series of random perturbations to eventually approximate a complex target data distribution. These models excel in capturing intricate data structures and can generate high-quality, high-dimensional samples, such as photo-realistic images or intricate audio sequences. Methods such as score matching [14], [15] offer a way to train diffusion models by optimizing an energy-based model to minimize the difference between the observed and generated data, helping them better approximate the target data distribution.

Similarly, an alternative approach to diffusion models is Continuous Normalizing Flows (CNFs) [16], another class of generative models that model a data distribution by transforming simple data distributions into more complex ones in a smooth and reversible manner. Methods such as flow Matching (FM) [17] offer a simulation-free approach for training CNFs by matching vector fields along fixed conditional probability paths, providing a more robust and efficient alternative to traditional diffusion and score-matching models. Readers are referred to Appendix **A** for detailed background information on diffusion and flow-matching models.

CARD: Building on the foundational principles of diffusion models, a noteworthy variant was introduced, called the Classification and Regression Diffusion (CARD) model [18]. These models uniquely merge a denoising diffusion-based conditional generative model with a pre-existing conditional mean estimator (classifier). This fusion accomplishes two main objectives: Firstly, it enables highly accurate prediction of the distribution of the output label given the input image sample, whereby a diffusion model iteratively refines (**post-process**) the prediction outcome of a classifier model. Secondly, it capitalizes on the inherent stochasticity of the generative model's outputs to yield a more nuanced, instance-level confidence assessment (uncertainty estimation) of the main classification task.

In *response* to the aforementioned challenges with existing bias mitigation techniques, we *leverage* diffusion and flow matching models **for the first time** for demographic bias mitigation of the facial attribute classification task in this study. To facilitate this, these diffusion and flow-matching models are utilized and improved as follows:

Utilizing generative models for full data distribution capture: To this end, we first adapt the existing Classification Diffusion Model (CDM) [18] in the context of mitigating demographic bias of the facial attribute classification task. In adapting CDM, we simultaneously address a prevalent issue in existing biometric classifiers: the lack of robust uncertainty quantification beyond basic softmax probabilities. Specifically, we can attach a diffusion model to the end of a given classifier, allowing it to be trained either independently or fine-tuned alongside the base classifier. This combination of classifier and diffusion acts as a predictor and refiner, respectively, and can capture the target data distribution better than a single classifier. Diffusion models, due to their stochastic nature, produce unique outputs with each inference, allowing for confidence interval estimation around predictions and thereby enabling test-time rejection. Further, when the base classifier is fine-tuned with the diffusion head, this improves generalization and enhances the fairness as well as trustworthiness of the base classifier.

Improving the inference bottleneck of diffusion models: However, diffusion models suffer from slow inference due to their reliance on solving the underlying differential equations, which demands hundreds of neural network evaluations (NFE) per prediction. This bottleneck is exacerbated when leveraging the model's stochasticity for multiple output predictions. To tackle this, we adopt a 2nd order solver that uses higher-order derivatives for faster convergence, thereby reducing the NFE and improving the inference time.

A better faster solution: Finally, we introduce a novel approach of integrating flow-matching models with a base classifier to mitigate demographic bias for the *first time*. We term our proposal as **Classification Conditional Flow Matching (CCFM)** model. Our proposed CCFM offers several performance advantages over CDM, including faster inference, stable training, and better performance. Like diffusion models, CCFM is also rooted in differential equations. Therefore, we also propose a similar 2nd order solver specifically tailored for CCFM. This drops neural network evaluations (NFE) from 101 to 36 steps compared to first-order solvers. Further, exploiting the model's linear behavior during inference, we develop a *single-step solver* that eliminates the computational overhead. Given that our diffusion model architecture is lightweight (1.2M params) compared to base classifiers like ResNet-18 (11.7M params), the additional computational overhead is minimal.

Summarized Contribution: An *overview* of our proposed approach[1] is illustrated in Figure 1. In summary, we investigate the effectiveness of diffusion and flow-matching models in enhancing the demographic fairness of facial-attribute classifiers for the first time. We have used (a) face-based gender classification with race as the protected attribute and (b) face-based multi-attribute classification with gender as the protected attribute, as the case studies in this work.

The specific contributions are enumerated as follows:

1) We propose to use a **classification diffusion model (CDM)**, which is a combination of a base classifier and a diffusion model, in the context of bias mitigation for the *first time*. When fine-tuned along with the base classifier, this model is shown to have better generalization and improved fairness. Joined together (base classifier and the diffusion model), this can be seen as an *in-processing* bias-mitigation approach targeted at reducing the variance of the combined model and thereby improving the generalization, a factor which has shown to improve fairness [9].
2) Further, the stochastic nature of diffusion models allows us to generate multiple output prototypes for the same input sample and can estimate confidence intervals from it. This enables us to use the modeled uncertainty for the **test-time rejection** of uncertain samples. This can be seen as a *post-processing operation* [1] applied on top of the classification prediction.
3) We propose a new class of **novel classification conditional flow-matching (CCFM) model** that is a combination of a base classifier and a flow-matching model. This offers the same properties of CDMs with faster inference, improved training stability, and overall generalization, as well as certain additional properties such as linear trajectories, which we will exploit further for even faster inference.
4) We propose two **fast solvers** for the CCFM class of models. The first is a 2nd order solver that brings the number of network evaluations from 101 to 36. We subsequently utilize the linear nature of solution trajectories to propose a single-step solver, effectively erasing the inference speed bottleneck of such generative models and thereby enabling fast confidence interval predictions.
5) Lastly, we did a **thorough evaluation** of our proposed approach on several facial attributes annotated datasets for the facial attribute classification task (both single and multi-attribute classification) with gender and race as the protected attributes. Additionally, we also demonstrated the efficacy of our proposed approach in mitigating the bias of ocular modality (see Appendix B.4).

Our contributions facilitate significant improvements in the generalization performance (accuracy) as well as the advancement of algorithmic fairness. Our method obtained SOTA results on max-min fairness (improvement on the most-disadvantaged group) (refer Appendix A.1 for more details on fairness preliminaries) and generalization performance on gender classification with race as the protected attribute and multi-attribute classification task (13 attributes) using Fairface [19] and CelebA [20] datasets, respectively. UTKFace [21], DiveFace [22], and Morph [23] datasets have been used for cross-dataset evaluation of the proposed models (out-of-distribution). We also evaluate the efficacy of our methods on ocular and periocular modalities using datasets such as VISOB [24], UFPR [25], Noredame-NIVL [26] and NDIris [27] (refer Appendix B.4). Our implementation codes and pre-trained models will be made *publicly available* at the GitHub link at the time of publication.

The paper is structured as follows: Section 2 describes relevant literature related to bias mitigation. Section 3 revisits existing models such as CARD [18] and discusses our modifications in terms of architecture as well as inference. Section 4 introduces our innovative 'Classification Flow Matching Models'. Section 5 explores the design choices behind our model development. In Section 6, we present the results of the experimental validation of our proposed models in bias mitigation. Finally, Section 7 discusses conclusions drawn and potential future research directions.

2 Related Work

In this section, we offer a review of the academic literature, focusing on the evaluation and mitigation of biases inherent in facial attribute classification algorithms. We systematically structure the discussion into two primary subsections: (1) investigation of bias in algorithms and (2) strategies for bias mitigation.

1. The terms approach, method, model, and strategy are used interchangeably in this work.

Investigation of Bias: Many studies have highlighted the systematic limitations of facial-attribute classification algorithms, especially concerning their performance with specific gender-racial groups. Biases of these algorithms can lead to disproportionate misclassification of certain groups, thus perpetuating societal inequities [1]. Studies such as [28] and [29] have shown that face-based gender classification algorithms have higher error rates for darker-skinned individuals and women and that factors such as age, hair length, and facial hair presence may be contributing to these disparities. A study by [2] evaluated the fairness of various CNN architectures for gender classification and found that the bias of the classifier varied across CNNs, with a substantial increase in misclassification errors for black females. Whereas a study by [30] determined through controlled experimentation that computer vision algorithms often reflect societal biases in gender and race identification due to skewed training datasets.

Mitigation of Bias: As the existence of demographic bias has been confirmed using the aforementioned studies, numerous strategies have been proposed to mitigate it [1]. [31] introduced a novel adversarial learning-based encoder to obtain race-invariant representations for gender classification. This model was tested on the UTKFace dataset and showed promising results.

[32] applied Domain Discriminative methods (DD) to mitigate bias by using a probabilistic method to adapt object classification systems at prediction time without retraining, effectively reducing error rates when applied to correlated, sequentially occurring images. [33] designed to accomplish three key objectives: 1) neutralize identifiable biases present in the dataset, 2) enhance classification performance under extreme bias, and 3) eliminate various extraneous variations from the feature representation for the primary facial attribute classification task through a process of joint learning and unlearning.

[34] introduced a semi-supervised method that exploits a type of group of fairness constraint expressed over large quantities of unlabeled data to build a better classifier and observe the improvement in the accuracy as well as fairness.

[8] proposed "fair mixup," a data augmentation technique that improves the generalization of the classifiers trained under group fairness constraints. Specifically, it trains the model on new samples created by blending or interpolating data points from different groups, thereby encouraging the model to make fairer predictions and demonstrating its efficacy across tabular, vision, and language benchmarks.

Furthermore, mitigation techniques built on generative perspectives have been proposed by [10], [35] and [9]. [10] adapts structured learning and proposes a method called NSL where neighboring views of the input sample are generated using a generative model, and then a neighbor loss that minimizes the distance between neighbors is applied as a regularizer. Their proposed method obtains state-of-the-art fairness on the FairFace dataset. While [10] utilized GAN-based latent vector editing in tandem with structured learning to alleviate gender classification bias, [35] and g-SMOTE proposed by [9] used the same technique to strategically augment the training set to mitigate bias.

Readers are referred to [5] for a comprehensive list of studies on the examination and mitigation of facial attribute classifiers.

3 Approach: Revisiting Classification Diffusion Models

In this section, we start with an existing class of classification diffusion model [18] and adapt it to a modern framework introduced by [36] called Elucidating Diffusion Models(EDM) that offers a general framework for diffusion models, and we introduce additional architectural and design choices to improve its overall performance **both in terms of generalization performance, fairness and also inference speed**. Divided into three subsections, in section 3.1, we discuss the fundamental aspects of employing diffusion models in a classification context. In 3.2, we discuss the Classification Diffusion Model (CDM) by expanding it into the context of EDM. Lastly, 3.3 subsection elaborates on how sampling can be done by breaking down continuous data into discrete units. A full mathematical background on the diffusion model is omitted here for brevity. Refer to Appendix A.2 for a more detailed background.

3.1 Diffusion for Classification

Given a dataset, $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, where each $x_i \in X$ is a feature vector and each $y_i \in Y$ is the corresponding class label, the classifier is a function $f : X \to Y$ that tries to map feature vectors to labels. Let's denote $P(Y|X)$ as the conditional probability of a label Y given the feature vector X. For any given instance with features x_i, the classifier assigns it to the class y_j that maximizes the conditional probability $P(Y = y_j|X = x_i)$. Standard classification models often output a deterministic function $f(\mathbf{x})$ that characterizes the class probabilities, which can be seen as estimates of $\mathbb{E}[\mathbf{y}|\mathbf{x}]$. While these models provide some level of uncertainty through these probabilities, they do not capture the full conditional distribution of the target variable given the features, limiting their ability for comprehensive uncertainty estimation. By using diffusion models, we aim to accurately recover the full distribution of y conditioned on x given $\mathcal{D}$, i.e. $p(y|x, \mathcal{D})$,.

Given a trained classification model f_ϕ that outputs the softmax probability scores $f_\phi(\mathbf{x})$. We want to train a diffusion model that takes $f_\phi(\mathbf{x})$ as a conditioning signal and iteratively refines it during the time-step evolution. We apply the following simplification. In CARD, the endpoint of the diffusion process is set to a distribution centered around $f_\phi(\mathbf{x})$, which involves significant mathematical modifications to the corresponding diffusion equations. In our approach, we use $f_\phi(\mathbf{x})$ as a conditioning signal and set the endpoint of the diffusion to be a standard normal distribution $\mathcal{N}(0, \mathbf{I})$. This simplification allows us to make use of any existing vanilla generative models with minimal modification, i.e., we may develop a diffusion model that takes the output softmax probability distribution from a classification model along with optional feature vectors as conditioning signals and uses it to reconstruct the target label.

3.2 EDM

Expanding on the foundational principles of diffusion models([37], see Appendix A.2), we extend our approach to encompass a broader range of generative models by adapting it into a more modern framework introduced by Elucidating Diffusion Models (EDM) [36]. EDM offers a comprehensive and general framework that accommodates various diffusion model variants. This approach of using EDM is in contrast to the existing method of training CDM [18], which builds on top of the denoising diffusion probabilistic model (DDPM) parameterization [37]. This change allows us to switch between different variants and inference methods without additional work. In the EDM framework, the generalized probability flow ODE (see Appendix A.2.3 for more details) is given by

$$d\mathbf{x} = \left[\frac{\dot{s}(t)}{s(t)}\mathbf{x} - s(t)^2\dot{\sigma}(t)\sigma(t)\nabla_{\mathbf{x}}\log p\left(\frac{\mathbf{x}}{s(t)};\sigma(t)\right)\right]dt \quad (1)$$

where $\sigma(t)$ is the noise schedule, the dot denotes a time derivative, and $s(t)$ is an additional scale schedule, i.e., consider $\mathbf{x} = s(t)\hat{\mathbf{x}}$ be a scaled version of the original, non-scaled variable $\hat{\mathbf{x}}$.

To keep the input and output signal magnitudes to fixed unit variance and to avoid large variations in gradient magnitudes, they also apply preconditioning to both input and output. For an input $\mathbf{x} = \mathbf{y} + n$ which is a combination of clean signal $\mathbf{y} \sim p_{data}$ and noise $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, the denoiser D_θ is therefore given by

$$D_\theta(\mathbf{x};\sigma) = c_{skip}(\sigma)\mathbf{x} + c_{out}(\sigma)F_\theta\left(c_{in}(\sigma)\mathbf{x}; c_{noise}(\sigma)\right), \quad (2)$$

where F_θ is the neural network to be trained, $c_{skip}(\sigma)$ modulates the skip connection, $c_{in}(\sigma)$ and $c_{out}(\sigma)$ scale the input and output magnitudes and $c_{noise}(\sigma)$ maps noise level σ into a conditioning input for F_θ. The updated training loss, therefore, becomes

$$\mathbb{E}_{\sigma,\mathbf{y},\mathbf{n}}\Big[\underbrace{\lambda(\sigma)\ c_{out}(\sigma)^2}_{\text{effective weight}}\Big\|\underbrace{F_\theta\big(c_{in}(\sigma)\cdot(\mathbf{y}+\mathbf{n}); c_{noise}(\sigma)\big)}_{\text{network output}} - \underbrace{\tfrac{1}{c_{out}(\sigma)}\big(\mathbf{y} - c_{skip}(\sigma)\cdot(\mathbf{y}+\mathbf{n})\big)}_{\text{effective training target}}\Big\|_2^2\Big] \quad (3)$$

where $\lambda(\sigma)$ is the time-dependent loss weighting. Deterministic variants of different model families are obtained by choosing appropriate design choices for these individual components. Table 1 of [36] details the specific design choices for Variance Preserving (VP), Variance Exploding (VE), and their proposed EDM formulations we will be appropriating. The Variance Preserving (VP) [37] and Variance Exploding (VE) [38] formulations, as outlined in these specific choices, contrast in their approach to variance management during the diffusion process: VP aims to stabilize variance, while VE allows for its increase, each presenting unique implications for model performance.

3.3 Solution by Discretization

Transitioning from the theoretical framework, we now delve into the practical aspects of solving the ODEs that govern the dynamics of our diffusion models. During inference, we get the outputs and other conditioning signals from the base classifier model. But for the diffusion model, the inference is done by integrating the ODE given by the Eq. 1. To solve the ODE in Eq. 1, numerical integration techniques are generally employed, which require finite step computations over discrete time intervals. When using numerical integration, two crucial choices must be made: the integration scheme, such as Euler's method or a variant of Runge-Kutta, and the selection of discrete sampling times denoted as $\{t_0, t_1, ..., t_N\}$. The choice of the number of timesteps and algorithm is significant, as the diffusion component of the model must be evaluated many times. This requirement serves as a key limitation for the broader adoption of diffusion-based models in this context. Although Euler's method has been the go-to choice for solving such ODEs in earlier CDM research, we adopt the 2nd-order solver proposed by [36], which offers a superior computational trade-off. Alongside this, we also follow their recommended time-step discretization method for the integration process.

4 Approach: Proposed Classification Flow Matching Models

In the preceding section, we discussed enhancements to established Classification Diffusion Models (CDMs). In this section, we introduce a **novel hybrid model** that combines elements of classification and flow-matching generative models, which we term the Classification Conditional Flow Matching Model (CCFM). This model adapts principles from a subset of generative models known as conditional flow matching models (refer Appendix A.3) for use in a classification context, similar to CDMs.

Although both models are governed by Ordinary Differential Equations (ODEs), CCFM employs a much simpler form. We leverage this nature of the ODE and introduce a streamlined, fast second-order solver that utilizes higher-order derivatives to converge to a solution much faster than standard solvers. We then make a crucial observation that the trajectory that a given sample takes from the initial noise to the final sample is of a linear nature. We then capitalize on the linear trajectory nature of the ODEs to develop a single-step algorithm that significantly accelerates inference through extrapolation. This effectively eliminates the slow inference problem that requires hundreds of network evaluations, a performance bottleneck associated with the previously proposed approaches, allowing for a single prediction to be generated with just one run of the network.

Section 4.1 delves into a specific type of flow matching under Gaussian conditions, exploring its functions and potential applications in classification tasks. Following this, in Section 4.2, we examine the application of flow matching in classification scenarios. Next, in Section 4.3, we discuss a refined approach to solving CCFMs using discretization, specifically employing a second-order solver. Lastly, section 4.4 sheds light on an advanced solving technique where extrapolation is used in a single-step solver scenario to enhance performance and efficiency. A full mathematical background on conditional flow matching is omitted here for brevity, refer to Appendix A.3 for detailed background information.

4.1 Gaussian Conditional Flow Matching Model

We employ Gaussian Conditional Flow Matching, a particular variant of CFM, for classification tasks in a manner analogous to the previously discussed CDM approach (Refer Appendix **A.3** for background on flow-matching models). This is obtained by setting, in the CFM formulations, the condition $z := (x_0, x_1)$ where x_0 and x_1 are sampled from $q(x_0)$ and $q(x_1)$ respectively and the conditionals be Gaussian flows between x_0 and x_1 with standard deviation σ. We have

$$q(z) = q(x_0)q(x_1) \tag{4}$$
$$p_t(x|z) = \mathcal{N}(x|tx_1 + (1-t)x_0, \sigma^2) \tag{5}$$
$$u_t(x|z) = x_1 - x_0 \tag{6}$$

with boundary conditions $p_0 = q_0 * \mathcal{N}(x|0, \sigma^2)$ and $p_1 = q_1 * \mathcal{N}(x|0, \sigma^2)$. Both $p_t(x|z)$ and $u_t(x|z)$ are efficiently computable. We can apply conditional flow matching to train our model subsequently.

4.2 Classification Flow Matching Model

Classification Flow Matching Model (CFM) works similarly to CDM. We set the start point of the flow to be a Gaussian source, $q(x_0)$ $\mathcal{N}(0, I)$, and the endpoint to the target distribution, $p(x)$. This allows us to learn a probability density path between these two distributions.

4.3 Solution By Discretization

Building upon the numerical techniques detailed in Section 3.3, we apply similar approaches for solving the ODE that governs CFM. We propose a simplified 2nd-order solver based on the 2nd-order solver proposed by [36] for diffusion models. This scheme offers a superior computational trade-off compared to the traditional Euler method, enhancing efficiency without compromising accuracy. Additionally, we determined that a uniform time-step discretization is both effective and simpler to implement without any loss in performance. A pseudocode is given in Algorithm 1 to generate a sample given a trained model v_θ.

Algorithm 1 Proposed Deterministic 2nd Order Heun solver for CCFM

procedure HEUNSAMPLER($v_\theta(x, t), t_i \in \{0, \dots, N\}$)
  sample $x_0 \sim \mathcal{N}(0, I)$
    ▷ Generate initial sample at t_0
  for $i \in \{0, \dots, N-1\}$ **do**
    ▷ Solve Flow ODE over N time steps
    $d_i \leftarrow v_\theta(t_i, x_i)$ ▷ $\frac{dx}{dt}$ at t_i
    $x_{i+1} \leftarrow x_i + (t_{i+1} - t_i)d_i$
      ▷ Euler step from t_i to t_{i+1}
    $d'_i \leftarrow v_\theta(t_{i+1}, x_{i+1})$ ▷ $\frac{dx}{dt}$ at t_{i+1}
    $x_{i+1} \leftarrow x_i + (t_{i+1} - t_i)(\frac{1}{2}d_i + \frac{1}{2}d'_i)$
      ▷ Explicit trapezoid rule of t_{i+1}

The algorithm given in Algorithm 1 is a deterministic 2nd-order Heun sampler (solver) for CCFM that aims to generate samples based on a flow-matching model. It starts by creating an initial random sample, x_0, from a standard normal distribution. It then iterates through N time steps, updating this sample at each step based on a specific differential equation. For each time step t_i, the algorithm calculates the derivative d_i at that point using the model's vector field $v_\theta(t_i, x_i)$. It makes an initial guess for the next sample x_{i+1} using Euler's method. Following this, it calculates another derivative d'_i at the next time step t_{i+1} using this initial guess. The final value for x_{i+1} is then updated using a weighted average of d_i and d'_i, following the explicit trapezoid rule, aiming for more accurate approximations. This process is designed to provide a more accurate sampling method by incorporating information from higher-order derivatives.

4.4 Proposed Single Step Solver

Although the 2nd-order solver significantly accelerates inference, it still lags behind the efficiency of a single forward pass in a standard classification model. In this section, we introduce an approximate single-step solver designed to bridge this computational gap further.

In section 6.5, we analyze the diffusion trajectories of various models. In the context of diffusion models, a diffusion trajectory refers to the path that a sample takes through the latent space as it evolves over time according to the stochastic differential equation (SDE) governing the model. Essentially, you start with a sample, typically some noise, and then iteratively refine it into the desired data distribution, like an image or text. This sequence of refined samples forms a trajectory in the high-dimensional space.

The trajectory helps visualize or understand how the model moves from the point of randomness toward generating something meaningful. Traditional diffusion models result in complex, curvy paths (trajectories) in the latent space as the sample evolves. However, the CFM (Continuous Flow Matching) model is designed to create straight-line paths that quickly approach the data's average value. The simpler target distribution for the classification tasks offers an advantage here. For example, in binary classification, the output possibilities are limited, often falling into just two categories like 0 or 1. Even if the output is continuous, it's typically easier to separate into these two clear-cut classes.

In contrast, tasks like image generation require much more nuanced outputs, where each pixel's value could fall anywhere between 0 and 1, and even slight deviations can significantly alter the final image. This makes the target distribution in image generation tasks far more complex and harder to approximate. The simplicity of the target distribution in classification tasks allows us to use a single-step solver. This solver can make accurate approximations over arbitrary time intervals through extrapolation from the first step, enabling faster and more efficient computations without sacrificing accuracy. We demonstrate the feasibility of our proposed solver compared to other solvers in Table 6. A pseudocode of the proposed solver is given in Algorithm 2.

Algorithm 2 Proposed Deterministic Single Step solver for CCFM

procedure SINGLESTEPSAMPLER($v_\theta(x,t), t_i \in \{0,\dots,N\}$)
  sample $x_0 \sim \mathcal{N}(0, I)$ ▷ Generate initial sample at t_0
  $d \leftarrow v_\theta(t_0, x_0)$ ▷ $\frac{dx}{dt}$ at t_0
  $b = x_0 - dt_0$ ▷ Calculate y-intercept
  $x_N = dt_N + b$ ▷ Extrapolate to t_N

The algorithm 2 starts by generating an initial sample x_0 from a standard normal distribution. It calculates the rate of change d at that initial time t_0 using the model's vector field $v_\theta(t_0, x_0)$. Instead of iterating through multiple time steps, it directly calculates the y-intercept b by subtracting dt_0 from x_0. Finally, it extrapolates to the end time t_N by calculating x_N as $dt_N + b$, thereby obtaining the final sample in a single step.

5 Design Choices

In this section, we dissect the pivotal decisions that underpin the design of our proposed models, divided into three key subsections. First, in section 5.1, we explain the rationale and benefits behind our choice to normalize the class prototypes in the output layer, detailing its influence on the overall performance. The section 5.2 presents an in-depth discussion on the chosen structure of our diffusion/flow-matching neural network, outlining how it supports and enhances our models' effectiveness. Lastly, in section 5.3, we explore the elements that introduce randomness into our models, illustrating how this stochasticity can benefit the robustness and aid in the calculation of uncertainty.

5.1 Normalizing output prototypes

In a typical classification model, the output is a categorical probability distribution. However, like the approach used in CARD [18], we treat the output of the generative model as continuous data within a state space. This allows us to maintain the framework of the Gaussian diffusion model. During the sampling process, the output y_0 is recreated within the real number range for each dimension rather than as a probability simplex vector. Rather than using one-hot encoding for target class prototypes, we found that applying unit normalization to the class prototypes obtained better results. This is in line with the recommended practice of maintaining consistent input and output signal magnitudes, such as unit variance when training diffusion neural networks [36].

5.2 Network Architecture

We utilize a straightforward 1D UNet-like architecture for the diffusion/flow-matching model. The choice for a 1D UNet-like architecture is motivated by its proven effectiveness in capturing hierarchical features and its computational efficiency, which is critical for real-time diffusion-based bias mitigation. It also allows for seamless integration with the Transformer's sinusoidal position embedding and the original classification model's feature representation, facilitating a more robust and interpretable framework. Figure 2 illustrates the overview of the complete model architecture. For embedding the timestep, we leverage the Transformer sinusoidal position embedding as mentioned. The feature representation of the image, denoted as $\mathbf{x}_{feat}$, is derived using the original classification model. In addition to the output of the previous encoder block, an encoder block also takes the timestep and the feature representation from the classification model as conditioning inputs. Similarly, the decoder block takes the output of the previous layer as input as well as the residual output from the encoder block of the same level as well as the timestep and the feature representation from the classification model as conditioning inputs as seen in Figure 2(left). Hadamard products are used in places where conditioning is applied. The detailed representation of the encoder and decoder block is visualized in Figure 2(right). With a relatively small model size, it consists of 1.2 million parameters and a hidden dimension of 512, thereby incurring minimal computational overhead.

5.3 Source of Stochasticity

The concept of uncertainty is addressed by incorporating the notion of model confidence at the granularity of individual instances. This pertains to the degree of certainty the model possesses regarding each of its predictions, facilitated by the inherent stochasticity of outputs derived from a generative model. Given a consistent set of covariates x, the stochastic nature of the generative model yields a distinct class prototype reconstruction $p(y_i|x)$ with each iteration of reverse process sampling. This capability empowers us to formulate predicted probability intervals for all class labels. In typical diffusion models, the unpredictability or *randomness* during the inference stage comes from two places. First, there's Langevin diffusion (see Appendix A.2 and A.3 for more details), which adds a sort of *noise* that makes each sample slightly different each time you run the model. Second, you start off with an initial random vector ($x_0 \sim \mathcal{N}(0, \sigma^2\mathbf{I})$) sampled from a Gaussian distribution, adding another layer of randomness. But, early on, we decided on the ODE version of diffusion/flow-matching models for simplicity and faster inference, which is a deterministic approach; therefore, the source of randomness must come exclusively from the initial random vector. Each random vector, together with the conditioning signal (classifier prediction + classifier features), produces a different output trajectory, hence a different output. Through our experiments, we identified that this single source of randomness is sufficient to estimate confidence intervals around the output for both CDM and CCFM.

6 Experiments

In this section, we showcase the experimental setup and results of our research. We conduct our evaluations on two major sets of experiments. First, face-based gender classification with Fairface as the training dataset and race as the protected attribute. The second experiment is multi-facial-attribute classification using the CelebA dataset, a standard benchmarking dataset, with gender as the protected attribute and 13 facial attributes chosen as target attributes. Section 6.1 and 6.1 details the experimental setup. We start with a base configuration to which our proposed methods

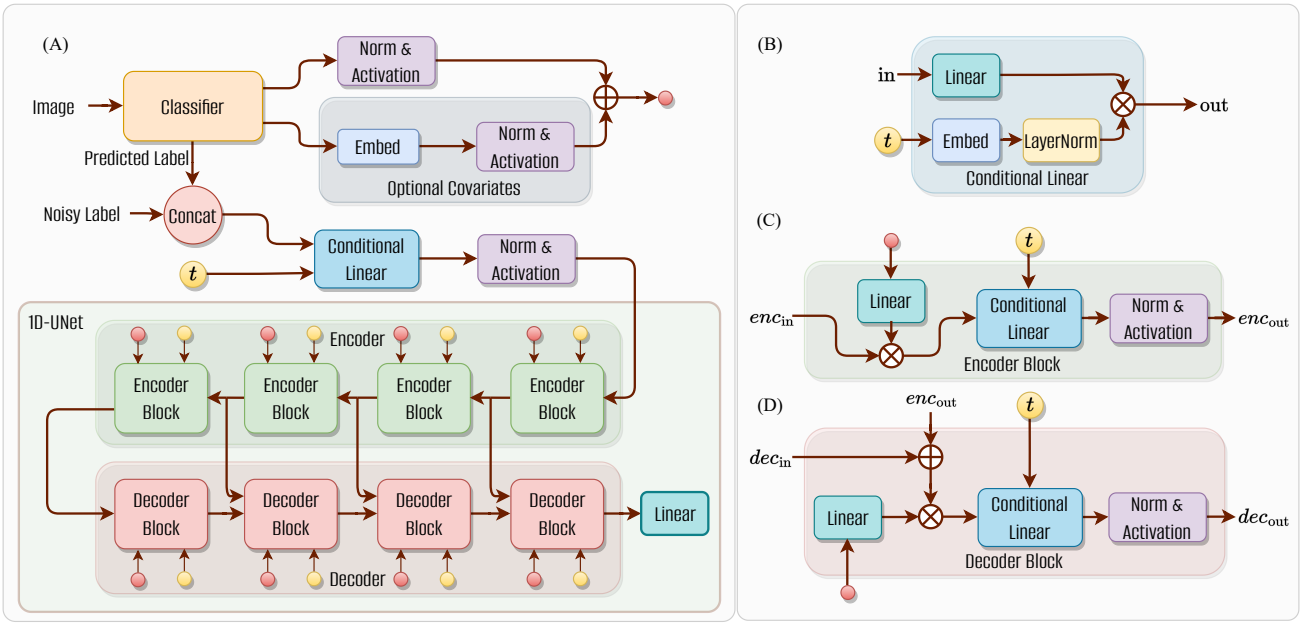


Fig. 2. (A) Overview of Complete Model Architecture. A classifier accepts an image and outputs a predicted label. The noisy true label is concatenated with the predicted label and serves as input to the UNet diffusion model. Optionally, embedded features (covariates) from the classifier are also included, along with a timestep embedding t. (B) A conditional linear block takes an input and applies a linear layer, and a Hadamard product with timestep embedding is used to get the output. (C) The encoder block consists of a conditional linear model followed by a normalization layer. The input to the conditional layer is a Hadamard product of the previous layer output and the conditional signals from the classifier. (D) The Decoder block has a similar structure as the encoder block, with an additional residual connection from the encoder block.

are applied. Section 6.2 to 6.6 incrementally adds modifications to the proposed model and examines its effects. Section 6.7 examines the results on larger models. Section 6.8, 6.9, and 6.10 compare our model with established methods, and test its robustness and generalizability across different datasets.

6.1 Datasets, Training and Testing Configuration

For all our experiments on face-based gender classification, we used the FairFace [19] as our training dataset. Testing was done on the test set of the FairFace as well as DiveFace [22], UTKFace [21], and Morph [23] datasets. For the multi facial-attribute classification task, the CelebA [20] dataset was used for both training and evaluation. Table 1 shows the characteristics of these datasets used in our study.

TABLE 1
Datasets used for training and evaluation

Dataset	Images	Demographic Groups
FairFace [19]	$100k$	White, Black, Indian, East Asian, Southeast Asian, Middle Eastern, Latino Hispanic
DiveFace [22]	$150k$	East Asian, Sub-Saharan,South Indian, Caucasian
UTKFace [21]	$20k$	White, Black, Indian, Asian
Morph [23]	$55k$	White, Black
CelebA [20]	$202K$	Not Available

Training Configuration: In this section, we discuss the specifics of our training configuration and its related aspects. For our experiments, we employ two distinct base classification models[2]. The first is the ResNet-18 [39] model, pre-trained on the Fairface dataset specifically for gender classification and pre-trained on CelebA for multi-attribute classification. The second model we used is the EfficientNetV2-L (NSL) as described in [10] trained on Fairface for gender classification. In the context of fairness, we focus our study on a specific protected attribute, namely the demographic group. Additionally, as the diffusion head, we use the U-Net architecture described in 5.2, which is designed to be computationally efficient.

Two Nvidia A8000 GPUs serve as our computational infrastructure. The training was done in batches of 128 over 1000 epochs. We used an Exponential Moving Average (EMA) of 0.999, a weight decay of 1e-5, and a learning rate of 4e-4 for the CDM/CCFM model. We adjusted the learning rate to 5.6e-6 while fine-tuning. The number of diffusion time steps was set to 1000. The base model ResNet-18 was pretrained using a learning rate of $3e-4$ with a batch size of 128 till convergence using Adam optimizer with rand-augment also applied. Whereas for the NSL model, we directly used the pre-trained model from [10].

Experimental Configuration and Metrics: All the conducted experiments in this study are described in Table 3. For all our experiments related to gender classification, the training dataset used was the FairFace dataset, with rand-augment data augmentation. During inference, the second-order solvers are used for CDM models [36], whereas the proposed fast single-step solver 2 is used for CCFM. When

2. Our approach is classifier-agnostic, ensuring enhanced performance regardless of the base model. We specifically chose ResNet for its widespread use in related research and EfficientNetV2-L for its proven effectiveness in state-of-the-art bias mitigation studies.

training the CDM/ CCFM model in the classification setting (refer row #2, Table 3), the classifier prediction and feature representation are given as conditioning signals to the model during training. However, we did an ablation study with classifier prediction and classifier feature as the conditioning signal individually. We noted classifier prediction's superior role in enhancing the model's generalization and fairness. However, the optimum results are obtained by using both the classifier's prediction and features as conditioning signals (refer to Appendix B.2 and B.3 for further details). Therefore, for all our experiments, we used both of them as conditioning signals. Our models' performance was evaluated using several *key metrics*, including average accuracy, degree of bias, selection rate, max-min fairness, demographic parity, and equal opportunity difference, detailed in Appendix A.1. These metrics enabled us to quantify the performance and fairness of our models.

Using the datasets in Table 1, we strictly *follow* the same experimental set-up as in [9], [32], [35] for the fair cross-comparison of the results. We compare against other bias mitigation techniques, including oversampling, domain discriminative training [40], domain-independent models [32], an adversarial approach [33], regularization [34], Fair-Mixup [8], GAN-based offline dataset debiasing [35] and g-SMOTE [9].

In the following sections, we incrementally add proposed changes (see Table 3) and examine their performance from various fronts.

TABLE 2
Mapping of Experiment configurations to Architectures

Configuration	CDM/CCFM Architecture
CDM-VP	Base Classifier + VP [37] Diffusion
CDM-VE	Base Classifier + VE [41] Diffusion
CDM-EDM	Base Classifier + EDM [36] Diffusion
CCFM	Base Classifier + CFM [42]

We can generate various configurations for our proposed CDM and CCFM models by leveraging different pre-established diffusion formulations. Each resulting configuration constitutes a unique variant within the broader CDM/CCFM model family. Refer to Table 2 for examples of configurations employed in our experiments.

6.2 Evaluation on FairFace: Gender Classification

In this section, we examine the results of the model when a pretrained ResNet-18 model is trained with the configuration described above (also refer to row #1 Table 3). The results of which are shown in Table 4. We note that merely incorporating a diffusion head into a classifier maintains the generalization performance yet grants the base configuration the capability to estimate uncertainty. This is especially important since existing uncertainty methods when applied to a given model degrade both the accuracy as well as fairness [18]. However, our proposed method maintains the performance profile without affecting the performance.

6.3 Confidence Interval Estimation and Uncertainty Quantification for Test-time Rejection

To assess the uncertainty of model predictions (refer to row #1, Table 3), we incorporate the concept of instance-level model confidence. This refers to the degree of certainty the model has regarding each of its predictions, which is determined by the stochastic nature of outputs from a generative model. Our approach involves evaluating the instance-level confidence of model predictions by generating multiple class prototype reconstructions for each test instance using the proposed model. The class prediction intervals can then be estimated using confidence intervals. For all our experiments, we used 95% confidence interval. To implement this framework, the classifier needs to avoid generating identical outputs on every occasion. This is necessary to construct prediction intervals for each class label. Hence, utilizing generative models, which can produce stochastic outputs rather than solely relying on point estimates like traditional classifiers, is a preferred modeling choice.

We utilize a generative model to stochastically rebuild each one-hot label, which we treat as a class prototype within a continuous real space, as detailed in Section 5. The principle underpinning this approach is that the classifier's certainty about the class of a specific instance is reflected in the accuracy of the prototype reconstruction: the more confident the classifier, the more closely the recreated prototype vector matches the original with minimal uncertainty. Conversely, if the classifier is unsure about the class, the reconstructions of different class prototypes for the same test instance will display more variations. This phenomenon is particularly noticeable in denoising diffusion models, where samples drawn from the prior distribution at a given timestep T lead to distinct label reconstructions.

Figure 3 showcases the relationship between acceptance rate thresholds and the corresponding fairness and accuracy metrics. The fairness metrics (STD, EOD) exhibit a consistent pattern of improvement in fairness and generalization as the acceptance rates decrease. This indicates that the model has developed the capability to effectively identify uncertain samples. Depending on the specific requirements of the application, an appropriate acceptance threshold can be selected and used for rejecting samples during test time. For instance, the Figure shows that at an acceptance rate of 94%, the STD of the model improves from 2.26 to ~ 1.75, whereas Avg. Acc improves from 93.2% to $> 95\%$ for all our proposed configurations, a marked improvement.

6.4 Fine-tuning the Base Classifier

In this section, we evaluate the efficacy of fine-tuning the base classifier along with training the diffusion head (refer to row #2, Table 3). Upon examination, fine-tuning appears to improve the performance of all configurations significantly. Notably, Avg. Acc increases for all configurations upon fine-tuning, with the CDM-VE and CDM-VP configurations displaying the highest Avg. Acc values, respectively, an improvement from 93.2% to 94.04%. The increase in Average Accuracy indicates that the fine-tuning process has resulted in a better overall model fit.

Looking at the standard deviation (STD), it is observable that fine-tuning reduces variability across all demographic

TABLE 3
Overview of experiments conducted in this study. For each experiment, their objective and the task on which it is evaluated along the protected attribute under consideration are given, along with the datasets used. Model configuration describes the status of the base classifier and the diffusion head. For, e.g. Base (Frozen) + Diffusion (Trained) implies only the diffusion model is trained while the base classifier is frozen.

Experiment Objective	Task (Protected Attribute)	Evaluated Datasets	Model Configuration	Base Classifier
Uncertainty Estimation (Sec. 6.2, 6.3)	Gender Classification (Race)	FairFace	Base (Frozen) + Diffusion (Trained)	ResNet-18
Bias Mitigation (Sec. 6.4)	Gender Classification (Race)	FairFace	Base (Finetuned) + Diffusion (Trained)	ResNet-18
Ablation: Classifier Prediction (See Appendix B.2)	Gender Classification (Race)	FairFace	Base (Finetuned) + Diffusion (Trained)	ResNet-18
Ablation: Classifier Feature (See Appendix B.3)	Gender Classification (Race)	FairFace	Base (Finetuned) + Diffusion (Trained)	ResNet-18
Impact on existing large capacity bias mitigation method (See Appendix B.1)	Gender Classification (Race)	FairFace	Base (Frozen) + Diffusion (Trained)	NSL
Impact on existing large capacity bias mitigation method (Sec. 6.7)	Gender Classification (Race)	FairFace	Base (Finetuned-LORA) + Diffusion (Trained)	NSL
Comparison: Uncertainty Estimation Methods (Sec. 6.8)	Gender Classification (Race)	FairFace	Base (Finetuned-LORA) + Diffusion (Trained)	NSL
Comparison: Bias Mitigation (Sec. 6.8)	Facial Attribute Classification (Gender)	CelebA	Base (Finetuned) + Diffusion (Trained)	ResNet-18
Cross Dataset Evaluation (Sec. 6.10)	Gender Classification (Race)	DiveFace, Morph, UTKFace	Base (Finetuned-LORA) + Diffusion (Trained)	NSL
Evalation on other modalities (See Appendix B.4)	Gender Classification	VISOB, UFPR, Notredame, NDIris	Base (Finetuned) + Diffusion (Trained)	ResNet-18, NSL

TABLE 4
Gender classification results of the proposed method without finetuning of base classifier across demographic groups when trained and tested on FairFace. Results indicate no loss of generalization performance.

Config	Avg. Acc ↑	STD ↓	SeR ↑	DEP ↓	Min Grp Acc ↑	Max Grp Acc ↑
Base-ResNet-18 [39]	93.2	2.26	90.05	17.89	86.39	95.94
CDM-VP	93.33±0.04	2.22±0.05	90.28±0.28	18.46±0.10	86.95±0.21	96.31±0.08
CDM-VE	93.31±0.03	2.13±0.01	90.50±0.10	18.47±0.10	87.16±0.10	96.31±0.00
CDM-EDM	93.36±0.03	2.38±0.05	89.12±0.12	18.64±0.16	86.29±0.19	96.83±0.14
CCFM	93.32±0.04	2.35±0.08	89.49±0.33	18.38±0.09	86.34±0.31	96.48±0.13

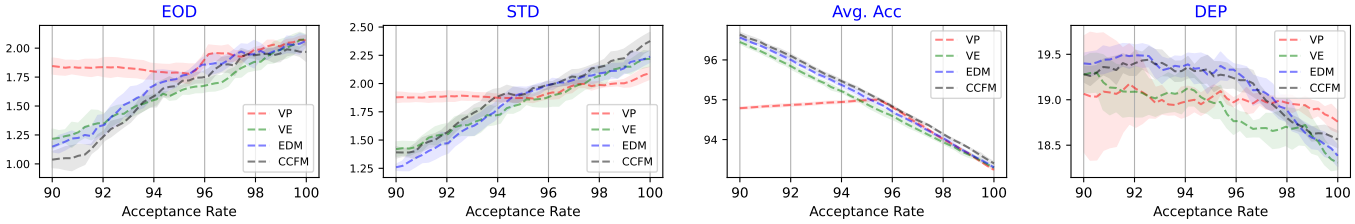


Fig. 3. Uncertainty curves obtained using the proposed method without finetuning of base classifier. The improved fairness (EOD, STD) and generalization performance (Avg. Acc) on various thresholds indicate the model's ability to detect uncertain samples for test-time rejection.

TABLE 5
Gender classification results of the proposed method across demographic groups when trained and tested on FairFace. The base classifier is also finetuned during training. Both the generalization performance as well as fairness improved substantially.

Config	Avg. Acc ↑	STD ↓	SeR ↑	DEP ↓	Min Grp Acc ↑	Max Grp Acc ↑
Base-ResNet-18	93.2	2.26	90.05	**17.89**	86.39	95.94
CDM-VP	94.03±0.04	**1.88±0.05**	92.61±0.21	18.84±0.09	89.83±0.22	97.00±0.06
CDM-VE	**94.04±0.01**	1.98±0.02	92.25±0.10	19.02±0.03	89.64±0.09	97.17±0.00
CDM-EDM	93.95±0.03	1.96±0.02	**92.63±0.10**	18.96±0.06	**89.94±0.09**	97.10±0.06
CCFM	93.98±0.02	2.02±0.01	91.89±0.10	18.89±0.06	89.70±0.08	**97.61±0.06**

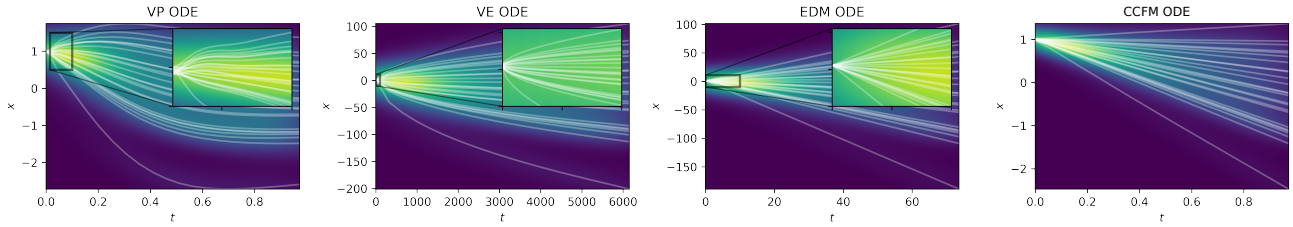


Fig. 4. **Trajectory Visualization** of the ODEs of various configurations (refer row #1, Table 3) during inference. On the time-axis on the right are randomly sampled points, and at $t = 0$, we obtain the final output prediction. The CDM-VP and CDM-VE trajectory shows large curvature, which only converges towards the end. Although CDM-EDM has a near linear trajectory, because of the high variance, it also only converges towards the end. Whereas CCFM has a near-linear trajectory throughout and has low variance.

groups, indicating that the results are more consistent and reliable following fine-tuning. The CDM-VP setup yields the lowest standard deviation, indicating it is the fairest configuration, reducing the STD from 2.26 to 1.88.

In terms of fairness, SeR also shows a positive trend with fine-tuning, meaning that the models are better at equally selecting the positive class across different demographic groups after fine-tuning, showing, on average, 2% improvement. However, the demographic parity (DEP), which measures the model's fairness, slightly increased in all configurations($+1\%$), indicating a small increase in discrimination.

The accuracy in the different groups, as indicated by Min Grp Acc ($+3.5\%$) and Max Grp Acc ($+1\%$), also shows an increase in performance. This suggests that fine-tuning improves overall performance and has positive effects across different groups in the data.

6.5 Trajectory Visualization

In this section, we demonstrate the Trajectory Visualization of the ODEs (Eq.1) of various configurations (Table 2) during inference. On the time-axis on the right are randomly sampled points, and at $t = 0$, we obtain the final output prediction.

The shape of the ODE solution trajectories (Figure 4) is defined by functions of various factors. The choice of these functions offers a way to reduce truncation errors during sampling. CDM-VP ODE solution trajectories flatten to horizontal lines at large σ, and gradients point towards data at small σ. The CDM-VE variant presents extreme curvature near data with curved solution trajectories. CDM-EDM ODE shows that as σ increases, solution trajectories become straight lines aiming at the data mean. In the case of CCFM, by definition, the trajectories remain straight lines throughout. It is through exploiting this linearity of trajectory, that we proposed the single-step solver to accelerate the sampling process of CCFM in section 4.4.

6.6 Effectiveness of Single Step Inference

In this section, we demonstrate the efficacy of our single-step solver to accelerate the sampling process of CCFM based on the trajectories visualization in section 6.5. Table 6 shows a comparison between various solvers used to solve the CCFM ODE. The results show that the single-step method, requiring only 1 NFE (Number of Function Evaluation), demonstrates competitive performance. Despite its low computational complexity, it obtains an average accuracy of 93.99%, with an insignificant decline compared to the ODE Solver (101 NFE) and Heun (36 NFE) methods, at 94.01% and 93.98% respectively. It also maintains a comparable standard deviation (STD), SeR, and DEP. The minimal group accuracy at 89.78% remains close to the maximum of 89.80% (ODE Solver) while achieving the maximum group accuracy of 97.66%, nearly identical to the ODE Solver. Therefore, our proposed Single Step solver presents a highly efficient inference method with its near-equivalent performance metrics while significantly reducing function evaluations.

6.7 Finetuning Larger Base Models with Low-Rank Adaptation

In our experiments, we initially utilized a smaller base model to manage the computational load of training a diffusion model with 1000 timesteps ($T = 1000$). Fine-tuning larger models, however, proved cost-prohibitive on lower-end machines, an issue we address in this section (see Table 3, row #6). To evaluate our approach's impact on an already state-of-the-art fairness method, we applied

TABLE 6
Effectiveness of Single Step Inference for CCFM. The number of function/network evaluations (NFE) is provided in the brackets. Both our proposed solvers are able to match the performance of an ODE solver [42] that takes significantly more steps in considerably fewer steps.

Config	Avg. Acc ↑	STD ↓	SeR ↑	DEP ↓	Min Grp Acc ↑	Max Grp Acc ↑
ODE Solver [42] (101 NFE)	94.01±0.02	2.04±0.01	91.93±0.06	18.83±0.06	89.80±0.07	97.68±0.02
Heun (Alg 1, 36 NFE)	93.98±0.02	2.02±0.01	91.89±0.10	18.89±0.06	89.70±0.08	97.61±0.06
Proposed Single Step (Alg 2, **1 NFE**)	93.99±0.01	2.04±0.02	91.92±0.07	19.05±0.02	89.78±0.06	97.66±0.00

TABLE 7
Results on SOTA NSL model with fine-tuning of the base model. Results indicate improved generalization and max-min fairness with finetuning (CDM-VP).

Config	Avg. Acc ↑	STD ↓	SeR ↑	DEP ↓	Min Grp Acc ↑	Max Grp Acc ↑
Base-NSL [10]	94.67	1.67	**93.78**	18.79	91.24	97.29
CDM-VP	**95.06**±0.02	1.70±0.03	93.48±0.07	19.15±0.07	**91.76±0.0**6	**98.15±0.00**
CDM-VE	94.96±0.02	**1.59±0.01**	93.16±0.07	18.73±0.02	91.10±0.06	97.79±0.00
CDM-EDM	94.87±0.02	1.60±0.02	92.72±0.13	**18.71±0.03**	90.78±0.13	97.91±0.00
CCFM	95.02±0.04	1.71±0.02	93.29±0.07	18.86±0.04	91.23±0.06	97.79±0.00

the NSL model from = [10], using a larger EfficientNetV2-L model compared to our initial ResNet-18. NSL, employing structured learning and generative models, creates *neighbor views* and uses *neighbor loss* for minimizing view distances, enhancing fairness on the FairFace dataset. Our objective is to demonstrate how our proposed method can enhance and extend the capabilities of an already SOTA bias mitigation model, especially when scaled to larger architectures. In experiments paralleling those in section 6.2, we added uncertainty estimation to NSL without sacrificing accuracy or fairness, a unique feature compared to conventional methods. Results are omitted here for brevity (see Appendix B.1), as they closely mirror those in section 6.2.

To fine-tune larger models with limited computational resources (refer row #6, Table 3), we employed the Low-Rank Adaptation method (LoRA) [43]. By freezing the pre-trained model weights, LoRA introduces trainable rank decomposition matrices into each model layer. This substantially lessens the count of trainable parameters for subsequent tasks. Initially proposed for transformers and large language models, LoRA has demonstrated comparable or superior results to traditional fine-tuning.

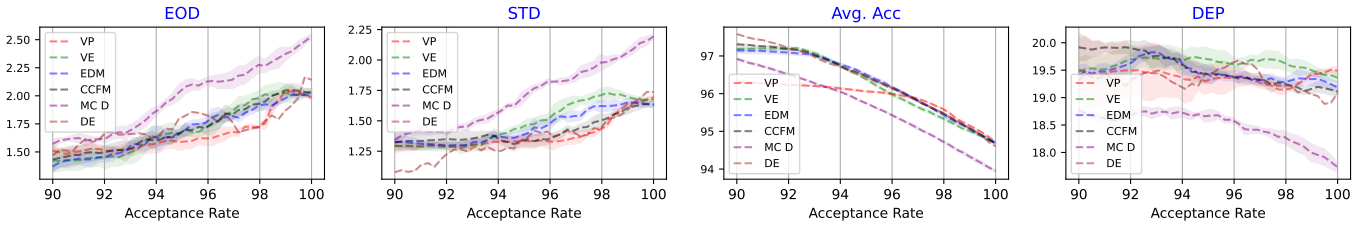
From the experiments, it was observed that finetuning with LoRA has shown improvements in Avg. Acc, and reductions in STD and DEP for most configurations, indicating enhanced fairness. The Avg. Acc of all finetuned models has increased or remained the same when compared to the base models. For instance, the CDM-VP model shows a notable increase from 94.71% to 95.06%. The STD, which represents model inconsistency across different data subsets, has mostly remained the same or slightly increased. Notably, the CDM-VE model shows a decrease in STD from 1.70 to 1.59, suggesting improved model consistency after fine-tuning. DEP, a measure of error disparity between groups, has generally decreased for the fine-tuned models, signifying a reduction in disparity after fine-tuning. Min Grp Acc and Max Grp Acc, the accuracy for the worst-off and best-off groups, respectively, have remained roughly the same or improved slightly for the fine-tuned models.

Looking specifically at the DEP, which is directly related to fairness, all finetuned models have shown a decrease in DEP compared to their base counterparts. The decrease in DEP signifies a reduction in prediction errors between different groups, indicating an improvement in fairness. Moreover, the Min Grp Acc of finetuned models has increased or remained the same, suggesting that the performance of the worst-off group hasn't been compromised. This is crucial for ensuring fairness, as it's important that the improvement in the model's performance doesn't come at the cost of a reduction in the accuracy of the disadvantaged groups.

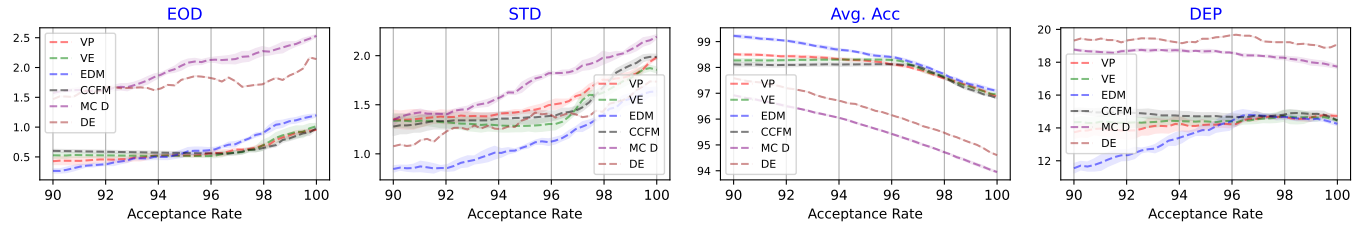
Overall, the low-rank adaptation method used for fine-tuning has been effective in improving fairness in the machine learning models, as seen by the decrease in DEP and an increase or stable performance in Avg. Acc, and improvement or unchanged Min Grp Acc. However, the changes in SeR and STD were minor, indicating the need for continued research into refining this finetuning process to achieve more consistent and equitable predictions across groups. This experiment highlights the potential of our method to not only boost the capabilities of existing state-of-the-art bias mitigation models but also hint at unexplored avenues for further advancements.

6.8 Comparison against existing uncertainty estimation methods for test-time rejection

In this section, we conducted a comparative analysis of our proposed uncertainty estimation method in comparison to two widely adopted methodologies: Monte Carlo Dropout [44] and Deep Ensembles [45] (refer row #7, Table 3) for test-time rejection of the samples. To ensure a fair comparison, we utilized the same base model (Base-NSL) as in our previous experiment and incorporated each of these three methodologies. For Deep Ensembles, we trained 100 distinct models to form a diverse ensemble capable of robust performance. Similarly, we applied the Monte Carlo



(a) Uncertainty Curves Without Finetuning



(b) Uncertainty Curves With Finetuning

Fig. 5. Comparison against existing uncertainty estimation methods for test-time rejection. MC-D: Monte Carlo Dropout, DE: Deep Ensembles. VP, VE, etc correspond to CDM-VP, CDM-VE, etc. model configurations respectively, with base classifier NSL model. (a). Without finetuning the base classifier, our proposed method reaches results equivalent to SOTA uncertainty estimation methods on various thresholds. (b) With finetuning, results improve substantially, showing much better fairness metrics over various thresholds as well as improvement in generalization performance.

Dropout method with 100 iterations to obtain comprehensive evaluations. Our proposed method was also evaluated 100 times to ensure fairness in comparison. The results were evaluated based on uncertainty estimation curves, as shown in Figure 5. In a test-time rejection setting, our method, without fine-tuning the base classifier, matches state-of-the-art uncertainty estimation methods across various acceptance thresholds and, when fine-tuned, demonstrates substantial improvements in generalization performance (Avg. Acc) as well as fairness (DEP).

The experimental data gathered indicates that our proposed method demonstrates comparable if not superior, performance when compared to existing methodologies. This observation is supported by the uncertainty estimation curves depicted in Figure 5. Furthermore, our method offers additional advantages beyond equivalent performance, including minor improvements in fairness and generalization performance on finetuning. For instance, the fine-tuned model, our best model (CDM-VP), has an improvement of Avg.Acc ($+0.3\%$), Min. Grp Acc ($+0.5\%$) and Max. Grp Acc ($+1\%$). These improvements have been achieved without compromising the ability to estimate uncertainty effectively, demonstrating the effectiveness of our fine-tuning efforts. For instance, with finetuning, at an acceptance rate of 94%, our models, on average, have Avg. Acc of 98% whereas existing methods at the same threshold have 96.5%, a $+1.5\%$ improvement. Thus, our proposed method presents a valuable proposition by balancing enhanced performance metrics with reliable uncertainty estimation.

6.9 Comparison against existing bias mitigation methods: Facial Attribute Classification

In this section, we conduct a comparative study between our proposed bias mitigation method and the established ones using the benchmark dataset CelebA for the task of multi-facial attribute classification (refer row #8, Table 3). For detailed information on established bias mitigation techniques, refer to Related work section 2. For the CelebA dataset, we adhere to the evaluation methodology as outlined in [9]. This method calculates the accuracies of 13 distinct attribute predictions, with gender serving as the protected attribute. To offer a robust comparison of fairness across different methods, we calculate and analyze the mean delta of minimum, maximum, and average accuracies.

The outcome of this comprehensive experiment is visually represented in Figure 6. To ensure a consistent comparison base, we opted for a ResNet architecture, given its widespread adoption among most of the existing methods in the literature. The findings derived from this analysis show that most of the existing bias mitigation approaches [8], [32]–[35] are *pareto inefficient*, also suggested in [9], and our proposed method *outperforms* most of the existing approaches. Further, our approach is capable of matching, if not surpassing, the performance of the current state-of-the-art on CelebA dataset, g-SMOTE method [9]. For instance, apart from g-SMOTE, all the other methods shown in Figure 6 show a negative change in Avg. Acc as well as Min and Max Grp Acc. g-SMOTE is the only other method that shows positive delta change. When compared to g-SMOTE, our best method (CDM-EDM) has a minor improvement of ($+0.2\%$) on Avg. and Min. Grp. Acc. However, g-SMOTE requires the additional overhead of training an image-generative model, such as GAN, to produce an augmented training dataset, which is computationally expensive compared to our proposed methods. Though the illustrated figure shows only CDM-EDM, similar performance is observed in other configurations, including CCFM.

The congruity between the two sets of results (gender classification, multi-attribute classification) underscores the versatility and generalizability of our approach. This

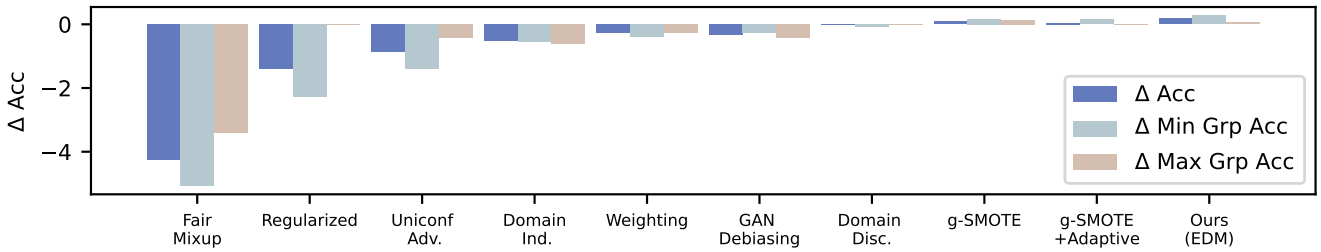
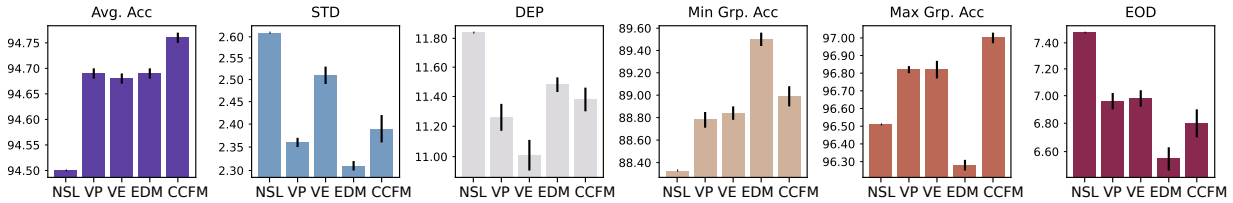
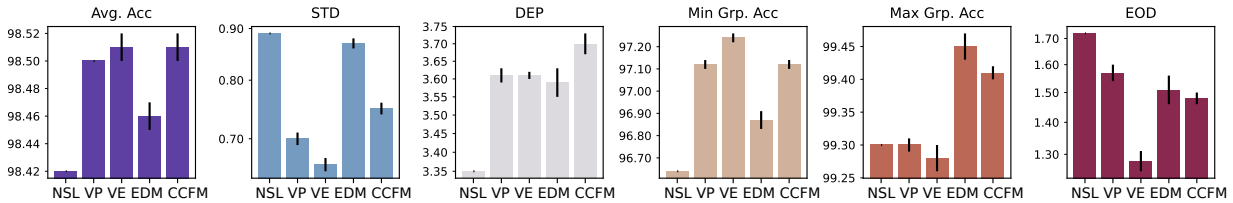


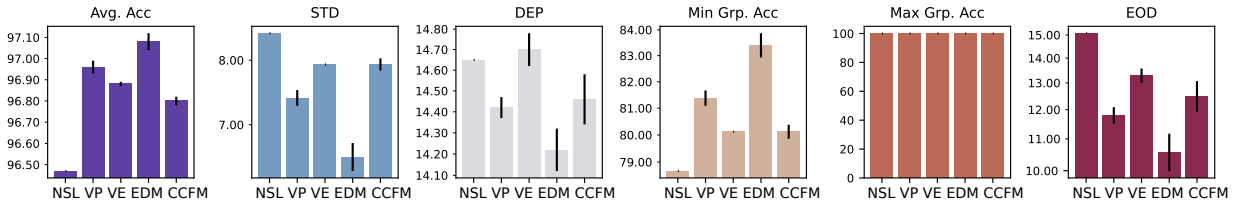
Fig. 6. Bias mitigation techniques on CelebA dataset. A decomposition of the change in accuracy into overall change, change in the best performing group, and the worst. Evaluated models are Fairmixup [8], Regularized [34], Uniconf. Adv [33], Domain Ind. [32], Weighting [32], GAN Debiasing [35], Domain Disc. [32], g-SMOTE [9]. Refer to section 2 for details on referenced bias mitigation techniques. Note that EDM refers to CDM-EDM. Variants like CCFM also show identical performance, so their results are excluded for brevity.



(a) UTKFace [21]



(b) DiveFace [22]



(c) Morph [23]

Fig. 7. Cross-dataset comparison of our proposed model with NSL [10]. The proposed model finetuned on NSL [10] (from section 6.7 is evaluated on UTKFace, Diveface, and Morph. VP, VE, etc. corresponds to CDM-VE, CDM-VP, etc., respectively (see Table 2.)

demonstrates the applicability of our proposed approach across different datasets and contexts, reinforcing its practical significance in the realm of bias mitigation. Thus, our proposed methodology not only shows equal or superior performance to the existing state-of-the-art methods but also offers promising adaptability across diverse demographic groups, tasks, and datasets.

6.10 Cross Dataset Evaluation

In this section, results of the cross-dataset evaluation (out-of-distribution) on UTKFace [21], DiveFace [22] and Morph [23] datasets are discussed (refer row #9, Table 3). Results are shown in Figure 7. We are using the trained model from section 6.7 and are evaluating them on these aforementioned datasets.

Our method is compared to a state-of-the-art (SOTA), called NSL [10], which utilizes generative views. NSL was chosen because of its SOTA performance on cross-dataset evaluation on the said datasets for the given task. Our findings indicate significant improvements over NSL in terms of both generalization performance and fairness across all the evaluated datasets. Specifically, min group accuracy and max group accuracy showed notable enhancements. For instance, on all 3 datasets, on average, our methods improved Avg. Acc (UTKFace-$+0.25\%$, DiveFace-$+0.10\%$, Morph-$+0.5\%$) and Min.Grp Acc (UTKFace-$+0.5\%$, DiveFace-$+0.50\%$, Morph-$+2\%$). Moreover, the equal opportunity

difference (EOD) metric, as well as the degree of bias (DoB), exhibited a decrease, suggesting an improvement in fairness. These outcomes highlight the effectiveness and fairness of our proposed method compared to the existing SOTA NSL approach. Finally, we also noted the effectiveness of our proposed models for bias mitigation on other biometric modalities (ocular and periocular). Readers are referred to Appendix B.4 for detailed information on bias mitigation of ocular and periocular biometric modalities using our proposed models.

7 Conclusion

In the pursuit of designing unbiased automated AI-based algorithms, addressing the pronounced demographic bias has been paramount. While the existing bias mitigation techniques present challenges in generalizability and often necessitate a trade-off between fairness and classification accuracy (Pareto efficiency), our proposed technique, which integrates diffusion and flow-matching models with a base classifier, showcases significant promise by delivering better Pareto efficiency than existing SOTA methods. Our choice of diffusion and flow-matching models was reinforced by their inherent capacity to effectively capture diverse data distributions and inherent stochasticity. The key takeaways from our research include: (1) By leveraging the robustness and stochastic nature of diffusion models, we not only improved classification accuracy but also introduced a capability for test-time rejection based on prediction uncertainty. (2) Our exhaustive evaluations across gender-annotated facial and ocular datasets (Appendix B.4) underscored the superiority of our method. Notably, our technique outperformed existing mitigation strategies in terms of both fairness and accuracy. (3) The approach consistently delivered robust performance across a gamut of tasks, from gender classification on the FairFace dataset to multifaceted attribute classifications on the CelebA dataset, and eliminated the inference bottleneck using advanced sampling strategies while remaining performant. The performance remained commendable even when tested on other modalities and in various data distribution scenarios. A key advantage of our method is that it doesn't necessitate the use of sensitive attribute annotations during the training process. It can function both as an in-processing technique improving generalization and as a post-processing method, offering the flexibility of test-time rejection.

Although our approach successfully enhances the fairness and generalizability of a pre-existing classifier, a notable limitation lies in the need to retrain an additional diffusion model. This adds a layer of complexity and resource requirements. Exploring ways to incorporate these techniques directly into the initial training phase of classifiers or developing zero-shot adaptation methods represents a promising direction for future research. This could streamline the process and potentially reduce the overhead associated with our current method.

In conclusion, this work offers a pivotal step forward in the development of unbiased face-based attribute classification algorithms. Our technique not only bridges the fairness-accuracy trade-off but also lays the groundwork for future research to further optimize and generalize bias mitigation for various computer vision algorithms. The consistent state-of-the-art results garnered in our evaluations underscore the potential and efficacy of our proposed method, advocating for its wider adoption in the realm of computer vision and beyond.

Acknowledgments

This work is supported by National Science Foundation (NSF) award no. 2129173. The research infrastructure used in this study is supported in part by grant no. 13106715 from the Defense University Research Instrumentation Program (DURIP) from the Air Force Office of Scientific Research.

References

[1] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Comput. Surv.*, vol. 54, no. 6, jul 2021.

[2] A. Krishnan, A. Almadan, and A. Rattani, "Understanding fairness of gender classification algorithms across gender-race groups," in *Proc. 19th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, 2020, pp. 1028–1035.

[3] V. Albiero, K. K. S, K. Vangara, K. Zhang, M. C. King, and K. W. Bowyer, "Analysis of gender inequality in face recognition accuracy," in *Proc. IEEE WACV Workshops 2020*, 2020, pp. 81–89.

[4] S. H. Abdurrahim, S. A. Samad, and A. B. Huddin, "Review on the effects of age, gender, and race demographics on automatic face recognition," *Vis. Comput.*, vol. 34, no. 11, pp. 1617–1630, 2018.

[5] A. Krishnan and A. Rattani, "A novel approach for bias mitigation of gender classification algorithms using consistency regularization," *Image Vis. Comput.*, vol. 137, p. 104793, 2023.

[6] P. Majumdar, R. Singh, and M. Vatsa, "Attention aware debiasing for unbiased model prediction," in *Proc. IEEE/CVF ICCVW 2021*, 2021, pp. 4116–4124.

[7] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating unwanted biases with adversarial learning," in *Proc. AAAI/ACM AIES 2018*. ACM, 2018, pp. 335–340.

[8] C. Chuang and Y. Mroueh, "Fair mixup: Fairness via interpolation," in *Proc. 9th Int. Conf. Learning Representations (ICLR) 2021*, 2021.

[9] D. Zietlow, M. Lohaus, G. Balakrishnan, M. Kleindessner, F. Locatello, B. Schölkopf, and C. Russell, "Leveling down in computer vision: Pareto inefficiencies in fair deep classifiers," in *Proc. IEEE/CVF CVPR 2022*, 2022, pp. 10 400–10 411.

[10] S. Ramachandran and A. Rattani, "Deep generative views to mitigate gender classification bias across gender-race groups," in *Proc. ICPR 2022 Int. Workshops and Challenges, Part III*, ser. Lecture Notes in Computer Science, vol. 13645. Springer, 2022, pp. 551–569.

[11] A. Das, A. Dantcheva, and F. Brémond, "Mitigating bias in gender, age and ethnicity classification: A multi-task convolution neural network approach," in *ECCV Workshops (1)*, ser. Lecture Notes in Computer Science, vol. 11129. Springer, 2018, pp. 573–585.

[12] X. Lin, S. Kim, and J. Joo, "Fairgrape: Fairness-aware gradient pruning method for face attribute classification," in *ECCV (13)*, ser. Lecture Notes in Computer Science, vol. 13673. Springer, 2022, pp. 414–432.

[13] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *ICML*, 2015, pp. 2256–2265.

[14] A. Hyvärinen, "Estimation of non-normalized statistical models by score matching," *J. Mach. Learn. Res.*, vol. 6, pp. 695–709, 2005.

[15] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural Comput.*, vol. 23, no. 7, pp. 1661–1674, 2011.

[16] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," in *Adv. Neural Inf. Process. Syst. 31 (NeurIPS 2018)*, 2018, pp. 6572–6583.

[17] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in *The Eleventh International Conference on Learning Representations*, 2023.

[18] X. Han, H. Zheng, and M. Zhou, "CARD: classification and regression diffusion models," in *NeurIPS*, 2022.

[19] K. Kärkkäinen and J. Joo, "Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation," in *WACV*, 2021, pp. 1547–1557.
[20] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. ICCV*, December 2015.
[21] Z. Zhang, Y. Song, , and H. Qi, "Age progression/regression by conditional adversarial autoencoder," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. IEEE, 2017.
[22] A. Morales, J. Fiérrez, R. Vera-Rodríguez, and R. Tolosana, "Sensitivenets: Learning agnostic representations with application to face images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 6, pp. 2158–2164, 2021.
[23] K. Ricanek and T. Tesafaye, "Morph: a longitudinal image database of normal adult age-progression," in *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, 2006, pp. 341–345.
[24] H. M. Nguyen, N. Reddy, A. Rattani, and R. Derakhshani, "VISOB 2.0 - the second international competition on mobile ocular biometric recognition," in *Proc. ICPR Int. Workshops and Challenges 2021, Part VIII*, ser. Lecture Notes in Computer Science, vol. 12668. Springer, 2020, pp. 200–208.
[25] L. A. Zanlorensi, R. Laroca, D. R. Lucio, L. R. Santos, A. S. Britto Jr., and D. Menotti, "A new periocular dataset collected by mobile devices in unconstrained scenarios," *Scientific Reports*, vol. 12, p. 17989, 2022.
[26] J. S. Bernhard, J. R. Barr, K. W. Bowyer, and P. J. Flynn, "Near-ir to visible light face matching: Effectiveness of pre-processing options for commercial matchers," in *BTAS*, 2015, pp. 1–8.
[27] K. W. Bowyer and P. J. Flynn, "The ND-IRIS-0405 iris image dataset," *CoRR*, vol. abs/1606.04853, 2016.
[28] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *FAT*, ser. Proceedings of Machine Learning Research, vol. 81. PMLR, 2018, pp. 77–91.
[29] V. Muthukumar, "Color-theoretic experiments to understand unequal gender classification accuracy from face images," in *Proc. IEEE CVPR Workshops 2019*, 2019, pp. 2286–2295.
[30] P. Barlas, K. Kyriakou, O. Guest, S. Kleanthous, and J. Otterbacher, "To "see" is to stereotype: Image tagging algorithms, gender recognition, and the accuracy-fairness trade-off," *Proc. ACM Hum. Comput. Interact.*, vol. 4, no. CSCW3, pp. 1–31, 2020.
[31] K. K. Teru and A. Chakraborty, "Towards reducing bias in gender classification," *CoRR*, vol. abs/1911.08556, 2019.
[32] Z. Wang, K. Qinami, I. C. Karakozis, K. Genova, P. Nair, K. Hata, and O. Russakovsky, "Towards fairness in visual recognition: Effective strategies for bias mitigation," in *Proc. IEEE/CVF CVPR 2020*, 2020, pp. 8916–8925.
[33] M. S. Alvi, A. Zisserman, and C. Nellåker, "Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings," in *Comp. Vis. - ECCV 2018 Workshops, Proc., Part I*, ser. Lecture Notes in Computer Science, vol. 11129. Springer, 2018, pp. 556–572.
[34] M. L. Wick, S. Panda, and J. Tristan, "Unlocking fairness: a trade-off revisited," in *Adv. Neural Inf. Process. Syst. 32 (NeurIPS 2019)*, 2019, pp. 8780–8789.
[35] V. V. Ramaswamy, S. S. Y. Kim, and O. Russakovsky, "Fair attribute classification through latent space de-biasing," in *CVPR*, 2021, pp. 9301–9310.
[36] T. Karras, M. Aittala, T. Aila, and S. Laine, "Elucidating the design space of diffusion-based generative models," in *NeurIPS*, 2022.
[37] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *NeurIPS*, vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851.
[38] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *NeurIPS*, 2019, pp. 11 895–11 907.
[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR 2016*, 2016, pp. 770–778.
[40] M. Saerens, P. Latinne, and C. Decaestecker, "Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure," *Neural Comput.*, vol. 14, no. 1, pp. 21–41, 2002.
[41] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *ICLR*, 2021.
[42] A. Tong, N. Malkin, G. Huguet, Y. Zhang, J. Rector-Brooks, K. Fatras, G. Wolf, and Y. Bengio, "Conditional flow matching: Simulation-free dynamic optimal transport," *CoRR*, vol. abs/2302.00482, 2023.
[43] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *ICLR*, 2022.
[44] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proc. 33rd Int. Conf. Mach. Learn. (ICML) 2016*, ser. JMLR Workshop and Conference Proceedings, vol. 48. JMLR.org, 2016, pp. 1050–1059.
[45] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Adv. Neural Inf. Process. Syst. 30 (NeurIPS 2017)*, 2017, pp. 6402–6413.

Sreeraj Ramachandran received a bachelor's and master's degree in computer engineering from the Indian Institute of Information Technology, Design & Manufacturing, Kancheepuram, India, in 2019. He is currently working toward a PhD degree from the School of Computing of Wichita State University, USA. His research interests include computer vision, biometrics, and generative models.

Ajita Rattani is an Assistant Professor and Director of the Visual Computing and Biometric Security Lab at the Dept. of Computer Science and Engineering at the University of North Texas at Denton, USA. She did her PhD and Post-doctoral studies at the Univ. of Cagliari, Italy, and Michigan State University, USA, respectively. Her research interests include computer vision, image analysis, deep learning, machine learning, fairness in AI, and biometrics. She is the principal investigator for various federal research grants from NSF and DOD. She has been the recipient of the Best Paper Awards from IEEE IJCB 2014, IEEE HST 2017, 2019. She has received Best Reviewer awards from Elsevier Journal on IACV 2018 and IEEE IJCB 2021. She has been serving as an NSF Panelist since 2020. She is the IAPR TC4 Education Subcommittee Chair and IEEE ISATC Task Force Chair on Biometrics.

Supplementary Materials

APPENDIX A
MATHEMATICAL BACKGROUND

This section provides a foundational mathematical background, divided into three key subsections. The first, section A.1 provides an overview of the concepts and principles pertinent to fairness in machine learning, including the fairness metrics used to evaluate the proposed models. The second subsection, A.2 introduces the core tenets of diffusion in the context of generative models, which are essential for understanding the mechanics of the discussed classification diffusion models. Finally, A.3 delves into the basic concepts behind flow matching, serving as a precursor to the subsequent discussion of our proposed Classification Flow Matching Models.

A.1 Fairness Preliminaries

Our exploration focuses primarily on the principles of fairness, specifically aiming to obtain a balanced distribution of classifier errors across diverse population subgroups. This can be accomplished by equalizing error rates across different demographics, such as gender and racial groups discussed comprehensively in [1]. The *equal opportunity difference* (EOD) metric is widely used in fairness evaluation and requires that a classifier exhibit the same true positive rates (TPR) for different subgroups [2] to be deemed fair. Whereas *degree of bias* [3] is obtained by calculating the standard deviation of individual subgroup utilities (STD). *Selection rate* (SeR) [4] is another fairness metric that calculates the ratio of the minimum utility group (least performing) to the maximum utility group (best performing). *Demographic parity* (DEP) is a fairness metric whose goal is to ensure a machine learning model's predictions are independent of membership in a sensitive group.

Another notion of fairness that has garnered significant attention is max-min fairness. In max-min fairness, we follow the Rawlsian principle of Max-Min welfare for distributive justice [5]. Unlike EOD and other group fairness metrics, max-min fairness aims to minimize the classification error for the worst-performing subgroup to the greatest extent possible. To quantify max-min fairness, we evaluate the minimum group accuracy, which is defined as the 1-error rate of the group with the minimum accuracy. A hypothesis h is said to satisfy Rawlsian Max-Min fairness principle [5], [6] if it maximizes the utility of the worst-off group, i.e., the group with the lowest utility.

A.2 Diffusion Preliminaries

In this section, we lay out the foundational theory of diffusion models, which is crucial for comprehending the working of Classification Diffusion Models (CDM). We begin with the denoising diffusion model, the most basic form, and then proceed to discuss score matching, a methodology for training these models.

A.2.1 Denoising Diffusion Model (DDPM)

Consider a data distribution denoted by $p_{data}(x)$ with a standard deviation σ_{data}. Adding i.i.d. Gaussian noise of standard deviation σ to the data may obtain a family of mollified distributions $p(\mathbf{x};\sigma)$. This distribution approaches a known prior distribution $\mathcal{N}(\mathbf{0},\sigma_T^2\mathbf{I})$ for $\sigma_{max} \gg \sigma_{data}$. Diffusion models operate by starting with a randomly sampled noise image $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0},\sigma_{max}^2\mathbf{I})$.

The goal is to progressively remove the noise from the image, generating a sequence of denoised images $\mathbf{x}_i$ with decreasing noise levels $\sigma_0 = \sigma_{max} > \sigma_1 > \cdots > \sigma_N = 0$, such that each image $\mathbf{x}_i \sim p(\mathbf{x}_i;\sigma_i)$. Eventually, the final image $\mathbf{x}_N$ obtained after the denoising process represents a sample that follows the data distribution.

A.2.2 Score Matching

In continuous-time diffusion models, denoising diffusion is extended to infinite steps via stochastic differential equations (SDEs). In the continuous-time setting of diffusion models as per [7], the SDE

$$d\mathbf{x} = \mathbf{f}(\mathbf{x},t)\,dt + g(t)\,d\omega_t, \tag{1}$$

guides sample $\mathbf{x}$ to adhere to distribution p. Here, ω_t is a standard Wiener process, and $\mathbf{f}(\cdot,t) : \mathbb{R}^d \to \mathbb{R}^d$ and $g(\cdot) : \mathbb{R} \to \mathbb{R}$ serve as the drift and diffusion coefficients, respectively, in a d-dimensional dataset. The Variance Preserving (VP) [8] and Variance Exploding (VE) [9] formulations are specific choices of these coefficients. VP stabilizes variance in the diffusion process, whereas VE allows it to increase, each with its own trade-offs in model performance. Specifically, $\mathbf{f}(\mathbf{x},t) = f(t)\mathbf{x}$, with $f : \mathbb{R} \to \mathbb{R}$, can be rewritten as

$$d\mathbf{x} = f(t)\mathbf{x}\,dt + g(t)\,d\omega_t, \tag{2}$$

This density evolution process can be reversed with a reversed SDE [10],

$$d\mathbf{x} = [f(t)\mathbf{x} - \frac{1}{2}g(t)^2\nabla_x \log p_t(\mathbf{x})]\,dt + g(t)\,d\bar{\omega}_t \tag{3}$$

where $\nabla_x \log p_t$ is the score function and $\bar{\omega_t}$ is the time-reversed Brownian motion. For a given noise level, the score function is a vector field that points towards higher data density. In other words, a tiny forward step of this differential equation moves the sample away from the data distribution at a rate that depends on how much the noise

level changes (forward SDE, Eq. 2), whereas a tiny backward step moves the sample closer to the data distribution (reversed SDE, Eq. 3). The reverse process can be approximated by learning the score function using a weighted *denoising score matching loss* [11] given by

$$\mathbb{E}_{t,\mathbf{x}_0\sim p_0,\mathbf{x}_t\sim p_t(\cdot|\mathbf{x}_0)}\lambda_t \left\|\mathbf{s}_\theta(\mathbf{x}_t,t)-\nabla_x \log p_t(\mathbf{x}_t|\mathbf{x}_0)\right\|^2 \quad (4)$$

With a learned score $\mathbf{s}_\theta(\mathbf{x},t) \approx \nabla_x \log p_t$, one can obtain a generative model by first sampling $\mathbf{x}_T \sim \mathcal{N}(0,\sigma_T^2 I)$ and then solving the reverse SDE by replacing the score function with the neural network approximation.

A.2.3 Probability Flow ODE

Deterministic ordinary differential equations (ODEs) converge faster compared to SDEs, making it easier to train and infer. They are also more stable during training and inference and have interpretable trajectories. For all the diffusion processes, there exists a corresponding deterministic process whose trajectories share the same marginal probability densities as the SDE. This deterministic process satisfies the ODE

$$d\mathbf{x} = \left[\mathbf{f}(\mathbf{x},t) - \frac{1}{2}g(t)^2\nabla_{\mathbf{x}}\log p_t(\mathbf{x})\right]dt \quad (5)$$

A.3 Flow Matching Preliminaries

Flow Matching (FM), as introduced by [12], presents a novel, simulation-free method to train Continuous Normalizing Flows (CNFs) [13] which represents a distinct category of generative models. The FM technique focuses on regressing vector fields along specific conditional probability trajectories, such as Gaussian and diffusion paths (trajectories). This offers a more stable and efficient substitute compared to the conventional diffusion and score-matching models. In this section, we lay the foundational theory of flow-matching models, which is crucial for comprehending the workings of our proposed CCFM. We begin with the continuous normalizing flows, a class of generative models, and then proceed to discuss flow-matching, a methodology to train these models efficiently.

A.3.1 Continuous Normalizing Flows (CNF)

Consider an input data space defined in $\mathbb{R}^d$ with densities $q(x_0)$ and $q(x_1)$ at times $t=0$ and $t=1$ respectively defined over $\mathcal{X} \subseteq \mathbb{R}^d$. We consider a *probability density path* $p:[0,1]\times\mathbb{R}^d \to \mathbb{R}^+$, which is a time-dependent probability density function, i.e., $\int p_t(x)\,dx = 1$ and *time-dependent vector field*, $v:[0,1]\times\mathbb{R}^d \to \mathbb{R}^d$. This vector field defines a unique time dependent flow $\phi:[0,1]\times\mathbb{R}^d\to\mathbb{R}^d$ defined by an ordinary differential equation (ODE):

$$\frac{d\phi_t(x)}{dt} = v_t(\phi_t(x)); \qquad \phi_0(x) = x \quad (6)$$

A push-forward $\phi_* : [0,1]\times\mathcal{P}(\mathbb{R}^d) \to \mathcal{P}(\mathbb{R}^d)$ pushes a density forward over time by

$$[\phi_t]_* p_0(x) = p_0(\phi^{-1}(x))|\det \nabla_x \phi_t^{-1}(x)| \quad (7)$$

A vector field v_t is said to generate a probability density path p_t if its flow ϕ_t satisfies equation 7. A neural network $v_\theta(t,x)$ may be used to model the vector field v_t, making it into a deep parametric model of the flow ϕ_t, called a *Continuous Normalizing Flow* (CNF) [13].

A.3.2 Flow Matching

Consider a random variable denoted as x_1, which follows an unknown data distribution $p_{data}(x_1) = q(x_1)$ and a probability path denoted as p_t, where p_0 corresponds to a simple distribution, such as the standard normal distribution $p(x) = \mathcal{N}(x|0,I)$. The distribution p_1 is chosen to be approximately equivalent to q, the data density. The objective of the flow matching approach is to align with this target probability path, enabling a smooth transition from p_0 to p_1. Given a target probability density path p_t and a corresponding vector field u_t that generates it, we may learn the flow that matches p_t through gradient descent on regression against the target vector field. The so-called flow matching objective [12] learns a time-dependent parameterized vector field $v_\theta(t,x):[0,1]\times\mathbb{R}^d\to\mathbb{R}^d$ by regression against some target vector field $u_t(x):[0,1]\times\mathbb{R}^d\to\mathbb{R}^d$ by minimizing a flow matching (FM) loss

$$\mathcal{L}_{FM}(\theta) = \mathbb{E}_{t,p_t(x)}\left\|v_\theta(t,x)-u_t(x)\right\|_2^2, \quad (8)$$

where θ denotes the learnable parameters of the CNF vector field v_t, $t\sim\mathcal{U}[0,1]$ and $x\sim p_t(x)$.

A.3.3 Conditional Flow Matching

Flow matching on its own is intractable since we have no prior knowledge of an appropriate p_t or u_t. But by using a mixture of simpler flows, say a mixture of conditional probability paths, we may get around this. Specifically, for any condition z independent of x and t, we may obtain our marginal probability density path and marginal vector field by

$$p_t(x) = \int p_t(x|z)q(x)\,dz, \quad (9)$$

$$u_t(x) = \mathbb{E}_{q(z)}\frac{u_t(x|z)p_t(x|z)}{p_t(x)} \quad (10)$$

where $u_t(x|z):\mathbb{R}^d\to\mathbb{R}^d$ is a conditional vector field that generates $p_t(x|z)$ from $p_0(x|z)$ and also generates p_t. The corresponding conditional flow matching (CFM) objective is equivalent to the FM objective under some mild conditions and is given by:

$$\mathcal{L}_{CFM}(\theta) = \mathbb{E}_{t,q(z),p_t(x|z)}\left\|v_\theta(t,x)-u_t(x|z)\right\|_2^2 \quad (11)$$

Since the flow is parametrized by an ODE, we can use efficient ODE solvers at inference time to generate samples.

APPENDIX B
ADDITIONAL EVALUATION

B.1 Training Larger Base Models: Effect on existing Bias Mitigation Method

In our initial experiments detailed in the main paper (refer to row #6, Table 3), we started with a smaller base model due to the demanding computational resources required for training a diffusion model with a high number of timesteps ($T = 1000$). The main challenge we faced was the high computational cost of finetuning larger base models on machines with limited resources. To evaluate the efficacy of our proposed approach on a state-of-the-art fairness technique, we integrated the model described in [14] with diffusion head. Unlike our previous experiments using the

TABLE 1
Training Larger Base Models: Effect on existing SOTA Bias Mitigation Method named NSL proposed by [14]. Results indicate an insignificant change in generalization and fairness performance, implying that the proposed method maintains accuracy and fairness while still providing the ability to estimate uncertainty.

Config	Avg. Acc ↑	STD ↓	SeR ↑	DEP ↓	Min Grp Acc ↑	Max Grp Acc ↑
Base-NSL [14]	94.67	1.67	**93.78**	**18.79**	91.24	97.29
CDM-VP	94.71±0.04	1.68±0.02	93.33±0.09	19.36±0.09	91.33±0.06	97.86±0.06
CDM-VE	**94.71±0.02**	1.70±0.02	93.34±0.12	19.49±0.08	91.36±0.06	**97.88±0.09**
CDM-EDM	94.67±0.01	**1.63±0.02**	93.38±0.12	19.17±0.08	91.36±0.06	97.84±0.06
CCFM	94.68±0.01	1.69±0.01	93.43±0.00	19.00±0.00	**91.36±0.00**	97.79±0.00

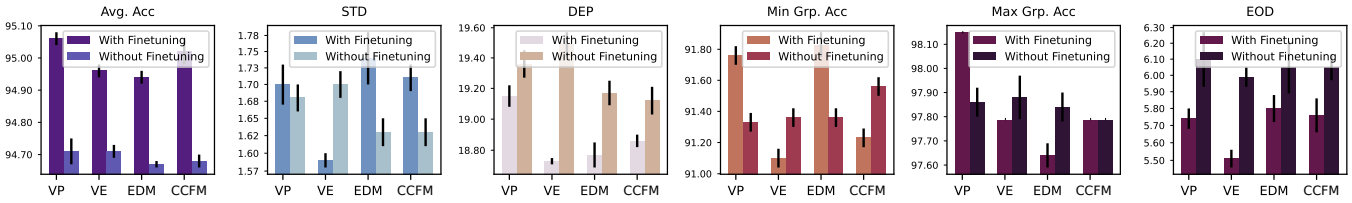


Fig. 1. Results on NSL model with and without fine-tuning the base model along with the diffusion head (CDM/CCFM). Results indicate improved generalization and max-min fairness with finetuning.

smaller ResNet-18 architecture, the method in [14] employs a larger EfficientNetV2-L model. Named NSL, this technique incorporates structured learning and uses generative models to create *neighbor views* for training. A regularization term, *neighbor loss*, minimizes the distance between these views, leading to exceptional fairness performance on the FairFace dataset. Our objective is to demonstrate how our proposed method can enhance and extend the capabilities of an already SOTA bias mitigation model, especially when scaled to larger architectures. We may completely avoid the computational overhead of the large model during diffusion training if the base model is not finetuned and kept frozen.

Table1 presents the results of the models trained without finetuning. We observe no significant improvements in performance metrics, except for a slight increase in the minimum group accuracy. However, it is to be noted that without altering the overall performance of the model, we have now added uncertainty estimation capability to the existing model. This aspect is particularly crucial, as current uncertainty methods [15], [16], when implemented in a specific model, tend to reduce generalization performance, as noted in [17]. Conversely, our proposed method preserves the performance profile without impacting overall performance. The uncertainty curves of which are illustrated and explained in Figure 5 in the main paper. These findings suggest that finetuning the base model may be necessary to achieve improved performance. For this purpose, we employed low-rank adaptation, which has minimal computational demands. This is discussed in section 6.8 of the main paper, and a comparison of their performance is illustrated in Figure 1.

B.2 Conditioning Signal: Classifier Prediction

When training the diffusion model in the classification setting (refer to row #2, Table 3 in the main paper), the classifier prediction and feature representation are given as conditioning signals to the model during training. In this section, we examine the impact of the classifier prediction by removing it as a conditioning signal. Without classifier prediction, the only conditioning signal will be the classifier features. The results of which are described in Figure 2. We observe that using the classifier prediction as conditioning signals significantly improves the overall performance of the model both in terms of generalization and fairness from Figure 2. For example, we observe an average improvement of $+0.8\%$ in Avg. Accuracy and $+3\%$ improvement in Min. Grp. Acc producing a more equitable model. This indicates that the diffusion model is essentially refining the output produced by the classification model. This underscores the value of integrating conditioning signals from classifier predictions into the model, thereby ensuring more reliable and fair results.

B.3 Conditioning Signal: Classifier Feature

In this section, we examine the impact of the classifier feature representation as a conditioning signal (refer to row #4, Table 3 in the main paper), the results of which are described in Figure 3. We observe that using the classifier feature as a conditioning signal significantly boosts both the model's generalization ability and its fairness. For, e.g., the Avg. Acc improves by $+0.8\%$ on average, and STD on average decreases by 0.25. However, when compared to the impact of using the classifier prediction as a conditioning signal in Figure 2, the effect of the classifier feature representation seems less pronounced. This indicates the classifier *prediction's superior role* in enhancing the model's generalization and fairness. However, the optimum results are obtained by using both the classifier's prediction and features as conditioning signals.

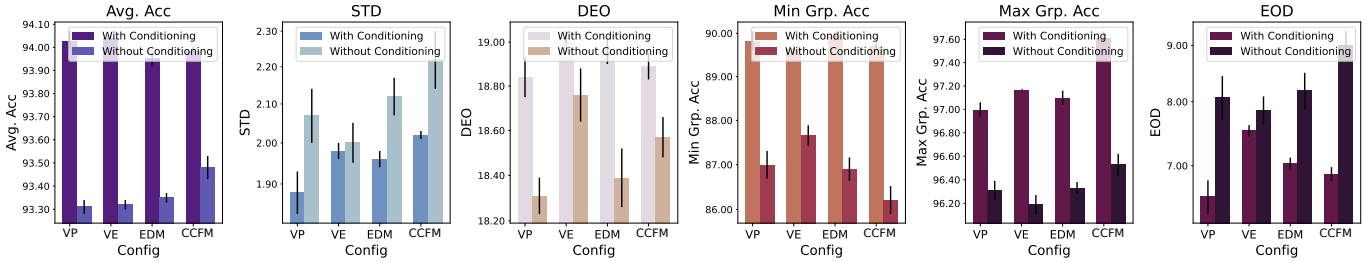


Fig. 2. Ablation study to determine the impact of classifier predictions as conditioning signals. Results indicate that removing classifier prediction significantly deteriorates the performance. Note that VP, VE, etc. refers to CDM-VP, CDM-VE, etc. (Refer to Table 2 in the main paper).

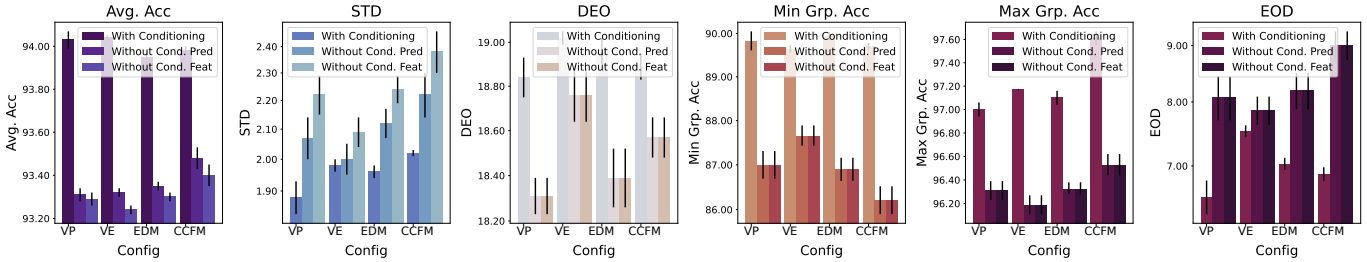


Fig. 3. Ablation study to determine the impact of classifier features as conditioning signals. Results indicate that removing the classifier feature significantly deteriorates the performance, similar to classifier predictions. Note that VP, VE, etc. refers to CDM-VP, CDM-VE, etc. (Refer to Table 2 in the main paper).

TABLE 2
Datasets used for training and evaluation on ocular and periocular modalities. Note that the demographic groups mentioned are the composition of the dataset, not the annotated demographic information. All the datasets are annotated with gender information.

Dataset	Images	Demographic Groups
UFPR [18]	$33K$	Brazilian
VISOB [19]	$150K$	White, Indian, Asian, Black
Notredame [20]	$22K$	White, Asian
NDIris [21]	$64K$	Caucasian, Asian

TABLE 3
Effectiveness of Our Method on ocular and periocular modalities. We illustrate the results from our optimal configuration; other configurations obtained similar results.

Dataset	Config	Avg. Acc ↑	Min Grp Acc ↑	Max Grp Acc ↑
UFPR [18]	Base-ResNet-18	**94.57**	94.21	**95**
	CDM-VE	94.51	**94.49**	94.54
Notredame [20]	Base-ResNet-18	93.82	90.81	96.03
	CDM-VE	**95.19**	**93.51**	**96.43**
NDIris [21]	Base-ResNet-18	83.3	74.38	**98.08**
	CDM-VE	**90.31**	**89.14**	92.26

B.4 Effectiveness of Our Method on other modalities

In an effort to evaluate the broad applicability of our proposed method, we extended our experiments to other biometric modalities, specifically focusing on ocular and periocular biometric modalities. Along with Notredame [20] and NDIris [21] dataset for ocular analysis on the NIR spectrum, we also used UFPR Periocular dataset [18] for periocular analysis on RGB spectrum with ResNet-18 model as the base classifier (in CDM/ CCFM setting). The demographic information of the dataset is provided in Table 2. The experiments are conducted in an intra-dataset setting. However, these datasets do not provide additional image-level demographic annotations other than gender attributes that could be used as a sensitive attribute for analysis, thereby limiting our evaluation across gender attributes alone. Despite this limitation, our method demonstrated its robustness and versatility. As summarized in Table 3, we observed marked improvements in accuracy across genders in both ocular and periocular biometrics on the same dataset evaluation (i.e., trained and tested on the same dataset). For instance, we have a $+3\%$ improvement of Min Grp Accuracy on the Notredame dataset, whereas a $+15\%$ improvement on the NDIris dataset when compared to the baseline model. The table displays CDM-VE results, but similar results were achieved with other configurations.

To analyze the generalizability of our approach, we also used VISOB dataset [19] with the NSL [14] model as the base classifier (in CDM/CCFM setting) for ocular-based gender classification with gender as the protected attribute. Note that the VISOB dataset is annotated with gender information. Our primary comparison is with the NSL model used in [14] for bias mitigation of ocular biometrics. This model represents the current state-of-the-art and is the only one

TABLE 4
Effectiveness of Our Method on VISOB [19] dataset. We illustrate the results from our optimal configuration; other configurations obtained similar results.

Config	Avg. Acc ↑	STD ↓	SeR ↑
NSL [14]	**89.17**	14.18	52.51
CDM-VE	84.7	**7.05**	**77.27**

to conduct similar experiments on the VISOB dataset (in an intra-dataset setting), allowing for a more equitable comparison. The experimental results, as tabulated in Table 4, offer strong evidence that our method not only improved the accuracies of the low-performing demographic groups as indicated by the significantly improved selection rate ($+25\%$) but also significantly diminished the level of bias inherent in the model predictions. For instance, the STD reduced significantly (-7%). The table displays CDM-VE results, but similar results were achieved with other configurations.

These results across diverse biometric modalities further underscore the effectiveness of our proposed bias mitigation method. Regardless of the specific type of biometric data and the dataset employed, our method consistently obtained enhanced accuracy and reduced bias. These outcomes, combined with our prior results, confirm the broad applicability, generalizability, and robustness of our method across varied contexts and data types.

REFERENCES

[1] S. Verma and J. Rubin, "Fairness definitions explained," in *Proceedings of the International Workshop on Software Fairness, FairWare@ICSE 2018, Gothenburg, Sweden, May 29, 2018*. ACM, 2018, pp. 1–7.

[2] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 2016, pp. 3315–3323.

[3] S. Gong, X. Liu, and A. K. Jain, "Debface: De-biasing face recognition," *CoRR*, vol. abs/1911.08080, 2019.

[4] F. Lin, Y. Wu, Y. Zhuang, X. Long, and W. Xu, "Human gender classification: a review," *Int. J. Biom.*, vol. 8, no. 3/4, pp. 275–300, 2016.

[5] J. Rawls, *Justice as fairness: A restatement*. Harvard University Press, 2001.

[6] P. Lahoti, A. Beutel, J. Chen, K. Lee, F. Prost, N. Thain, X. Wang, and E. H. Chi, "Fairness without demographics through adversarially reweighted learning," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[7] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *ICLR*, 2021.

[8] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *NeurIPS*, vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851.

[9] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *NeurIPS*, 2019, pp. 11 895–11 907.

[10] B. D. Anderson, "Reverse-time diffusion equation models," *Stochastic Processes and their Applications*, vol. 12, no. 3, pp. 313–326, 1982.

[11] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural Comput.*, vol. 23, no. 7, pp. 1661–1674, 2011.

[12] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in *The Eleventh International Conference on Learning Representations*, 2023.

[13] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," in *Adv. Neural Inf. Process. Syst. 31 (NeurIPS 2018)*, 2018, pp. 6572–6583.

[14] S. Ramachandran and A. Rattani, "Deep generative views to mitigate gender classification bias across gender-race groups," in *Proc. ICPR 2022 Int. Workshops and Challenges, Part III*, ser. Lecture Notes in Computer Science, vol. 13645. Springer, 2022, pp. 551–569.

[15] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proc. 33rd Int. Conf. Mach. Learn. (ICML) 2016*, ser. JMLR Workshop and Conference Proceedings, vol. 48. JMLR.org, 2016, pp. 1050–1059.

[16] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Adv. Neural Inf. Process. Syst. 30 (NeurIPS 2017)*, 2017, pp. 6402–6413.

[17] X. Han, H. Zheng, and M. Zhou, "CARD: classification and regression diffusion models," in *NeurIPS*, 2022.

[18] L. A. Zanlorensi, R. Laroca, D. R. Lucio, L. R. Santos, A. S. Britto Jr., and D. Menotti, "A new periocular dataset collected by mobile devices in unconstrained scenarios," *Scientific Reports*, vol. 12, p. 17989, 2022.

[19] H. M. Nguyen, N. Reddy, A. Rattani, and R. Derakhshani, "VISOB 2.0 - the second international competition on mobile ocular biometric recognition," in *Proc. ICPR Int. Workshops and Challenges 2021, Part VIII*, ser. Lecture Notes in Computer Science, vol. 12668. Springer, 2020, pp. 200–208.

[20] J. S. Bernhard, J. R. Barr, K. W. Bowyer, and P. J. Flynn, "Near-ir to visible light face matching: Effectiveness of pre-processing options for commercial matchers," in *BTAS*, 2015, pp. 1–8.

[21] K. W. Bowyer and P. J. Flynn, "The ND-IRIS-0405 iris image dataset," *CoRR*, vol. abs/1606.04853, 2016.