Harnessing Diffusion-Generated Synthetic Images for Fair Image Classification

Anonymous CVPR submission

Paper ID 26

Abstract

Image classification systems inherit biases from uneven group representation, e.g., blond hair disproportionately associated with females in face datasets, reinforcing stereotypes. A recent approach leverages the Stable Diffusion model to generate balanced training data, but these models often struggle to preserve the original data distribution. In this work, we explore multiple diffusion-finetuning techniques, e.g., LoRA and Dreambooth, to generate images that more accurately represent specific training groups by learning directly from their samples. We propose Clustered Dreambooth, clustering group images and training separate models for clusters to handle intra-group diversity. Using these models, we generate images uniformly across groups to pretrain a classification model, followed by finetuning on real data. Experiments on multiple benchmarks demonstrate that the studied finetuning approaches, especially Clustered DreamBooth, outperform vanilla Stable Diffusion on average and achieve results comparable to state-of-the-art debiasing techniques like Group-DRO, while surpassing them as the dataset bias severity increases.

1. Introduction

Image classification models often exhibit harmful biases, posing significant risks for real-world deployment [17, 30, 35]. Biases stem from dataset imbalances; e.g., in CelebA, blond females outnumber blond males, causing misclassification. While numerous debiasing techniques have been proposed [13, 18, 25], mitigating bias becomes increasingly difficult when dataset imbalances become severe. With the recent breakthroughs in image generation using diffusion models like Stable Diffusion [23], we pose a critical question: *Can we harness the generative power of these models to create images that facilitate the training of fair classification systems, even in the presence of extreme dataset bias?*

We first leverage the vanilla Stable Diffusion (SD) [23] to train fair classification models by combining class and bias labels in the prompts (e.g., "`photo of a blond male person`"). However, we find that the generated images often diverge from the original data distribution due to the stochastic nature of diffusion models [27]. They may also fail to follow prompt instructions precisely. Attempts to prompt SD to generate 'waterbird on land' images often produce water backgrounds, even when explicitly instructed otherwise. Generation quality improves with specific, detailed prompts, as demonstrated by FFR [20], which uses exact bird names and background descriptions. However, without precise domain knowledge about the dataset, such prompts risk producing irrelevant or out-of-distribution images. This limitation motivates us to explore methods that can generate in-distribution images by directly learning from the dataset. To address these challenges, we explore LoRA-based finetuning [8] and DreamBooth [24], which finetune Stable Diffusion on specific training groups. Additionally, we introduce Clustered Dreambooth, which clusters images within each group and trains separate Dreambooth models on each cluster to better capture intra-group variations. Using these methods, we generate group-balanced synthetic images to pretrain a classification model, followed by finetuning on real data. Our contributions:

- We explore diffusion models and finetuning mechanisms like LoRA and Dreambooth to generate group-specific images for fair classification. We then propose Clustered Dreambooth, which clusters group images and trains separate Dreambooth models on each cluster to better capture intra-group variations.
- We generate group-balanced synthetic images to pretrain a classification model, followed by finetuning only the softmax on real data.
- Extensive experiments on fairness benchmarks demonstrate that our methods outperform existing approaches, particularly under severe dataset biases where they beat traditional methods like Group-DRO [25] by a large margin.

2. Related Work

Bias Mitigation. Bias mitigation falls into two categories: known and unknown biases. For known biases, where spurious attributes are predefined, methods include worst-group optimization (GroupDRO [25]), last-layer retrain-

ing [13], and semi-supervised approaches with partial bias annotations [11, 18]. For unknown biases, dual-branch networks [15, 18] and contrastive methods [32, 33] refine feature representations by clustering same-class samples and identifying pseudo bias labels.

Data Augmentation using Generative Models. Generative models have been widely used for data augmentation [3, 4, 16, 29, 36]. Early works used GANs [6], while recent methods employ diffusion models. DA-Fusion [29] uses Textual Inversion [5] to generate augmentations, while DiffuseMix [10] combines natural and generated images to combat adversarial attacks.

Generative Models for Debiasing. Generative models have also been used for debiasing classification systems [2, 20, 22, 26]. GAN-based approaches [2, 22] synthesize bias-conflicting samples to augment training data. Diffusion-based methods utilise pre-trained models adapted to generate group-balanced images. FFR [20] generates group-balanced images from Stable Diffusion and finetunes classifiers with real data. We extend this by leveraging LoRA and Dreambooth to generate in-distribution images directly from training groups.

3. Problem Statement and Methodology

3.1. Preliminaries

The goal of this work is to train fairer image classification models using synthetic data from diffusion models. Let $\mathcal{X}$ be the set of real training images, where each $x_i \in \mathcal{X}$ is associated with a class label $y_i \in \mathcal{Y}$, a bias label $a_i \in \mathcal{A}$, and a group label $g_i \in \mathcal{G}$ where $g_i = (y_i, a_i)$. A model $f : \mathcal{X} \rightarrow \mathcal{Y}$ is optimized to classify the images, consising of: a) Feature encoder e, pretrained on a large dataset, and b) Classifier c, finetuned with e to learn class labels. This model is usually trained using the Cross-Entropy (CE) loss (see supplementary A.1). Bias arises when training data is imbalanced across groups, leading to disparities in test performance across groups.

3.2. Generating Synthetic Images

With advancements in generative modeling [23, 28], we explore their potential for training fair classifiers by generating images that reflect the training distribution and enhance minority group generalization.

Vanilla Stable Diffusion (SD). We generate images from each group $g = (y, a)$ by specifying only y and a in the prompts. Since these generations are independent of training data, domain mismatches or misinterpretations may occur as mentioned before.

LoRA-based Finetuned Stable Diffusion. To align generations with training data, we finetune an SD model on each group g, training on $l = \min |g| : g \in \mathcal{G}$ randomly selected samples. Images are then generated using prompts specifying y and a.

Dreambooth. To improve resemblance between training and generated images, we use Dreambooth [24], a text-to-image model that imitates objects or concepts from a small image set. It finetunes a pretrained text-to-image model by learning a unique identifier (e.g., "`[V]`") such that on inference time, if the model is queried by that identifier (e.g., "`photo of a [V] dog`"), it generates new images of the given object. Likewise, we sample 100 images from each training group, and train a separate Dreambooth model h on the same, where the prompt is of the form "`photo of a [V]` y".

Clustered Dreambooth. Dreambooth expertises in learning a concept from $3 - 5$ images. However, a training group like Blond Male consists of images of many individuals sharing a common trait, hair color. To avoid overwhelming a single Dreambooth model, we cluster the CLIP embeddings [21] of images in each group. Let k_D^g denote the number of clusters, where D and g refer to the training dataset and a group in D respectively. We train a pool of Dreambooth models $\mathcal{H}^g = \{h_1^g, h_2^g, h_3^g, \cdots, h_{k_D^g}^g\}$ on the obtained clusters. We implement Clustered-Dreambooth (i.e., the Dreambooth pool $\mathcal{H}^g$) using LoRA-based finetuning [9], which ensures lesser, feasible training time. Finally the trained models are utilized to generate images for each g. For simplicity, we assume equal k_D^g for each group g, and denote the number of clusters as k_D for the rest of the paper.

3.3. Stage 1: Training with the Generated Images

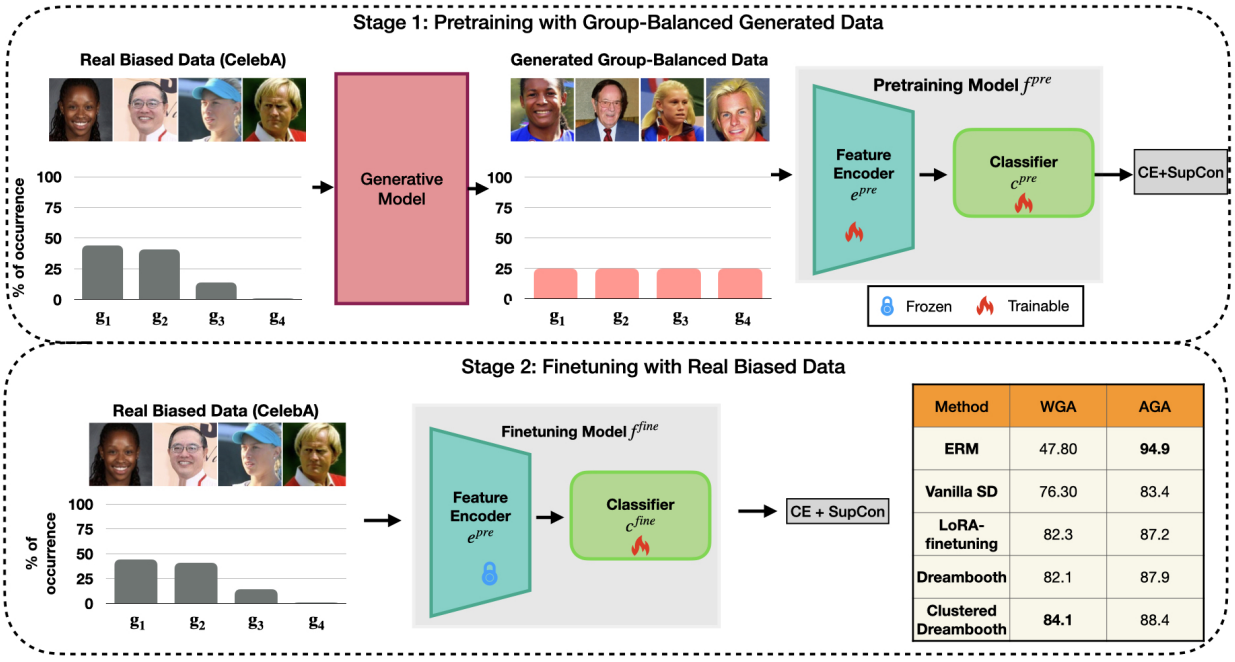
Once the generative models are trained with the individual data groups, we generate M images from each group g using Vanilla SD, LoRA-finetuned SD and Dreambooth. For Clustered Dreambooth, we generate M_D^{cl} images from each cluster in a group belonging to dataset D, such that the total number of images generated from the group is $M_D^{cl} \times k_D = M$. We use a CLIP [21]-based filter to rank and select the top m relevant images per group.

CLIP-based Filtering. To find the most relevant images, we apply a CLIP score in two ways for each image I.

1. **CLIP-Label**$(\mathbf{I}, \mathbf{p^c})$: We compute the image-text similarity of I with a prompt p^c, of the format "`Photo of a {c}`", where c is the class label.
2. **CLIP-Centroid**$(\mathbf{I}, \bar{\mathbf{z}}^{\mathbf{g}})$: To ensure chosen images resemble the group g, we compute the centroid of CLIP embeddings, $\bar{z}^g = \frac{1}{M_g} \sum_{i=1}^{M_g} z_i^g$, where M_g is the group size, and z_i^g is the i^{th} image embedding. We calculate the CLIP similarity between each generated image and its corresponding group centroid.

The final scoring function becomes a combination of CLIP-Label(I, p^c) and CLIP-Centroid$(I, \bar{z}^g)$:

$$\text{CLIP-Score}(I, p^c, \bar{z}^g) = \alpha.\text{CLIP-Label}(I, p^c) + (1 - \alpha).\text{CLIP-Centroid}(I, \bar{z}^g) \quad (1)$$



Method	WGA	AGA
ERM	47.80	**94.9**
Vanilla SD	76.30	83.4
LoRA-finetuning	82.3	87.2
Dreambooth	82.1	87.9
Clustered Dreambooth	**84.1**	88.4

Figure 1. **Overview of the studied pipeline.** In Stage 1, we generate images uniformly from each group (e.g., non-blond female (g_1), non-blond male (g_2), blond female (g_3), blond male (g_4)) using the approaches in Section 3.2 and train a classification model f^{pre} with CE and SupCon losses. In Stage 2, we finetune only the linear classifier on the original dataset. Finally, we compare WGA and AGA across methods, showing Clustered Dreambooth outperforms others.

where α is a hyperparameter. After selecting the top-ranked images from each group, a classification model f is trained on these group-balanced synthetic images. This balanced pretraining enables f^{pre} to learn fair representations.

3.4. Stage 2: Finetuning with Original Data

After pretraining on group-balanced synthetic data, we adapt the model to real data through finetuning.

Last Layer Retraining with Real Data. To prevent bias reintroduction, we finetune only the classification layer c^{pre} of the pretrained model f^{pre}, freezing the feature encoder e^{pre}. To address any class imbalance, each finetuning batch samples classes uniformly. We refer to this method as LLR_{all}, and the finetuned model as f^{fine}. Unlike FFR, which fine-tunes the entire model, our approach reduces hyperparameter dependency. Figure 1 illustrates our two-stage method on CelebA [1].

We train both stages using a weighted sum of CE loss and Supervised Contrastive (SupCon) loss [12] (see supplementary A.1).

4. Experiments and Results

We evaluate diffusion model variants on three datasets and analyze their performance. Implementation details and design choices are elaborated upon in the supplementary material (see section A.2).

4.1. Datasets For Evaluation

Waterbirds [25] consists of bird images labeled as waterbird or landbird, with background bias, only a few waterbird images have land background and vice-versa. CelebA [1] contains 202,599 face images; we use Blond Hair as the target attribute, which exhibits gender bias. UTKFace [34] includes 20,000 face images annotated with age, gender, and ethnicity; we use gender as the target and age as the bias attribute. We report worst-group (WGA) and average-group (AGA) accuracies following prior work [14, 25], additionally evaluating methods under varying bias severity, including extreme cases with a bias ratio of 0.999. This setting is particularly challenging for traditional debiasing methods, as they often struggle to maintain performance under severe bias. Methods used for generation under the severe bias ratio (0.999) scenario are further detailed in supplementary section A.5.

4.2. Key Results

Table 1 presents WGA and AGA across datasets, showing generative models' effectiveness in mitigating bias. Clustered Dreambooth achieves the highest WGA across benchmarks, outperforming other generative approaches and traditional debiasing methods like Group-DRO by capturing intra-group variations. Under severe bias conditions (bias ratio $=$ 0.999), traditional debiasing methods expe-

Table 1. **Final Classification Performance** on the Original Dataset and Bias Ratio 0.999 variant using Vanilla SD, LoRA, Dreambooth, and Clustered Dreambooth across three datasets. While LoRA, Dreambooth, and Clustered Dreambooth achieve worst group accuracy (WGA) comparable to state-of-the-art debiasing methods like GDRO [25] and SELF [14] on the original datasets, they significantly outperform these methods in the high bias-ratio setting. Results are averaged over three random seeds. † denotes implementation using existing codebases. The best and second-best scores are marked in bold and underlined, respectively.

Dataset	Method	Synthetic Data?	Original Dataset		Bias Ratio 0.999		Average Performance	
			Worst	Average	Worst	Average	Worst	Average
Waterbirds	ERM	✗	63.7	88.0	29.0	66.7	46.3	77.3
	FFR† [20]	✓	69.5	84.0	57.3	84.2	63.4	84.1
	Vanilla SD	✓	$74.6^{\pm 2.90}$	$80.5^{\pm 0.34}$	$69.9^{\pm 0.70}$	$80.1^{\pm 0.13}$	72.2	80.3
	LoRA-finetuning	✓	$86.5^{\pm 3.81}$	$89.9^{\pm 0.76}$	$61.5^{\pm 0.40}$	$84.0^{\pm 0.11}$	74.0	87.0
	Dreambooth [24]	✓	$\mathbf{89.3}^{\pm 0.75}$	$\underline{90.1}^{\pm 0.50}$	$82.4^{\pm 0.25}$	$\underline{88.3}^{\pm 0.22}$	$\underline{85.9}$	$\underline{89.2}$
	Clustered Dreambooth	✓	$88.1^{\pm 0.92}$	$\mathbf{90.2}^{\pm 0.11}$	$\mathbf{84.2}^{\pm 0.46}$	$\mathbf{88.5}^{\pm 0.14}$	**86.0**	**89.3**
	GDRO† [25]	✗	91.4	93.5	23.5	65.5	57.4	79.5
	SELF† [14]	✗	93.0	94.0	25.5	64.2	59.2	79.1
CelebA	ERM	✗	47.8	94.9	31.7	67.3	39.7	81.1
	FFR† [20]	✓	68.9	85.7	22.8	47.7	45.9	66.7
	Vanilla SD	✓	$76.4^{\pm 1.27}$	$84.2^{\pm 0.37}$	$77.1^{\pm 0.42}$	$\underline{84.7}^{\pm 0.67}$	76.7	84.4
	LoRA-finetuning	✓	$\underline{82.3}^{\pm 1.51}$	$87.2^{\pm 0.56}$	$73.5^{\pm 2.83}$	$83.2^{\pm 0.29}$	77.9	85.2
	Dreambooth [24]	✓	$82.1^{\pm 0.00}$	$\underline{87.9}^{\pm 0.32}$	$\underline{78.8}^{\pm 0.21}$	$84.6^{\pm 0.19}$	$\underline{80.4}$	$\underline{86.2}$
	Clustered Dreambooth	✓	$\mathbf{84.1}^{\pm 0.63}$	$\mathbf{88.4}^{\pm 0.19}$	$\mathbf{81.8}^{\pm 0.35}$	$\mathbf{85.9}^{\pm 0.28}$	**82.9**	**87.1**
	GDRO† [25]	✗	88.9	92.9	27.2	75.2	58.0	84.0
	SELF† [14]	✗	83.9	91.1	45.6	95.4	64.7	93.2
UTKFace	ERM	✗	74.3	84.5	31.0	48.9	52.6	66.7
	FFR† [20]	✓	67.4	81.4	55.0	68.0	61.2	74.7
	Vanilla SD	✓	$62.0^{\pm 3.89}$	$83.3^{\pm 0.92}$	$\mathbf{67.8}^{\pm 1.27}$	$\mathbf{82.7}^{\pm 0.61}$	64.9	$\underline{83.0}$
	LoRA-finetuning	✓	$68.6^{\pm 3.91}$	$\mathbf{85.6}^{\pm 0.76}$	$64.5^{\pm 2.59}$	$\underline{82.4}^{\pm 0.26}$	$\underline{66.5}$	**84.0**
	Dreambooth [24]	✓	$57.9^{\pm 3.91}$	$80.9^{\pm 0.76}$	$\underline{66.9}^{\pm 2.59}$	$77.1^{\pm 0.26}$	62.4	79.0
	Clustered Dreambooth	✓	$\mathbf{76.0}^{\pm 1.22}$	$\underline{83.5}^{\pm 0.35}$	$60.5^{\pm 1.22}$	$80.8^{\pm 0.35}$	**68.2**	82.1
	GDRO† [25]	✗	81.6	85.9	30.5	50.3	56.0	68.1
	SELF† [14]	✗	65.9	82.3	0.6	50.5	33.3	66.4

rience significant performance degradation, while generative methods, particularly Clustered Dreambooth, remain robust due to their inherent imagination abilities. This robustness is attributed to the diversity of synthetic images generated by clustering within groups, which helps mitigate the impact of extreme dataset imbalances. On UTKFace, vanilla Stable Diffusion slightly outperforms others. Overall, our results demonstrate that Clustered Dreambooth not only enhances fairness but also maintains strong performance across varying bias severities, making it a promising approach for training fair classifiers. We also conduct further experiments highliting the effectiveness of generative methods and its nuances like time complexity in the supplementary material.

5. Conclusion

In this work, we explored the use of diffusion models and their finetuning mechanisms, such as LoRA and Dreambooth, to generate group-specific synthetic images for training fair classification models. We introduced Clustered Dreambooth, which clusters group images and trains separate Dreambooth models on each cluster to better capture intra-group variations. Extensive experiments on multiple fairness benchmarks demonstrated that our methods outperform existing approaches, particularly under severe dataset biases showing that it is possible to train robust classifiers that remain effective even in highly imbalanced settings using generative models. Future work will focus on improving generative diversity and unsupervised methods that do not require group labels, enabling automatic group detection.

References

[1] CelebA dataset. http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html. 3, 2, 4, 6

[2] Jaeju An, Taejune Kim, Donggeun Ko, Sangyup Lee, and Simon S Woo. Aˆ 2: Adaptive augmentation for effectively mitigating dataset bias. In *Proceedings of the Asian Conference on Computer Vision*, pages 4077–4092, 2022. 2

[3] Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J Fleet. Synthetic data from diffusion models improves imagenet classification. *arXiv preprint arXiv:2304.08466*, 2023. 2

[4] Xuefeng Du, Yiyou Sun, Jerry Zhu, and Yixuan Li. Dream the impossible: Outlier imagination with diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[5] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 2, 7

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[7] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 6

[8] Edward Hu et al. Lora: Efficient fine-tuning of large models. Hugging Face Blog, 2021. Accessed: 2025-03-06. 1

[9] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2

[10] Khawar Islam, Muhammad Zaigham Zaheer, Arif Mahmood, and Karthik Nandakumar. Diffusemix: Label-preserving data augmentation with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27621–27630, 2024. 2

[11] Sangwon Jung, Sanghyuk Chun, and Taesup Moon. Learning fair classifiers with partially annotated group labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10348–10357, 2022. 2

[12] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020. 3, 1

[13] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022. 1, 2

[14] Tyler LaBonte, Vidya Muthukumar, and Abhishek Kumar. Towards last-layer retraining for group robustness with fewer annotations. *arXiv preprint arXiv:2309.08534*, 2023. 3, 4

[15] Jungsoo Lee, Eungyeup Kim, Juyoung Lee, Jihyeon Lee, and Jaegul Choo. Learning debiased representation via disentangled feature augmentation. *Advances in Neural Information Processing Systems*, 34:25123–25133, 2021. 2

[16] Giovanni Mariani, Florian Scheidegger, Roxana Istrate, Costas Bekas, and Cristiano Malossi. Bagan: Data augmentation with balancing gan. *arXiv preprint arXiv:1803.09655*, 2018. 2

[17] Danaë Metaxa, Michelle A Gan, Su Goh, Jeff Hancock, and James A Landay. An image of society: Gender and racial representation and impact in image search results for occupations. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–23, 2021. 1

[18] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: Training debiased classifier from biased classifier. In *Advances in Neural Information Processing Systems*, 2020. 1, 2

[19] Maan Qraitem, Kate Saenko, and Bryan A Plummer. Bias mimicking: A simple sampling approach for bias mitigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20311–20320, 2023. 1, 2

[20] Maan Qraitem, Kate Saenko, and Bryan A Plummer. From fake to real (ffr): A two-stage training pipeline for mitigating spurious correlations with synthetic data. *arXiv preprint arXiv:2308.04553*, 2023. 1, 2, 4, 6, 7

[21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2

[22] Vikram V Ramaswamy, Sunnie SY Kim, and Olga Russakovsky. Fair attribute classification through latent space de-biasing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9301–9310, 2021. 2

[23] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1, 2, 7

[24] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv preprint arXiv:2208.12242*, 2022. 1, 2, 4, 3, 7

[25] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019. 1, 3, 4, 2, 6

[26] Viktoriia Sharmanska, Lisa Anne Hendricks, Trevor Darrell, and Novi Quadrianto. Contrastive examples for addressing the tyranny of the majority. *arXiv preprint arXiv:2004.06524*, 2020. 2

[27] Joonghyuk Shin, Minguk Kang, and Jaesik Park. Fill-up:

Balancing long-tailed data with generative models. *arXiv preprint arXiv:2306.07200*, 2023. 1

[28] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[29] Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. *arXiv preprint arXiv:2302.07944*, 2023. 2

[30] Jialu Wang, Yang Liu, and Xin Eric Wang. Are gender-neutral queries really gender-neutral? mitigating gender bias in image search. *arXiv preprint arXiv:2109.05433*, 2021. 1

[31] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200, 2010. 6

[32] Michael Zhang and Christopher Ré. Contrastive adapters for foundation model group robustness. *Advances in Neural Information Processing Systems*, 35:21682–21697, 2022. 2

[33] Michael Zhang, Nimit S Sohoni, Hongyang R Zhang, Chelsea Finn, and Christopher Ré. Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations. *arXiv preprint arXiv:2203.01517*, 2022. 2

[34] Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5810–5818, 2017. 3, 1, 2

[35] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *arXiv preprint arXiv:1707.09457*, 2017. 1

[36] Chenyu Zheng, Guoqiang Wu, and Chongxuan Li. Toward understanding generative data augmentation. *Advances in neural information processing systems*, 36:54046–54060, 2023. 2

[37] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 6

Harnessing Diffusion-Generated Synthetic Images for Fair Image Classification

Supplementary Material

We organize this appendix as follows. We begin with details on the loss functions used to train our classification pipeline (subsection A.1), elaboration about the implementation details and some design choices (subsection A.2), followed by the data split for the UTKFace dataset [34](subsection A.3). Next, we describe the clustering process used to implement the Clustered Dreambooth pipeline (subsection A.4), the method used for generation under the sever bias ratio scenario (subsection A.5) and the prompts used for generating images from different diffusion model variants (subsection A.6). We then present key ablations on our design choices in subsection A.7, where we demonstrate the importance of the CLIP-Score introduced in eq. 2 (subsection 3.3 in the main paper), analyze various strategies for the classification finetuning stage, and examine the impact of selecting different proportions of synthetic images for pretraining. Additionally, we discuss the roles of CE and SupCon losses in the classification pipeline. We conclude with an analysis of hyperparameter effects during pretraining with group-balanced synthetic data (subsection A.8), an evaluation of representation similarity between Clustered Dreambooth images and real data (subsection A.9), a report on worst-group classification performance after the pretraining stage (subsection A.10), and qualitative examples of images generated by different methods explored in this work (subsection A.11).

A. Experiments - Extended Details

A.1. Loss Function for the Two-Stage Pipeline

Recall that our two stage pipeline is trained by a weighted combination of the Cross-Entropy (CE) loss and the Supervised Contrastive (SupCon) loss. We define both of them formally here. Given the set of training images $\mathcal{X}$, where each $x_i \in \mathcal{X}$ is associated with a class label $y_i \in \mathcal{Y}$, the CE loss is defined as:

$$L_{\text{CE}} = \frac{1}{|\mathcal{X}|} \sum_{i=1}^{|\mathcal{X}|} \sum_{j=1}^{|\mathcal{Y}|} -p_{ij} \log \hat{p}_{ij} \tag{2}$$

where $[p_{i1}, p_{i2}, \cdots, p_{i|\mathcal{Y}|}]$ is the one-hot vector representation of y_i and $[\hat{p}_{i1}, \hat{p}_{i2}, \cdots, \hat{p}_{i|\mathcal{Y}|}]$ is the corresponding softmax vector, obtained from the model classification model f (see Section 3.1 in the main paper).

The SupCon loss encourages the model to learn more discriminative features by promoting greater separation between samples from different classes. We consider a mini-batch of size $\mathcal{B}$ of features denoted as $\{\mathbf{e(x_1)}, \mathbf{e(x_2)}, \cdots, \mathbf{e(x_\mathcal{B})}\}$ and corresponding class labels as $\{y_1, y_2, \ldots, y_\mathcal{B}\}$ (e is the feature encoder). Let us consider the current sample with index j, and the set of positive examples from the mini-batch by $P_j : \{i \in \mathcal{B} \quad s.t. \quad y_i = y_j\}$. Similarly, the set of negative examples is denoted by $N_j : \{i \in \mathcal{B} \quad s.t. \quad y_i \neq y_j\}$. The SupCon loss [12] for a image x_j is defined as,

$$L_{\text{sup-con}} = \sum_{j \in \mathcal{B}} \frac{-1}{|P_j|} \sum_{p \in P_j} \log \frac{\exp(\mathbf{e(x_j)}^T \mathbf{e(x_p)}/\tau)}{\sum\limits_{n \in N_j} \exp(\mathbf{e(x_j)}^T \mathbf{e(x_n)}/\tau)}, \tag{3}$$

We set the temperature $\tau = 1$ for all experiments as we do not find any significant improvements by changing its value. The final loss function becomes a combination of the CE loss and SupCon loss:

$$L = \beta . L_{\text{CE}} + (1 - \beta) L_{\text{sup-con}}$$

where $\beta = 0.5$ for all experiments.

Table 2. **Classification Performance vs Time Complexity Tradeoff** for the generative-based methods, averaged across all datasets and both bias ratios. While Clustered Dreambooth incurs high time complexity, it is the best performing method on average, followed by Dreambooth and LoRA-finetuning.

Method	Time Complexity	WGA	AGA
Vanilla SD	$\mathcal{O}(1)$	71.3	82.6
LoRA-finetuning	$\mathcal{O}(\lvert\mathcal{G}_D\rvert)$	72.8	85.4
Dreambooth	$\mathcal{O}(\lvert\mathcal{G}_D\rvert)$	76.2	84.8
Clustered Dreambooth	$\mathcal{O}(\lvert\mathcal{G}_D\rvert.k_D)$	**79.1**	**86.2**

A.2. Implementation Details and Design Choices

We use SD v1.4 for all variants, generating $M = 5000$ images per group. Images are ranked using the CLIP-Score (eq. 1), and the top 75% are selected. A ResNet-50 model is pretrained on synthetic data and finetuned on real data for 20 epochs using SGD. Hyperparameters are uniformly set across datasets. We analyze the tradeoff between performance and time complexity as well (Table 2). While Clustered DreamBooth incurs higher time complexity, it achieves the best overall performance. The method used for generating images for the severse bias ratio case is displayed in Figure 2 and explained further in A.5.

A.3. UTKFace Splits

Following previous works [19, 20], we deliberately introduce biases into the UTKFace dataset [34], wherein we use Gender as the target attribute and Age as the bias attribute

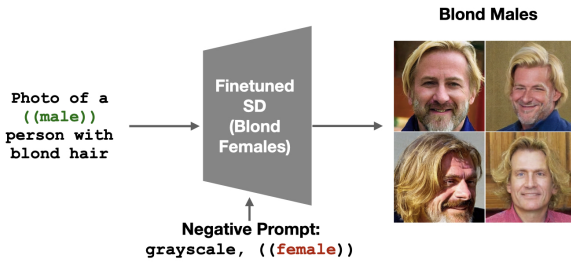


Figure 2. **Image Generation Pipeline for Bias Ratio= 0.999**. For approaches that finetune SD on training images, bias-conflicting samples (e.g., Blond Males in CelebA) are generated using diffusion models trained on bias-aligned images (e.g., Blond Females). Here, we omit the V token in Dreambooth prompts for more accurate target group generation.

with 90% bias ratio. Particularly, we ensure that the number of images in the training set for the different groups are: 103, 934, 5730 and 636 for male adults, male children, female adults and female children respectively. To binarize the Age attribute, ages ≤ 10 are considered as children, and those ≥ 20 are considered adults [19, 20]. Thus, the females are biased towards older age, whereas the males are biased towards the children.

A.4. Choice of Clusters for Clustered Dreambooth

Recall that Clustered Dreambooth first clusters the images from a group into k_D clusters (D is the dataset) and then trains Dreambooth [24] models on the individual clusters to ensure that each Dreambooth model is trained on similar images. We choose k_D based on the size of the smallest group in D, denoted by M_{g_s} (g_s represents the smallest group). To ensure that the Dreambooth models have enough images to train on for each cluster, we choose k_D such that approximately atleast 20 samples are present per cluster in the smallest group, i.e., $k_D \approx \frac{m_{g_s}}{20}$. However, for larger datasets, k_D may become large as the smallest group grows in size, increasing the complexity of Clustered Dreambooth. Therefore, we choose K_D to be $\min(\frac{M_{g_s}}{20}, 20)$. Likewise, for Waterbirds [25] ($M_{g_s} = 56$), we choose $k_D = 3$, for CelebA [1] ($M_{g_s} = 1387$), k_D is fixed at 20, and for UTKFace [34] ($M_{g_s} = 103$), k_D is set to 5. Note that k_D is fixed for a single dataset for simplicity, i.e., it is same for groups other than the smallest group in the dataset, assuming that the intra-group variations are generally consistent across multiple training groups. Further optimization on the choice of k_D is left as a future work.

A.5. Generation method for varying bias severity

For the scenarios where the dataset is severely biased, setting the bias ratio to 0.999 (i.e., 99.9% of training images belong to bias-aligned groups), the diffusion models must generate images for minority groups to counteract bias severity. For Vanilla SD and FFR, images are generated by prompting with the bias label a and class label y. As LoRA-finetuned SD, DreamBooth, and Clustered DreamBooth rely on the training group images, the bias-conflicting images are generated using models trained on bias-aligned groups (e.g., Blond Males are generated from the model trained on Blond Females). Interestingly, for the Dreambooth models, we find that during the bias-conflicting sample generation, removing the learnt V token from the prompt leads to more accurate depiction of the target group descriptions. On manual inspection, we find that this way of transferring the style of one group into another (described in Fig. 2) leads to images that visually follow the distribution of the input data, while imitating the target group. Generated images are filtered using the CLIP-Label score with $\alpha = 1$, as the minority groups lack sufficient samples for the CLIP-Centroid computation. The classifier is then pretrained on group-balanced synthetic images and finetuned on the severely biased dataset for each method.

A.6. Prompts used for Generation

Here we present the prompts we used for each method and dataset to generate images from every group.

Prompts for Vanilla SD [23]. For Waterbirds, the prompt used is "`photo of a {class-label} on {bias-label}.`", where class-label can be landbird or waterbird, and bias-label can be land or water. For UTKFace, the prompt used is "`photo of a {class-label} {bias-label}.`", where class-label can be female or male, bias-label can be child or adult. Finally, for CelebA, we use the following template: "`photo of a {bias-label} person with blond hair`" to generate males and females (i.e., the bias-labels) with blond hair. For the non-blond class, we prompt the model with "`photo of a {bias-label} person`", whereas we use a negative prompt having '`blond hair`', to force the model to generate non-blond males and females. As the vanilla SD is not finetuned on our training sets, we use the same set of prompts for the bias ratio 0.999 case.

Prompts for Finetuned Diffusion Models. For LoRA-finetuning, we use prompts of the format '`Photo of a {class-label} on {bias-label}`' for Waterbirds (`class-label` $\in$ {`waterbird, landbird`}, `bias-label` $\in$ {`water, land`}), and '`Photo of a {class-label} person who is a {bias-label}`' for UTKFace (`class-label` $\in$ {`male, female`}, `bias-label` $\in$ {`adult, child`}). For CelebA, the prompts used are: '`Photo of a non-blond {bias-label} person`' for the class Non-Blond, and '`Photo of a {bias-label} person with blond hair`' for the class Blond, where `bias-label` $\in$ {`male, female`}. We use

Table 3. **Effect of CLIP-Score weighting parameter** α on Clustered Dreambooth final worst group accuracies. We observe that setting $\alpha = 1$ outperforms $\alpha = 0$, highlighting the effectiveness of CLIP based similarity between the textual form of the class label and the images. Setting $\alpha = 0.5$ (as reported in the main paper) works best for most datasets. Randomly selecting the images without using any scoring functions is also seen to perform on par with the other settings, the performance is generally weaker. All scores are with respect to the original versions of the datasets.

Selection Method	Waterbirds		CelebA		UTKFace	
	WGA	AGA	WGA	AGA	WGA	AGA
$\alpha = 1$	87.0	89.9	83.8	87.9	72.7	84.5
$\alpha = 0$	84.7	90.15	83.3	87.95	68.8	82.9
$\alpha = 0.5$	**88.1**	90.2	**84.1**	**88.4**	**76.0**	83.5
Random sampling	84.9	**90.4**	81.1	87.0	73.0	**84.62**

the same set of prompts for Dreambooth [24] (i.e., single model per group) and Clustered Dreambooth. Recall that each of these methods learn specific tokens to represent the groups (or clusters in the groups in case of Clustered Dreambooth). We denote the learnt tokens by '`[V]`'. Likewise, for Waterbirds, we use the prompt "`photo of a [V] bird`", where '`[V]`' represents the learnt tokens by the model trained on the specific group or cluster. For UTKFace and CelebA, we find that providing the class-label in the prompt generates more accurate images. Hence, for UTKFace, the prompt is of the form "`photo of a [V] {class-label} person`". For CelebA, the blond-class images are generated using "`photo of a [V] person with blond hair`", whereas for the non-blond class, the prompt is "`photo of a [V] person`" with blond hair as the negative prompt.

Prompts for Bias Ratio $= 0.999$ Scenario. The prompts used are similar to the case of the original dataset for each generation method. For UTKFace, we generate the female children faces from model(s) trained on the male children faces, and male adult faces from those trained on the female adult faces. To enforce the model to generate correct images from the bias-conflicting groups, we emphasize the `class-label` in the prompt by placing it inside double parenthesis. We also add the opposite class-label to the negative prompt with double parenthesis. For example, the generation prompt for female children is '`Photo of a ((female)) person who is an child`', with an additional negative prompt '`((male))`'. For CelebA, we generate blond males from blond female models, and non-blond females from non-blond male models. For Waterbirds, we generate landbird on water images from landbird on land models, whereas waterbird on land images are generated from waterbird on water images. Similar to the UTKFace case, we put double parenthesis in the prompts, but on the `bias-label` for these two datasets, with the opposite bias-label added to the negative prompts.

A.7. Extended Ablation Studies

Here we present further ablation studies of the proposed approach as an extension to Section 4.4 in the main paper.

Role of The CLIP-Score. We have described the CLIP-Score in eq. 2 (main paper, subsection 3.3), used to filter the best 75% images out of the generated ones for each group. We vary the weighting parameter α to understand the role of the label-based score function CLIP-Label(I, p^c) and the group-centroid based score function CLIP-Centroid$(I, \bar{z}^g)$. Setting $\alpha = 1$ denotes that the scoring function is only dependent on CLIP-Label, whereas $\alpha = 0$ denotes otherwise. We present the results of these variants on the Clustered Dreambooth pipeline in Table 3 (with respect to the original dataset versions). Recall that the numbers reported in the main paper correspond to $\alpha = 0.5$. We also show a baseline where from each group, images are selected randomly instead of ranking them using the scoring function. We observe that the performance of the pretrained model trained on the images chosen by setting $\alpha = 1$ always outperforms its counterpart trained on images selected by setting $\alpha = 0$. However, in general, our choice of $\alpha = 0.5$ works best across datasets. Random selection also performs on par with the other variants, which shows that the generated images are mostly useful in training fairer classifiers, however, their performances are lower than those involving image selection with the CLIP-Score.

Direct Combination of Real and Synthetic Data. We observe two settings: a) *Real+Group-Balanced Synthetic* Images: Combine the entire real data and the group-balanced synthetic images generated using Clustered Dreambooth, b) *Group-Balanced (Real+Synthetic) Images*: Combine the real and the synthetic data in such a way that the final images are group-balanced. For both settings, there is only a single stage of training, with the combination of real and synthetic images. Using the images obtained from each stage, we train a classification model, and compare their performances with the pretraining and finetuning stage of the Clustered Dreambooth pipeline. Our experiments show

that for Waterbirds and CelebA, the performance drops for both the settings compared to the Clustered Dreambooth pretraining and finetuning stages. UTKFace is an exception, where the worst group-accuracy is high for the *Real+Group-Balanced Synthetic* setting. However, the average group accuracy drops, showing that the accuracy of the groups other than the worst group remains low. Overall, we observe that our two stage approach is considerably more effective than the single stage alternatives (Table 4), justifying our choice in the main paper.

Effect of Selection Percentage of Generated Images. In this subsection, we study the effect of selecting different percentages of top ranked images as per the CLIP-Score defined in eq. 2 (Section 3.3, main paper) on Waterbirds (both original and the severely biased variants) for Clustered Dreambooth. Investigating the classifier performance based on top 100%, 75% and 50% generated images, across both the dataset variants, we find that selecting the top 75% of the synthetic images appears to be more beneficial for performance, though we do not observe a drastic fall in scores with the other percentages as well. The results are shown in Table 5.

Effect of the Loss Functions in Stage 1 and Stage 2 Training. To train the classification model in the synthetic image pretraining stage and real image finetuning stage, we use a weighted combination of the CE and SupCon losses. Here, we show the importance of combining these losses in both stages, by demonstrating their effects on the performance of Clustered Dreambooth for Waterbirds and UTKFace, in case of both the original dataset and the severely biased variant (see Tables 6 and 7 respectively). We find that while the effect of the SupCon loss is less pronounced for the original versions of both the datasets, its impact is clearly visible for the severely biased versions, especially for UTKFace. Recall that for Clustered Dreambooth, the images of the bias-conflicting samples are generated from the diffusion models finetuned on the bias-aligned images. The SupCon loss helps bring the samples of bias aligned and bias conflicting groups within the same class together in the feature space, thus facilitating improved learning of their representations, as the resultant bias-conflicting images may deviate from the original data distribution as well as those of the generated bias-aligned samples.

Group-Balanced Finetuning. After pretraining on generated data, we finetune the classification layer on real data (LLR_{all}) (see Section 3.3, main paper). To assess the benefits of group-balanced real data (sized to the smallest group), we explore: a) *Last Layer Retraining with Balanced Real Data* (LLR_{b}): Finetuning only the classification layer using the balanced real dataset instead of the full biased set. b) *Full Fine-Tuning with Balanced Real Data* (FT_{b}): Finetuning the entire network with the balanced real dataset. Results show that FT_{b} is beneficial only for CelebA, likely due to its larger size compared to Waterbirds and UTKFace (Table 8 for Clustered Dreambooth). In contrast, LLR_{b} reduces worst group accuracy across all datasets, indicating that finetuning on the entire dataset yields better performance.

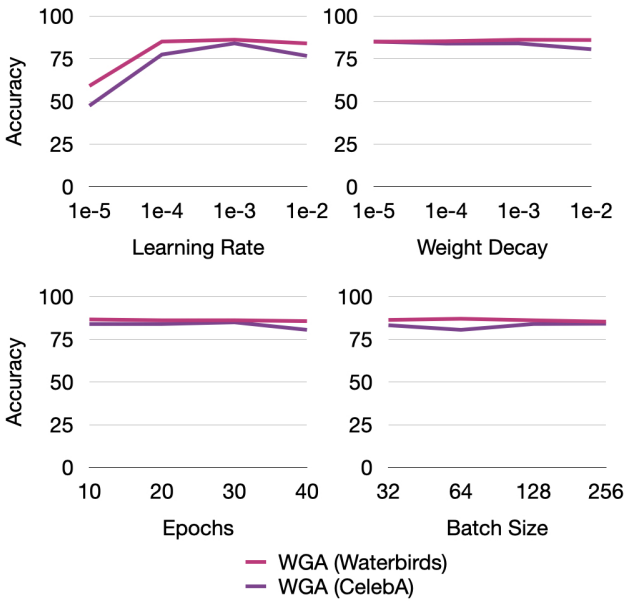


Figure 3. **Variations in worst group accuracies (WGA) with changing Hyperparameter values** for the pretraining stage on the CelebA [1] and Waterbirds [25] datasets. Apart from a drastic fall at learning rate=$1e-5$, we do not observe any significant variation in the model performances across the observe range of hyperparameters.

A.8. Hyperparameters for Classification

While we do not experiment with hyperparameters during classification, using a learning rate of $1e-3$, weight decay of $1e-3$, batch size of 128 and total epochs = 20, we assess the variance of the classifier worst group performance on the pretraining stage of CelebA [1] and Waterbirds [25] using the Clustered Dreambooth images. Specifically, we explore a set of learning rates : $\{1e-5, 1e-4, 1e-3, 1e-2\}$, a range of weight decay values: $\{1e-5, 1e-4, 1e-3, 1e-2\}$, training epochs: $\{10, 20, 30, 40\}$, and batch sizes: $\{32, 64, 128, 256\}$. We observe a drastic fall in accuracy scores with learning rate $= 1e-5$ for both datasets, which is expected, considering the low value of the learning rate. While the classifier performance on Waterbirds seems more consistent across hyperparameter values than that on CelebA, we do not observe any significant variation in model performance for different hyperparameters for any of the examined datasets. We plot our findings in Fig. 3.

We next describe the hyperparameters used for synthetic

Table 4. **Classification Performance on Different Combinations of Real and Synthetic Data**. We evaluate classification systems for two cases: a) The entire real data + group-balanced synthetic images, b) Combination of real and synthetic data ensuring that the resultant dataset is group-balanced. These cases are compared against the pretraining and finetuning stages reported in the main paper for the Clustered Dreambooth pipeline. The experiment shows the two observed cases to be inadequate compared to our pipeline of pretraining with synthetic data and finetuning with real data.

Selection Method	Waterbirds		CelebA		UTKFace	
	WGA	AGA	WGA	AGA	WGA	AGA
Real+Group-Balanced Synethic Images	77.9	88.4	48.3	83.2	74.8	82.5
Group-Balanced (Real+Synthetic) Images	79.1	88.5	46.7	82.4	70.4	82.0
Real Images (Finetuning Stage)	88.1	90.2	84.1	88.4	76.0	83.5

Table 5. **Percentage of Synthetic Images selected for Stage 1 Training vs Performance** for Clustered Dreambooth with respect to Waterbirds. We find that across the moderate and severe bias ratios, while scores do not vary drastically with selection percentages, it is beneficial to choose the top 75% images, as it leads to better performance than the other percentages across both the dataset variants.

Selection Percentage	Clustered Dreambooth		Bias Ratio 0.999	
	WGA	AGA	WGA	AGA
50%	86.4	89.8	82.5	88.1
75%	**88.1**	**90.2**	**84.2**	**88.5**
100%	**88.1**	90.0	81.8	87.2

Table 6. **Role of the SupCon loss on the performance of Clustered Dreambooth in case of Waterbirds**, for both the original and severely biased variants. Recall that $\beta = 1$ denotes only using the CE loss, while $\beta = 0.5$ refers to both losses having equal weight. While the performance on the original dataset does not show any significant change, the supcon loss improves accuracies for the high bias ratio version of the dataset.

β		Waterbirds Original		Waterbirds Bias Ratio=0.999	
Stage 1	Stage 2	WGA	AGA	WGA	AGA
1	1	88.2	90.2	83.3	88.0
1	0.5	88.1	90.2	83.8	88.0
0.5	1	88.2	90.2	84.0	88.0
0.5	0.5	88.1	90.2	84.2	88.5

data generation. For all the facial datasets, we include a negative prompt that discourages the model to generate grayscale images. While training the Dreambooth models, we use the default set of hyperparameters for all datasets. For LoRA-finetuning, we use the r and α parameters of LoRA as 16, and use no learning rate scheduler. The model is finetuned for 200 training steps. For all diffusion based methods, we set guidance-scale $= 7.5$ and the number of timesteps to be 50, except Clustered Dreambooth for UTKFace, where the number of timesteps is set to be 25, based on manual inspection.

Table 7. **Role of the SupCon loss on the performance of Clustered Dreambooth in case of UTKFace**, for both the original and severely biased variants. Recall that $\beta = 1$ denotes only using the CE loss, while $\beta = 0.5$ refers to both losses having equal weight. While the performance on the original dataset shows slight increases, the supcon loss considerably improves accuracies for the high bias ratio version of the dataset.

β		UTKFace Original		UTKFace Bias Ratio=0.999	
Stage 1	Stage 2	WGA	AGA	WGA	AGA
1	1	72.7	88.2	49.6	80.7
1	0.5	73.6	83.4	50.4	80.8
0.5	1	76.0	83.5	60.0	80.4
0.5	0.5	76.0	83.5	60.5	80.8

Table 8. **Finetuning with group-balanced training data**. Upon manipulating the pretrained model with group-balanced training data, with and without finetuning the feature encoder e^{pre} (FT_b and LLR_b respectively), we find that they are generally not advantageous, compared to LLR_all. The only exception is CelebA, where the WGA becomes 88.80% for FT_b. CD: Clustered Dreambooth

Dataset	Method	WGA	AGA
Waterbirds	CD + LLR_all	$87.8^{\pm 0.46}$	$90.2^{\pm 0.14}$
	CD + LLR_b	$87.4^{\pm 0.31}$	$90.2^{\pm 0.12}$
	CD + FT_b	$87.7^{\pm 0.30}$	$90.7^{\pm 0.01}$
CelebA	CD + LLR_all	$85.0^{\pm 0.35}$	$88.6^{\pm 0.28}$
	CD + LLR_b	$83.9^{\pm 1.36}$	$88.2^{\pm 0.28}$
	CD + FT_b	$88.8^{\pm 1.32}$	$91.4^{\pm 0.39}$
UTKFace	CD + LLR_b	$76.0^{\pm 1.22}$	$83.5^{\pm 0.35}$
	CD + LLR_b	$70.20^{\pm 1.47}$	$83.78^{\pm 1.66}$
	CD + FT_b	$74.6^{\pm 2.55}$	$83.8^{\pm 0.20}$

Table 9. **Evaluation of FID (↓).** The Clustered Dreambooth images outperform those of the other generative methods for all four groups of the CelebA dataset (Non-Blond Female (NF), Non-Blond Male (NM), Blond Female (BF), Blond Male (BM)).

Method	NF	NM	BF	BM
Vanilla SD	128.57	133.53	78.06	92.48
FFR [20]	90.66	98.16	66.61	100.27
Dreambooth	71.02	69.18	47.04	58.51
Clustered Dreambooth	**58.72**	**56.48**	**38.50**	**46.90**

A.9. Quality of Representations learnt via Synthetic Data.

To assess the similarities between the real and the generated distributions for each of the investigated approaches, we determine the Fréchet Inception Distance (FID) [7] measured between the generated images and real images of each group of CelebA. Notably, Clustered Dreambooth images outperform all competing methods by a large margin, pointing to their similarity with the training images (see Table 9).

A.10. Performance of Stage 1 pretraining on Test Data

Recall that we pretrain the classification model on group-balanced synthetic images for each dataset. Evaluating on the test set, we find the worst group accuracy of such a model to be surprisingly high for the finetuned diffusion-based models, especially for Clustered Dreambooth, even without training the model on a single sample from the training set. This shows that the synthetic images resemble the real images closely. We present the results for the original training sets in Table 10.

A.11. Qualitative Examples

Images generated by different models of Clustered Dreambooth. Here, we show examples from four different clusters in each group in the CelebA dataset [1]. We see that some of these models, while preserving the characteristics of its group (e.g. gender and hair color), captures different attributes like age, skin color, profession, hair color shades, etc. These variations can be clearly seen in Fig. 4, where the clusters are seen to generate older people, children, young adults, people of fair or dark skin, people from Indian descent, etc for the various groups. Interestingly, for all groups in CelebA, we also find clusters representing sportspeople as well, some of which are shown in the figure.

Vanilla SD failures in Waterbirds Vanilla SD models often fail to follow prompts accurately, perhaps because of inherent biases embedded in them. For example, when the model is instructed to generate '`Photo of a waterbird on land`', many of the generated images have water in them, even when we set a negative prompt '`water`'. This problem is highlighted in Fig. 5, where more than 50% of the images have water in the images. Moreover, while we agree that the generations are aesthically of high quality, the Waterbirds [25] dataset itself is created by pasting bird images from the Caltech-UCSD Birds-200-2011 (CUB) dataset [31] into images from the Place dataset [37], which often gives the images from the original dataset an unnatural look. Thus, this leads to a domain mismatch between the training and generated images, explaining the lower worst group accuracies of Vanilla SD.

Failure cases of Clustered Dreambooth for UTKFace (Bias Ratio 0.999). As observed in Table 1 (main paper), Clustered Dreambooth's performance suffers for the severely biased version of UTKFace compared to other methods. The worst group is Female Children, with an accuracy of 60.5%, which is 15.5% lower than that of the original data variant. We examine the reasons behind this performance drop by manually inspecting the Female Children images, and find that while many images are grayscale (inspite of having the word grayscale in the negative prompt), some are of Male Children. Alarmingly, some images fail to follow the training data domain, even though they correctly belong to Female Children. Such images get selected by our CLIP-Score as they accurately reflect the group description, and in the absence of the training images for the given group, we cannot compute the CLIP-Centroid Score to ensure that the selected images are as close to the training data domain as possible. We believe such issues potentially contribute towards the low worst group accuracies for the UTKFace Female Children group. Example images are shown in Fig. 6.

Generated Bias-Conflicting Samples from Clustered Dreambooth (Bias Ratio $= 0.999$). Recall that for bias ratio 0.999, bias-conflicting images are generated using models trained on bias-aligned groups (e.g., Blond Males are generated from the model trained on Blond Females) for LoRA-finetuning, Dreambooth and Clustered Dreambooth. We present samples generated by the Clustered Dreambooth from the bias-conflicting groups of each dataset in Fig. 7, and observe that the generated images resemble the distribution of the input dataset, while reflecting the requirements of the target group.

Images generated by each Diffusion-based Mechanism. Here, we show the images generated by a) Vanilla SD, b) LoRA-finetuned SD, c) Dreambooth, and d) Clustered Dreambooth for each of the datasets Waterbirds, CelebA and UTKFace (all original versions), for each of the groups present in them. Figures 8, 9 and 10 present the examples for the three datasets (Waterbirds, CelebA and UTKFace) respectively.

Table 10. **Classification Performance.** We report the classifier performance for Stage 1 (Generative Images Pretraining) and Stage 2 (Real Image Finetuning, denoted as LLR_{all}) for Vanilla SD, Dreambooth, and Clustered Dreambooth on the three datasets (original version). For Clustered Dreambooth, Stage 1 test accuracies are notably high across datasets.

Dataset	Method	Stage 1		Stage 2	
		Worst	Average	Worst	Average
Waterbirds	FFR† [20]	45.3	71.3	69.5	84.0
	Vanilla SD [23]	65.4	79.9	74.6	80.5
	LoRA finetuning [5]	82.3	89.3	86.5	89.9
	Dreambooth [24]	85.7	89.5	89.3	90.1
	Clustered Dreambooth	87.4	89.5	88.1	90.2
CelebA	FFR† [20]	48.9	75.3	68.9	85.7
	Vanilla SD [23]	78.3	84.3	76.40	84.2
	LoRA finetuning [5]	82.1	87.1	82.3	87.2
	Dreambooth [24]	81.7	86.5	82.1	87.9
	Clustered Dreambooth	83.9	88.0	84.1	88.4
UTKFace	FFR† [20]	66.1	77.5	67.4	81.4
	Vanilla SD [23]	68.0	82.3	62.0	82.3
	LoRA finetuning [5]	57.0	82.0	68.6	85.6
	Dreambooth [24]	56.2	80.7	57.9	80.9
	Clustered Dreambooth	66.9	83.0	76.0	83.5

Figure 4. **Images generated from four different clusters of the CelebA groups.** We note the different characteristics followed by the images of a cluster for each group. We show two images per cluster and put the perceivable attributes seen in each cluster on the top (e.g., old age, young age, sportspersons and retro hairstyles for blond females).

CVPR
#26

Figure 5. **Waterbird on Land Images generated the vanilla Stable Diffusion 1.4.** We note that the model often fails to follow the prompt instruction, and many images contain water even when explicitly prohibiting it in the negative prompt.

Figure 6. **Female Children images generated by Clustered Dreambooth for the high bias variant of UTKFace**. We note that the model often generates grayscale images, out-of-domain images, and also irrelevant ones.

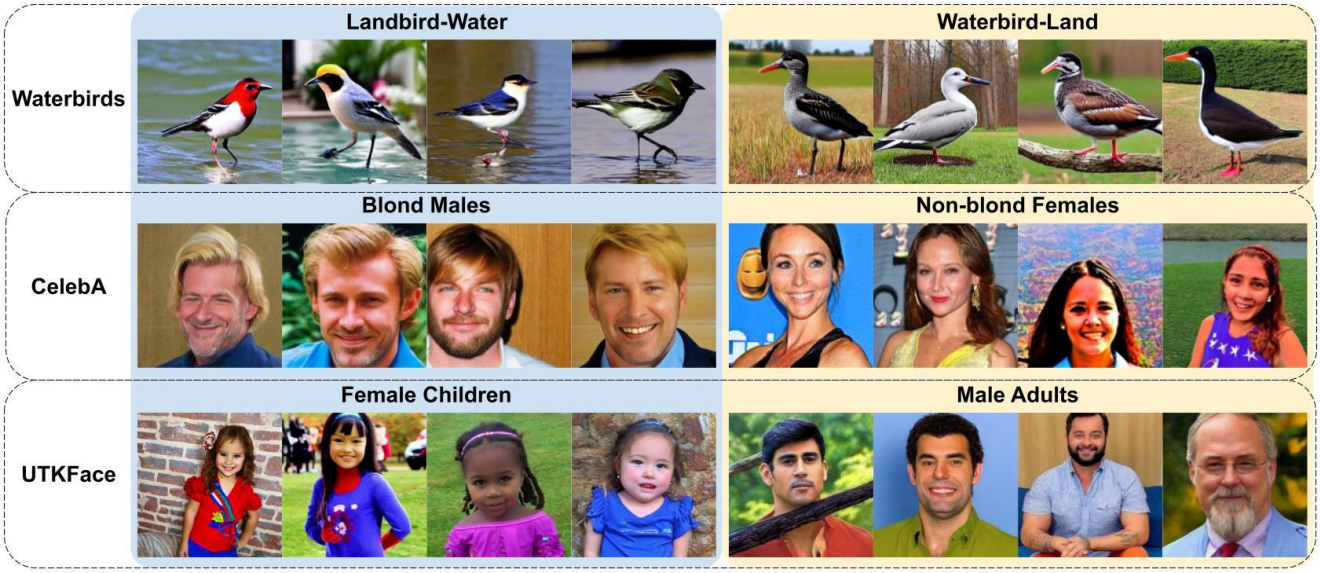


Figure 7. **Bias Conflicting samples generated for Bias Ratio$= 0.999$** using Clustered Dreambooth. Here we note that even in the absence of finetuned generative models for the bias-conflicting groups, the images generated from the models finetuned on the bias-aligned groups closely tend to follow the distribution of the dataset, while maintaining the requirements of the target group.

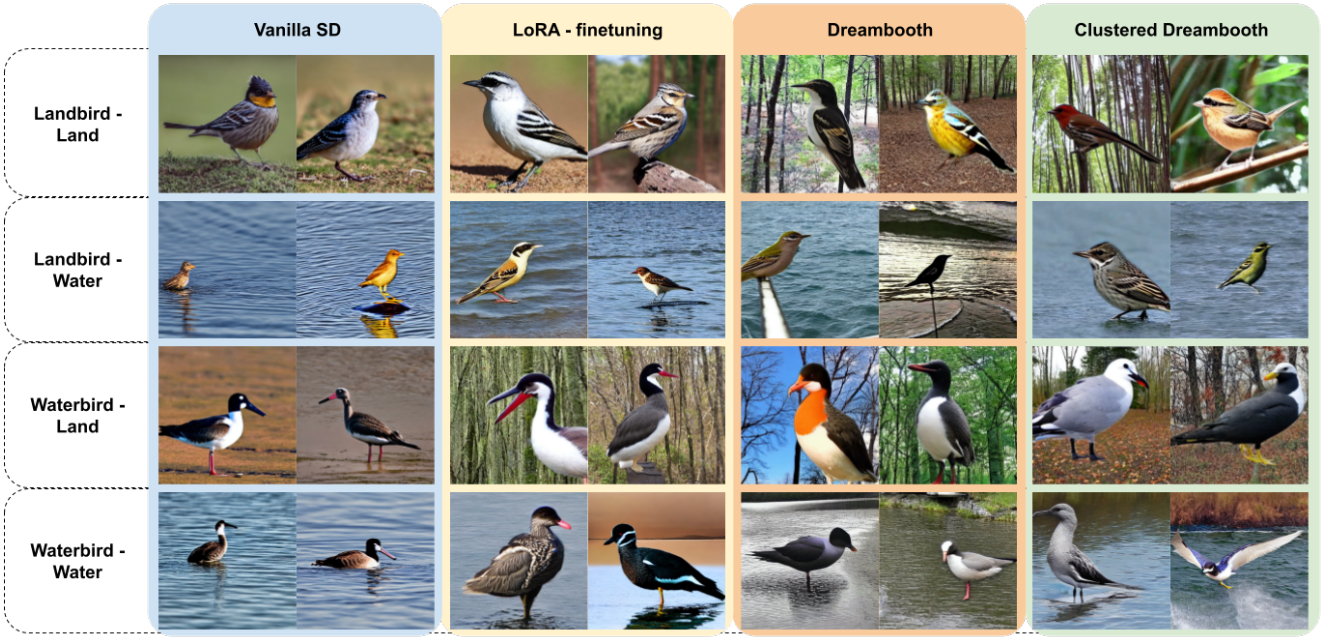


Figure 8. **Group-wise images for Waterbirds** generated by Vanilla SD, LoRA-finetuned SD, Dreambooth and Clustered Dreambooth.

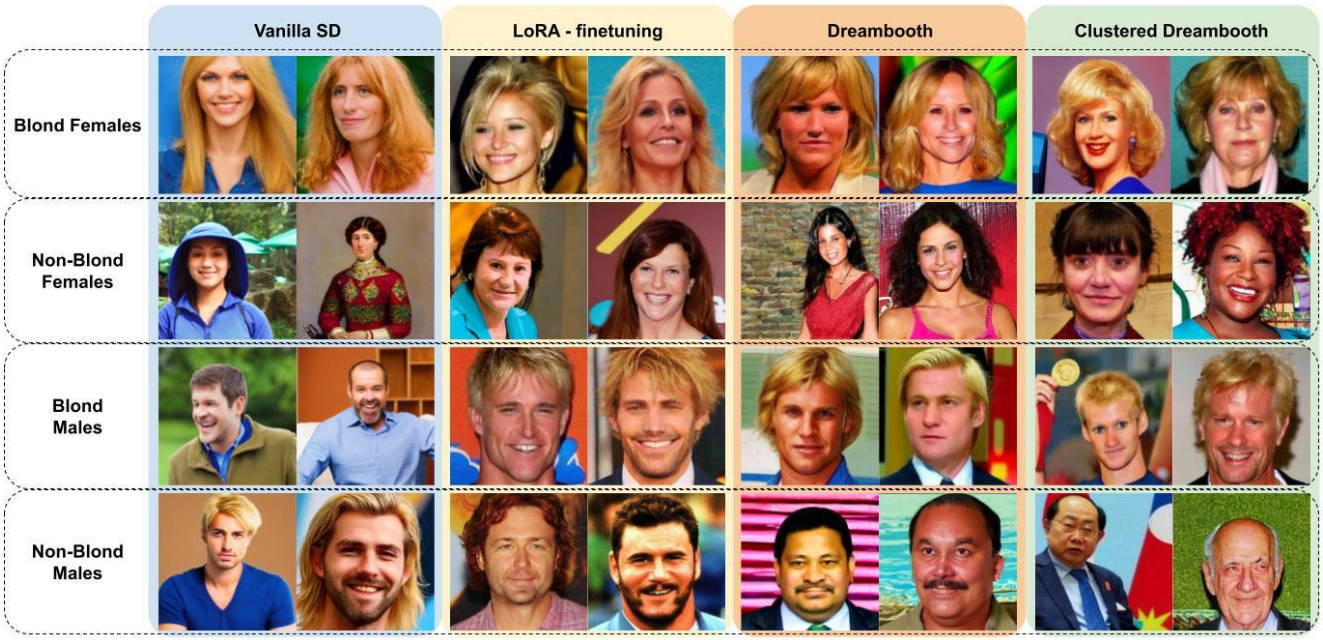


Figure 9. **Group-wise images for CelebA** generated by Vanilla SD, LoRA-finetuned SD, Dreambooth and Clustered Dreambooth.

Figure 10. **Group-wise images for UTKFace** generated by Vanilla SD, LoRA-finetuned SD, Dreambooth and Clustered Dreambooth.