AIM-Fair: Advancing Algorithmic Fairness via Selectively Fine-Tuning Biased Models with Contextual Synthetic Data

Supplementary Material

7. Addition Details of the AIM-Fair

Details for Instructing LLM. For the CelebA dataset [26], we use the following instruction as input for GPT-4, with all other attributes derived from the CelebA dataset:

Generate {Number} diverse text prompts for human face attributes that always include {Target Attribute} and {Protected Attribute}. Each prompt should also consider some of the following attributes: 5 o'clock shadow, arched eyebrows, attractive, bags under eyes, bald, bangs, big lips, big nose, black hair, blurry, brown hair, bushy eyebrows, chubby, double chin, eyeglasses, goatee, grey hair, heavy makeup, high cheekbones, mouth slightly open, moustache, narrow eyes, no beard, oval face, pale skin, pointy nose, receding hairline, rosy cheeks, sideburns, straight hair, wavy hair, wearing earrings, wearing a hat, wearing lipstick, wearing necklace, wearing necktie, and young. Additionally, each prompt should include a variety of head poses, such as slight tilts, turns, and different head orientations (e.g., head turned slightly left, tilted upward, facing slightly downward) to ensure diversity in the generated image angles. The prompts should be for a "Portrait face photo of a," ensuring the image only contains the head part.

For the UTKFace dataset [60], given the broad age distribution, we use the following instruction for GPT-4:

Generate {Number} diverse text prompts for human face attributes that always include {Target Attribute} and {Protected Attribute}. Include details about facial expressions, hairstyles, and any other distinguishing features that can help in generating a realistic image. And also contain age start from 1 and end at 100. The prompts should be for a "Portrait face photo of a," ensuring the image only contains the face part.

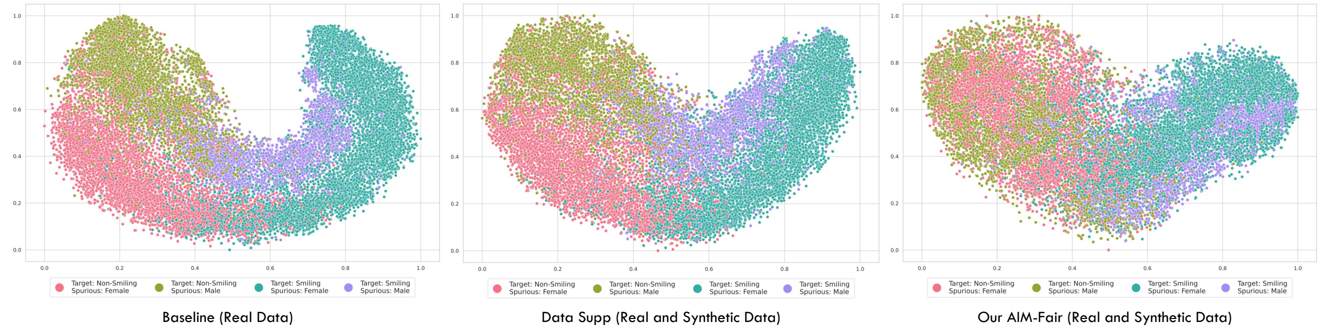
Algorithm for the Selective Fine-Tuning. The algorithm of the proposed selective fine-tuning method is depicted in Algorithm 1.

Algorithm 1 Pseudo Code for the Selective Fine-Tuning

```
Require: Pre-trained model $f_\theta(x)$; Unfair real dataset $(X_R, Y_R) \in \mathcal{D}_R$; Unfair synthetic dataset $(X_{S_1}, Y_{S_1}) \in \mathcal{D}_{S_1}$; Fair synthetic dataset $(X_{S_2}, Y_{S_2}) \in \mathcal{D}_{S_2}$; Top-$k$ value $k$.
Ensure: Fine-tuned model $f_{\theta^*}(x)$
  1. Compute Gradients:
  $\boldsymbol{g}_R \leftarrow \nabla_\theta \mathcal{L}(f_\theta(X_R), Y_R)$
  $\boldsymbol{g}_{S_1} \leftarrow \nabla_\theta \mathcal{L}(f_\theta(X_{S1}), Y_{S_1})$
  $\boldsymbol{g}_{S_2} \leftarrow \nabla_\theta \mathcal{L}(f_\theta(X_{S2}), Y_{S_2})$
  2. Calculate Gradient Differences:
  $\Delta_1 \leftarrow |\boldsymbol{g}_R - \boldsymbol{g}_{S_1}|, \quad \Delta_2 \leftarrow |\boldsymbol{g}_{S_1} - \boldsymbol{g}_{S_2}|$
  3. Select Parameters to Fine-Tune:
  $R_1 \leftarrow$ parameters sorted ascendingly by $\Delta_1$
  $R_2 \leftarrow$ parameters sorted descendingly by $\Delta_2$
  $K \leftarrow$ top-$k$ parameters in $R_1 \cap$ top-$k$ parameters in $R_2$
  4. Update Model Parameters:
  for each parameter $\theta_j$ do
      if $\theta_i \in K$ then
          Set $\theta_j$requires_grad $\leftarrow$ True
      else
          Set $\theta_j$.requires_grad $\leftarrow$ False
      end if
  end for
  5. Fine-tune the Model:
  Train $f_\theta(x)$ on $(X_{S2}, Y_{S2})$ using parameters in $K$
```

8. Additional Experimental Results

Ablation Analysis on ViT Model. We also applied our method to the ViT-32-Small model on CelebA. As shown in Tab. 8, our approach consistently outperforms other training methods, achieving the highest overall accuracy and the



Baseline (Real Data)

Data Supp (Real and Synthetic Data)

Our AIM-Fair (Real and Synthetic Data)

Figure 6. T-SNE visualizations for the learned representations on CelebA with target attribute ***Smiling*** and protected attribute ***Male***.

Table 8. Comparisons of varied training strategies with ViT-32-Small on CelebA.

Methods	T: Smiling; P: Male							T: Smiling; P: Young						
	Target	Protected		ACC (↑)	WST (↑)	EO (↓)	STD (↓)	Target	Protected		ACC (↑)	WST (↑)	EO (↓)	STD (↓)
		P=0	P=1						P=0	P=1				
Baseline	t=0	82.59	98.31	89.00	71.12	20.18	10.91	t=0	75.37	95.41	89.32	75.37	14.61	8.17
	t=1	95.76	71.12					t=1	94.86	85.69				
Trained on Synthetic Data	t=0	90.10	86.90	86.19	83.73	**2.32**	2.61	t=0	82.68	89.00	86.31	82.68	**3.53**	**2.27**
	t=1	83.73	83.85					t=1	85.01	85.16				
Data Supplementation	t=0	81.54	97.89	88.97	72.89	19.72	10.35	t=0	76.43	95.39	89.47	76.43	13.75	7.65
	t=1	95.99	72.89					t=1	94.43	85.90				
Fully Fine-Tuning	t=0	89.68	91.68	89.37	83.95	4.17	2.92	t=0	83.83	92.24	88.80	83.83	5.28	3.06
	t=1	90.01	83.95					t=1	88.86	86.85				
LoRA [16]	t=0	91.20	91.93	90.79	86.20	3.14	2.37	t=0	83.38	91.15	89.81	83.38	5.29	3.47
	t=1	91.76	86.20					t=1	92.41	89.61				
LTGC [61]	t=0	78.17	95.20	88.75	78.17	17.20	8.63	t=0	80.20	96.64	89.18	80.20	13.20	6.80
	t=1	96.82	79.44					t=1	93.09	83.13				
Selective Fine-Tuning	t=0	89.90	91.20	**90.89**	**87.79**	3.27	**1.85**	t=0	84.96	92.80	**90.30**	**84.96**	5.12	2.98
	t=1	92.84	87.79					t=1	91.45	89.05				

best worst-group accuracy. This demonstrates its generalization capability across different models. We also experimented by comparing our selective fine-tuning with LoRA [16] fine-tuning, and as shown in Tab. 8, our approach consistently outperforms LoRA on both datasets. We believe that LoRA is effective for parameter-efficient adaptation but restricts the model's ability to correct deeply embedded biases, while our method selectively fine-tunes the layers that contribute most to bias, ensuring targeted fairness improvements. Additionally, we compare our method with LTGC [61] which was trained with a mix-up of real and synthetic data for addressing the long-tail problem. As shown in the results in Tab. 9, while the mix-up method successfully addresses long-tail issues, it does not appear to enhance model fairness.

Evaluations of Different Number of Synthetic Data. We also evaluate fine-tuning with varying amounts of balanced synthetic data on UTKFace. Similar to the experiment on the CelebA, we use the real training data count as a reference and set different ratios to determine the number of synthetic data. As shown in Tab. 10, the results demonstrate the consistency with our observations on the CelebA.

9. Visualizations

Latent Visual Feature Distributions. To further illustrate how our method works, we provide visualizations of the learned representations using t-SNE in Fig. 6. We divide the CelebA test set into four groups based on target and spurious attributes. We observe that the baseline ERM, trained either solely on biased real data or on a mix of real and synthetic data, learns spurious correlations, resulting in representations that can be separated by the spurious attributes. In contrast, the representations learned by our AIM-Fair method contain less information on spurious correlations, thereby contributing to fairer classification.

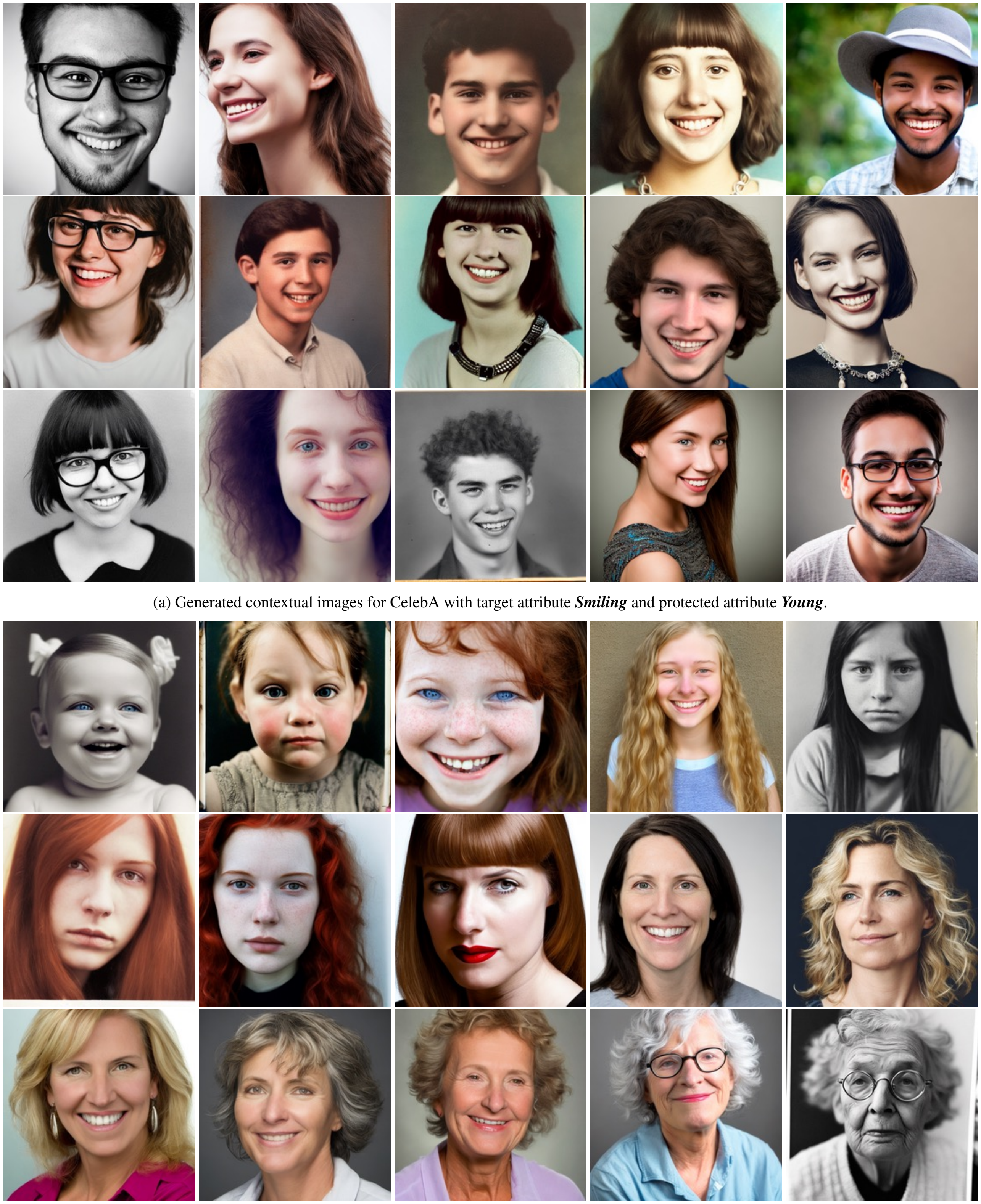
Table 9. Comparisons of varied training strategies with ResNet-18 on CelebA (T=Smiling, P=Male).

Methods	Target	Protected		ACC (↑)	WST(↑)	EO (↓)	STD (↓)
		P=0	P=1				
Balseline	t=0	83.57	98.50	**89.23**	71.52	23.84	10.65
	t=1	95.36	71.52				
LTGC [61]	t=0	79.48	97.93	89.05	75.42	21.27	10.03
	t=1	96.65	75.42				
AIM-Fair (Ours)	t=0	88.16	91.40	89.02	**84.20**	**6.07**	**2.74**
	t=1	90.25	84.20				

Table 10. Classification results on UTKFace with different numbers of synthetic data.

Ratio To Real Data	Target	Protected		ACC (↑)	WST (↑)	EO (↓)	STD (↓)
		P=0	P=1				
Baseline	t=0	79.58	96.16	88.86	79.58	16.59	7.26
	t=1	95.75	83.96				
0.5	t=0	82.61	89.35	88.91	82.61	6.74	3.78
	t=1	92.06	91.61				
1.0	t=0	84.26	89.08	88.30	**84.26**	**4.81**	**2.41**
	t=1	90.62	89.25				
1.5	t=0	83.70	89.15	88.23	83.70	5.45	2.65
	t=1	89.84	90.25				
2.0	t=0	83.04	89.76	**88.98**	83.04	6.73	3.51
	t=1	91.71	91.41				

More Generated Contextual Images. We provide additional generated contextual images with target attribute



(a) Generated contextual images for CelebA with target attribute ***Smiling*** and protected attribute ***Young***.

(b) Generated contextual images for UTKFace with target attribute ***Female*** and protected attribute ***White***.

Figure 7. Generated contextual images.

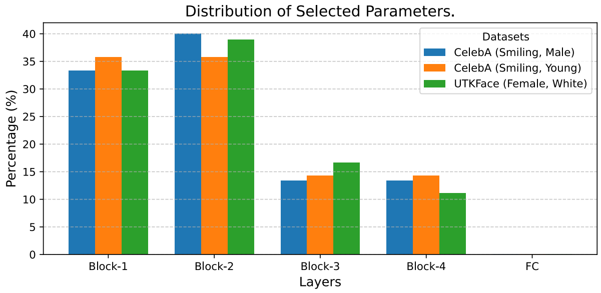


Figure 8. Distribution of selected parameters.

Table 11. Classification results on CelebA dataset (T=Male, P=Young) under different training strategies.

Methods	Target	Protected		ACC	WST	EO	STD
		P=0	P=1	(↑)	(↑)	(↓)	(↓)
Baseline	t=0	93.50	98.80	**97.16**	93.50	5.37	2.58
	t=1	98.98	93.99				
Fully Fine-Tuning	t=0	95.19	97.83	96.47	93.79	3.24	1.55
	t=1	96.88	93.79				
AIM-Fair (Ours)	t=0	94.41	97.34	96.00	**93.86**	**3.00**	**1.35**
	t=1	95.83	93.86				

Smiling and protected attribute *Young* and target attribute *Female* and protected attribute *White* in Fig. 7. Fig. 7a demonstrates that the generated images of smiling young individuals include diverse genders, hairstyles, hair colours, accessories, and head poses. Fig. 7b shows that the generated images of white females include a range of ages, facial expressions, hairstyles, hair colours, accessories, and head poses. This diversity in unmentioned attributes of the plain prompt helps mitigate bias related to these attributes.

Generated Mask Distribution. We analysed the distribution of the selected fine-tuned parameters across different layers. Our empirical results in Fig. 8 show that most of the selected parameters are located in early layers – this is consistent across different datasets.

10. Failures and Limitations

Our method still faces challenges in enhancing model fairness while retaining utility. As shown in Tab. 11, the baseline results indicate that the model exhibits only a small bias on the protected attribute. In this setting, our method improves model fairness, but it sacrifices accuracy by 1.16%. Additionally, compared to the fully fine-tuning method, our approach achieves better fairness but still results in lower accuracy. We argue that when the model achieves very high accuracy across all groups, it becomes challenging for our method to distinguish which parameters are sensitive to domain shift and which are sensitive to group shift. To address this limitation, further effort should be invested either in improving the synthetic data generation process or in designing strategies for faithful synthetic data selection.