ON FAIRNESS MEASUREMENT FOR GENERATIVE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep generative models have made significant progress in improving the diversity and quality of generated data. Recently, there has been increased interest in *fair* generative models. Fairness in generative models is important, as some bias in the sensitive attributes of the generated samples could have severe effects in applications under high-stakes settings (e.g., criminal justice, healthcare). In this work, we conduct, for the first time, an in-depth study on *fairness measurement*, a critical component to gauge the research progress of fair generative models. Our work makes two contributions. ***As our first contribution***, we reveal that there exist considerable errors in the existing fairness measurement framework. We attribute this to the lack of consideration for errors in the sensitive attribute classifiers. *Contrary to prior assumptions, even highly accurate attribute classifiers can result in large errors in fairness measurement, e.g.* a ResNet-18 for `Gender` with $\sim$97% accuracy could still lead to 4.98% estimation error when measuring the fairness of a StyleGAN2 trained on the CelebA-HQ. ***As our second (major) contribution***, we address this error in the existing fairness measurement framework by proposing a CLassifier Error-Aware Measurement (CLEAM). CLEAM applies a statistical model to take into account the error in the attribute classifiers, leading to significant improvement in the accuracy of fairness measurement. Our experimental results on evaluating fairness of state-of-the-art GANs (StyleGAN2 (Karras et al., 2019) and StyleSwin (Zhang et al., 2021)) show that ***CLEAM is able to significantly reduce fairness measurement errors, e.g. by 7.78% for StyleGAN2 (8.68%→0.90%), and by 7.16% for StyleSwin (8.23%→1.07%)*** when targeting the `Gender` attribute. Furthermore, our proposed CLEAM has minimal additional overhead when compared to the existing baseline. Code and instructions to reproduce the results are included in Supplementary.

1 INTRODUCTION

Recently, *fair generative models* have increasingly attracted more attention (Frankel & Vendrow, 2020; Choi et al., 2020; Humayun et al., 2022; Tan et al., 2020). In generative models, fairness is defined as *equal representation* (Hutchinson & Mitchell, 2019) of some *sensitive attributes*. For example, a fair generative model *w.r.t.* `Gender` has an equal probability of producing `Male` and `Female` samples. This is an important research area as any bias in generated data could limit the application efficacy of generative models. For instance, the super-resolution model PULSE (Menon et al., 2020) is found to exhibit biases *w.r.t.* `Race`, where it outputs faces with lighter skin and colored eyes, regardless of the input sample. Another example is the use of generative models for criminal Suspect Face Generation (Jalan et al., 2020). Fairness *w.r.t.* `Race` and `Gender` in this application is paramount to avoid inaccuracies *e.g.* racial profiling. Lastly, generative models are often used for supplementing scarce datasets for the training of classifiers, where data is often limited *e.g.* medical data, due to privacy concerns. Concerns arise when the bias within these supplementary data limits or hurts the performance of the model. For example, Larrazabal et al. (2020) demonstrates that in their disease diagnosis application on x-ray images, the classifier's training datasets must be fair *w.r.t.* `Gender`, and any bias degrades the performance of the classifier.

FAIRNESS MEASUREMENT. Recognizing the importance of fair generative models, several methods have been proposed to mitigate biases in generative models (Choi et al., 2020; Tan et al., 2020; Frankel & Vendrow, 2020). In this work, instead, we focus on ***fairness measurement*** of deep gen-

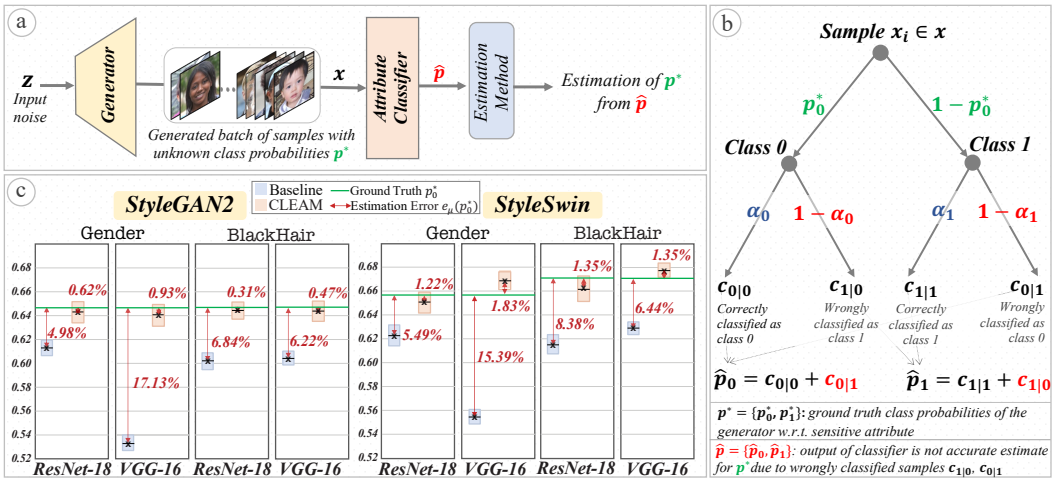


Figure 1: ⓐ An overview of the fairness measurement framework of a generative model. Since the output samples from the generator are not labeled, the ground-truth (GT) class probabilities *w.r.t.* a sensitive attribute (p^*) is unknown. The fairness measurement framework applies an attribute classifier to estimate p^*. A key component in the framework is the *estimation method*. In the existing framework (Baseline), the estimation method simply computes the mean value of the output of the attribute classifier ($\hat{p}$) to estimate p^*. Inaccuracies in the attribute classifier could result in inaccurate estimates for p^*. In our proposed CLEAM, we propose Alg. 1 as an estimation method to rectify the classifier output and estimate p^* more accurately. ⓑ In our proposed CLEAM, we model the classification error in the attribute classifier. See Sec. 3.2 for details. ⓒ Comparing errors in fairness measurements: Baseline (Choi et al., 2020; Frankel & Vendrow, 2020; Tan et al., 2020) and our proposed CLEAM, in estimating the GT value of p_0^* for SOTA StyleGAN2 (Karras et al., 2019) and StyleSwin (Zhang et al., 2021). We conducted experiments for a range of attribute classifiers, including ResNet-18 and VGG-16 as shown here for sensitive attributes `Gender` and `BlackHair`. Each panel in the figure shows the baseline and CLEAM estimates for a particular GAN on a certain sensitive attribute using an attribute classifier. We observe that fairness measurement errors in baseline are significant, while CLEAM substantially reduces the errors in all setups *e.g.* using ResNet-18 as the attribute classifier for `Gender`, error is reduced from 4.98% to 0.62%.

erative models, *i.e.* assessing and quantifying the bias of the generated data. Note that accurate measurement of fairness is critical to reliably gauge the progress of bias mitigation techniques. The general procedure of fairness measurement utilized in previous work to assess their proposed fair generators (Choi et al., 2020; Tan et al., 2020; Frankel & Vendrow, 2020) is shown in Fig. 1. Given a generator, first, a batch of samples is generated. These samples are then passed into an *attribute classifier*, which classifies each sample *w.r.t.* an attribute. In previous work, the results of the attribute classifier are directly taken as the estimation of bias. For example, if eight out of ten generated face images are classified as `Male`, then the generator is deemed biased at 0.8 towards `Male`. Then, the bias estimation could be fed to some fairness metric (*e.g.* L2-distance (Choi et al., 2020), or KL-divergance (Tan et al., 2020)) to report the fairness of the generator. *Critically, in existing work, the effect of* ***classification errors*** *in the attribute classifiers has not been studied.* In particular, classification errors in the attribute classifiers are possible, and they could affect bias estimation of the generator. Although, one may argue that the attribute classification could be of high accuracy, therefore the effect of classification error may not be significant. However, as there has not been any in-depth study on this topic, the impact of the classification error remains unclear, casting doubt on the results based on existing fairness measurements. Note that the *attribute classifier is indispensable for automated fairness measurement*.

In Our Work, we make two contributions to measuring fairness in generative models. As ***our first contribution***, we reveal that accuracy of the existing fairness measurement framework is not adequate, due to a lack of consideration for the attribute classifier inaccuracy. Importantly, our results suggest that using existing measurement framework, even in situations where the accuracy of the attribute classifier is high, the error in fairness measurement could be significant, *e.g.* ***using ResNet-***

18 trained on `Gender` ***with ∼97% average accuracy, can lead up to 4.98%, and 5.49% estimation error when measuring the fairness of StyleGAN2, and StyleSwin*** (Tab. 1). This undermines findings in Choi et al. (2020); Tan et al. (2020); Frankel & Vendrow (2020) which are based on existing fairness measurement frameworks, and do not account for the inaccuracy of the classifier.

To address this issue, ***as our second (major) contribution***, we propose CLassifier Error-Aware Measurement (CLEAM), a new method for bias estimation in the fairness measurement framework to obtain *point estimates* (PE) and *interval estimates* (IE) of biases of generated data. More specifically, in CLEAM, we model the classifier errors and use this model to rectify the errors and derive more accurate estimation of bias. We evaluate the accuracy of CLEAM on fairness measurement using state-of-the-art GANs (StyleGAN2 (Karras et al., 2019), and StyleSwin (Zhang et al., 2021)), which were trained on large public datasets. We then compare the performance of CLEAM against the existing framework for fairness measurements. To evaluate the performance of the proposed method under varying degrees of bias, we further design a controlled set of experiments using *pseudo-generators*. Our experimental results show that CLEAM is accurate for fairness measurement for both real generators and pseudo-generators (see Sec. 5). We remark that CLEAM is not a new fairness metric, but an improved estimation of bias in the fairness measurement framework that could achieve more accurate bias estimation for use in various fairness metrics.

One major challenge in analyzing the accuracy of fairness measurement is that generated data are usually unlabeled *w.r.t.* sensitive attributes. We overcome this challenge by manually labeling samples output from two SOTA GANs (StyleGAN2 and StyleSwin) *w.r.t.* different attributes, resulting in a new dataset with ∼9K samples for each GAN. These new datasets are utilized in our work to evaluate the performance of existing fairness measurement framework and our proposed CLEAM. We remark that these manually-labelled datasets are used solely in evaluation and are not used in our proposed CLEAM.

2 Fairness Measurement Framework

Fig. 1(a) illustrates fairness measurement framework for generative models as in Choi et al. (2020); Frankel & Vendrow (2020); Tan et al. (2020). Assume that, using some noise vector $\mathbf{z} \sim q_{\mathbf{z}}$ as input, a generative model G_θ synthesizes a sample $\mathbf{x}_i \sim q_\theta$. Generally, as synthesized samples are not labeled by the generator, the ground truth (GT) class probability of these samples *w.r.t.* a sensitive attribute (denoted by p^*) is unknown. Therefore, an attribute classifier $C_{\mathbf{u}}$ is utilized to estimate p^*. In particular, for each sample $\mathbf{x}_i \in \mathbf{x}$, $C_{\mathbf{u}}(\mathbf{x}_i) = Pr(\mathbf{u}|\mathbf{x}_i)$, where $\mathbf{u}$ is a one-hot vector representation of the attribute, and $Pr(\mathbf{u}|\mathbf{x}_i)$ is the argmax classification. In existing work, the expected value of the classifier output over a batch of samples, $\hat{p} = \mathbb{E}_{\mathbf{x}_i \sim q_\theta} C_{\mathbf{u}}(\mathbf{x}_i)$ (or the average of $\hat{p}$ over multiple batches of samples), is used as an estimation of p^*. This estimate may then be used in some fairness metric f to report the fairness value for the generator G_θ, *e.g.* fairness discrepancy metric between $\hat{p}$ and a uniform distribution $\bar{p}$ (Choi et al., 2020; Teo & Cheung, 2021) (see Supp. A.3 for some fairness metrics and their details). Note that ***the general assumption behind existing framework is that with a reasonably accurate attribute classifier, $\hat{\mathbf{p}}$ could be an accurate estimation of $\mathbf{p}^*$***. In the next section, we will present a deeper analysis of the effects of classifier inaccuracy on fairness measurement, which suggests that even for highly accurate classifiers, the measurement error could be rather significant.

Note that, one may argue that conditional GANs (CGANs) (Mirza & Osindero, 2014; Odena et al., 2017) may be used to generate samples conditioning on the attributes and eliminate the need for an attribute classifier. However, CGANs are not considered in previous works for fair generative models due to several limitations. These include the lack of availability of a large dataset (with attribute labelled) for training, the unreliability of output attribute labels and sample quality (Thekumparampil et al., 2018), and exponentially increasing conditional terms by increasing numbers of attributes.

3 A Closer Look at Fairness Measurement

In this section, we take a closer look at the existing fairness measurement framework by examining its performance in estimating p^* of the samples generated by SOTA GANs. Our experimental results reveal that the estimation errors of the current fairness measurement framework could be significant. Then, we develop a statistical model for the attribute classifier output (Fig. 1(b)) to help us better un-

Table 1: Comparing the **point estimates** of Baseline (Choi et al., 2020), Diversity (Keswani & Celis, 2021) and our proposed CLEAM measurement framework in estimating p^* of datasets sampled from (A) StyleGAN2 (Karras et al., 2019) and (B) StyleSwin (Zhang et al., 2021). The p_0^* value for each GAN with a certain attribute is determined by manually hand-labeling the generated data (see Supp F). We utilize four different classifier Resnet-18/34 (He et al., 2016), MobileNetv2 (Sandler et al., 2018) and VGG-16 (Simonyan & Zisserman, 2014), with different accuracy $\boldsymbol{\alpha}$, to classify attributes `Gender` and `BlackHair`, to obtain $\hat{p}$. Each $\hat{p}$ utilizes $n = 400$ samples and is evaluated for a batch-size of $s = 30$. We repeat this for 5 experimental runs and report the mean error rate, per Eqn. 1. More analysis with different n and s are included in Supp D.

(A) StyleGAN2							
Classifier	$\boldsymbol{\alpha} = \{\alpha_0, \alpha_1\}$	**Baseline**(Choi et al., 2020)		**Diversity**(Keswani & Celis, 2021)		**CLEAM (Ours)**	
		Estimate $\mu_{\text{Base}}(\hat{p}_0)$	Error $e_\mu(p_0^*)(\downarrow)$	Estimate $\mu_{\text{Div}}(\hat{p}_0)$	Error $e_\mu(p_0^*)(\downarrow)$	Estimate $\mu_{\text{CLEAM}}(\hat{p}_0)$	Error $e_\mu(p_0^*)(\downarrow)$
`Gender` with GT class probability $\boldsymbol{p_0^*}$**=0.642**							
ResNet-18	{0.947, 0.983}	0.610	4.98%	—	—	0.638	**0.62%**
ResNet-34	{0.932, 0.976]}	0.596	7.17%	—	—	0.634	**1.25%**
MobileNetv2	{0.938, 0.975}	0.607	5.45%	—	—	0.637	**0.78**%
VGG-16	{0.801, 0.919}	0.532	17.13%	0.550	14.3%	0.636	**0.93**%
		Avg Error	8.68%	Avg Error	14.30%	Avg Error	**0.90**%
`BlackHair` with GT class probability $\boldsymbol{p_0^*}$**=0.643**							
ResNet-18	{0.869, 0.885}	0.599	6.84%	—	—	0.641	**0.31%**
ResNet-34	{0.834, 0.916}	0.566	11.98%	—	—	0.644	**0.16**%
MobileNetv2	{0.839, 0.881}	0.579	9.95%	—	—	0.639	**0.62**%
VGG-16	{0.851, 0.836}	0.603	6.22%	0.582	9.49%	0.640	**0.47**%
		Avg Error	8.75%	Avg Error	9.49%	Avg Error	**0.39**%
(B) StyleSwin							
`Gender` with GT class probability $\boldsymbol{p_0^*}$**=0.656**							
ResNet-18	{0.947, 0.983}	0.620	5.49%	—	—	0.648	**1.22%**
ResNet-34	{0.932, 0.976}	0.610	7.01%	—	—	0.649	**1.07**%
MobileNetv2	{0.938, 0.975}	0.623	5.03%	—	—	0.655	**0.15**%
VGG-16	{0.801, 0.919}	0.555	15.39%	0.562	14.33%	0.668	**1.83**%
		Avg Error	8.23%	Avg Error	14.33%	Avg Error	**1.07**%
`BlackHair` with GT class probability $\boldsymbol{p_0^*}$**=0.668**							
ResNet-18	{0.869, 0.885}	0.612	8.38%	—	—	0.659	**1.35%**
ResNet-34	{0.834, 0.916}	0.581	13.02%	—	—	0.662	**0.90**%
MobileNetv2	{0.839, 0.881}	0.596	10.78%	—	—	0.659	**1.35**%
VGG-16	{0.851, 0.836}	0.625	6.44%	0.608	8.98%	0.677	**1.35**%
		Avg Error	9.66%	Avg Error	8.98%	Avg Error	**1.24**%

derstand the relation between inaccuracy in classifier and error in estimation. Subsequently, in Sec. 4, we will also apply this statistical model when proposing CLEAM for better fairness measurement.

3.1 Measurement Error in Existing Framework

A critical problem with the existing measurement framework, called **baseline**, is that there could be ***considerable discrepancies between measured $\hat{p}$ and p^**** (Fig. 1) even when the accuracy of the classifier is considerably high. We refer to this discrepancy as estimation error or measurement error, interchangeably. Here, we design an experiment to demonstrate these errors when evaluating bias in SOTA GANs. Following previous work (Choi et al., 2020), our main focus is on binary attributes which take values in $\{0, 1\}$. Note that, we assume that the accuracy of the attribute classifier $C_{\mathbf{u}}$ is known and is characterized by the vector $\boldsymbol{\alpha} = \{\alpha_0, \alpha_1\}$, where $\alpha_i, i \in \{0, 1\}$ is the probability of correctly classifying a sample with true attribute label i. In practice, $\boldsymbol{\alpha}$ can be measured during the validation stage of $C_{\mathbf{u}}$. Also, note that p^* can be assumed to be a constant vector, given that the samples generated by G_θ can be considered to come from an infinite population, as theoretically there is no limit on the number of samples generated by a generative model *e.g.* GAN.

Experimental Setup. To evaluate the existing fairness measurement framework, we utilize StyleGAN2 (Karras et al., 2019) and StyleSwin (Zhang et al., 2021) pretrained on CelebA-HQ

Figure 2: **Comparing the empirical results against our proposed statistical model.** We randomly sample s=400 batch of $n = 400$ samples from our labeled StyleGAN2 dataset and input them into a ResNet-18 (Top) and VGG-16 model (Bottom), with known $\boldsymbol{\alpha}$, per Tab. 1, to evaluate $\hat{p}$ on Gender. Then with the p^* and $\boldsymbol{\alpha}$, we evaluate the statistical distribution with Eqn. 5 and 6. Notice that the statistical model is a very good approximation of the empirical results.

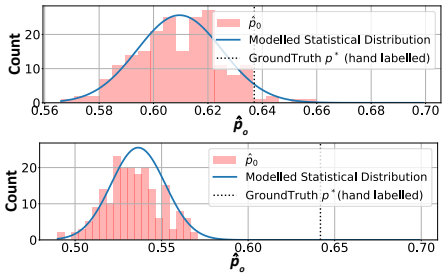


Table 2: Re-evaluating the **point estimates** of previously proposed bias mitigation method, importance-weighting (imp-weighting) (Choi et al., 2020) with CLEAM. We first evaluate the bias of a BIGGAN Brock et al. (2019) with and without implementing imp-weighting *i.e.* unweighted and weighted, with the Baseline. Then, we apply CLEAM to obtain a more accurate measurements, and compare it against the Baseline. We do this for both Gender and BlackHair attributes.

Test	Baseline	Diversity	CLEAM (Ours)
	$\mu_{\text{Base}}(\hat{p}_0)$	$\mu_{\text{Div}}(\hat{p}_0)$	$\mu_{\text{CLEAM}}(\hat{p}_0)$
$\boldsymbol{\alpha}$=[0.976,0.979], Gender			
Unweighted	0.727	0.711	0.738
Weighted	0.680	0.671	0.690
$\boldsymbol{\alpha}$=[0.881,0.887], BlackHair			
Unweighted	0.729	0.716	0.803
Weighted	0.716	0.706	0.785

(Lee et al., 2020) for sample generation. Note that the major limitation of evaluating the existing fairness measurement framework is that p^* is not available. Therefore, ***to pave the way for an accurate evaluation, we create a new dataset by manually labeling the GAN-generated samples***. Specifically, we utilize Amazon Mechanical Turks to hand-label the samples *w.r.t.* Gender and BlackHair, resulting in ∼9K samples for each GAN, see Supp. F for details. These labeled samples provide the p^* for each GAN, which are then compared with the estimated baseline ($\hat{p}$).

To calculate each $\hat{p}$ value, a batch of $n = 400$ samples are randomly drawn from the created dataset and passed into $C_{\mathbf{u}}$ for attribute classification. We repeat this for $s = 30$ batches. We report the mean results denoted by μ_{Base} and the 95% confidence interval denoted by ρ_{Base}. For comprehensive analysis, we repeat the experiment with four different attribute classifiers, namely Resnet-18, ResNet-34 (He et al., 2016), MobileNetv2 (Sandler et al., 2018), and VGG-16 (Simonyan & Zisserman, 2014). See Supp. D for training details. As seen in Tab. 1, all the classifiers demonstrate reasonably high accuracy with average accuracy $\in [84\%, 96\%]$. Note that as we focus on binary attributes (*e.g.* Gender:{Male, Female}), both p^* and $\hat{p}$ have two components *i.e.* $p^* = \{p_0^*, p_1^*\}$, and $\hat{p} = \{\hat{p}_0, \hat{p}_1\}$. After computing the μ_{Base} and ρ_{Base}, we calculate *normalized L1 point error* e_μ, and *interval max error* e_ρ *w.r.t.* the p_0^* (as GT value) to evaluate the measurement accuracy of the baseline method:

$$e_{\mu_{\text{Base}}} = |p_0^* - \mu_{\text{Base}}(\hat{p}_0)|/p_0^* \tag{1}$$

$$e_{\rho_{\text{Base}}} = max(|min(\rho_{\text{Base}}(\hat{p}_0)) - p_0^*|, |max(\rho_{\text{Base}}(\hat{p}_0)) - p_0^*|)/p_0^* \tag{2}$$

A smaller value of $e_{\mu_{\text{Base}}}$, and $e_{\rho_{\text{Base}}}$ indicates a better estimate of p^*. Tab. 1 tabulates the predicted value of μ_{Base} by baseline, and the point error *w.r.t.* the GT value of p_0^* for attributes Gender, and BalckHair. Note that as $p_1^* = 1 - p_0^*$, similar behaviour is observed for p_1^*.

From our results in Tab. 1, the GT value of p_0^* (obtained via manual labeling of generated samples) shows that there is a considerable amount of bias in both StyleGAN2 and StyleSwin *w.r.t.* Gender, and BalckHair. Note that for a fair GAN we have $p_0^* = p_1^* = 0.5$. Then, even though the utilized attribute classifiers are reasonably accurate, we observe significantly large estimation errors in the current fairness measurement framework, *i.e.* $e_{\mu_{\text{Base}}}$ in the range of 4.98% to 17.13%. In particular, looking at the highest accuracy classifier (ResNet-18 on Gender attribute) with average accuracy $\sim$ 97%, we observe significant discrepancies between GT p^* and μ_{Base}, with $e_{\mu_{\text{Base}}} = 4.98\%$. These errors generally worsen as accuracy marginally degrades *e.g.* using ResNet-34 with accuracy $\approx$ 95% results in $e_{\mu_{\text{Base}}} = 7.17\%$, suggesting that there is a direct relationship between estimation error and classifier inaccuracy. See Supp. Tab. 9 for ρ_{Base} results. These considerably large errors contradict prior assumptions – that for a reasonably accurate classifier, we can assume $e_{\mu_{\text{Base}}}$ to be fairly negligible. This finding is particularly concerning, as *in some cases improvements by some*

bias mitigation techniques e.g. imp-reweighting Choi et al. (2020), can be as small as $\approx 2\%$, thereby potentially making the comparison between bias mitigation techniques inaccurate. After observing a large measurement error for the current framework, in what follows, we propose a statistical model for the classifier output to get a better understanding of the relationship between measurement error and classifier inaccuracy. We will also use this statistical model later when developing CLEAM for improving fairness measurement.

3.2 Proposed Statistical Model for Classifier Output

As shown in Fig. 1(a), to measure the fairness of G_θ, we feed n samples generated by G_θ to the attribute classifier $C_{\mathbf{u}}$. The output of the attribute classifier ($\hat{p}$) is in fact a random variable that aims to approximate the p^* (GT value of class probabilities *w.r.t.* attributes). Here, we propose a statistical model to derive the distribution of $\hat{p}$.

As Fig. 1(b) demonstrates, in our running example of a binary attribute, each generated sample is from *Class 0* with probability p_0^*, or from *Class 1* with probability p_1^*. Then generated sample from *Class i*, $i \in \{0, 1\}$, will be classified correctly with the probability of α_i, and wrongly with the probability of $\alpha_i' = 1 - \alpha_i$. So, for each sample, there are four mutually exclusive possible outputs $\mathbf{c}$ with the corresponding probability vector $\mathbf{p}$:

$$\mathbf{c}^T = \begin{bmatrix} c_{0|0} & c_{1|0} & c_{1|1} & c_{0|1} \end{bmatrix} \quad , \quad \mathbf{p}^T = [p_0^*\alpha_0 \quad p_0^*\alpha_0' \quad p_1^*\alpha_1 \quad p_1^*\alpha_1'] \tag{3}$$

where $c_{i|j}$ denotes the event of assigning label i to a sample with GT label j. This process is performed independently for each of the n generated samples. Denoting the probability of counts for each of these possible outputs after n trials as $N_{\mathbf{c}}$, we can model it as a multinomial distribution, $N_{\mathbf{c}} \sim Multi(n, \mathbf{p})$ (Rao, 1957; Kesten & Morse, 1959). Since $\mathbf{p}$ is not near the boundary of the parameter space, for a large n, the $Multi(n, \mathbf{p})$ can be approximated by a multivariate Gaussian distribution, $N_{\mathbf{c}} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with $\boldsymbol{\mu} = n\mathbf{p}$ and $\boldsymbol{\Sigma} = n\mathbf{M}$ (Geyer, 2010), where $\mathbf{M}$ is defined as:

$$\mathbf{M} = diag(\mathbf{p}) - \mathbf{p}\mathbf{p}^T \tag{4}$$

and $diag(\mathbf{p})$ denotes a square diagonal matrix corresponding to vector $\mathbf{p}$ (see Supp. A.1 for expanded form). The marginal distribution of this multivariate Gaussian distribution gives us the distribution for the count of each output in Eqn. 3. For example, the distribution of the count for event $c_{0|0}$, denoted by $N_{c_{0|0}}$, can be modeled as $N_{c_{0|0}} \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11})$. After calculating the distribution of $N_{c_{i|j}}$, we can find the total rate of data points labeled as class i in the n trials using normalized sum of the related random variables $\hat{p}_i = \frac{1}{n}\sum_j N_{c_{i|j}}$. For our binary example, the distribution of $\hat{p}_i$ can be calculated by summing two dependent random variables with Gaussian distribution. Therefore, $\hat{p}_0 \sim \mathcal{N}(\tilde{\mu}_{\hat{p}_0}, \tilde{\sigma}^2_{\hat{p}_0})$, where:

$$\tilde{\mu}_{\hat{p}_0} = \boldsymbol{\mu}_1 + \boldsymbol{\mu}_4 = p_0^*\alpha_0 + p_1^*\alpha_1' \tag{5}$$

$$\begin{aligned} \tilde{\sigma}^2_{\hat{p}_0} &= \boldsymbol{\Sigma}_{11} + \boldsymbol{\Sigma}_{44} + 2\boldsymbol{\Sigma}_{14} \\ &= \tfrac{1}{n}[(p_0^*\alpha_0 - (p_0^*\alpha_0)^2) + (p_1^*\alpha_1' - (p_1^*\alpha_1')^2)] + \tfrac{2}{n}p_0^*p_1^*\alpha_0\alpha_1' \end{aligned} \tag{6}$$

Similarly $\hat{p}_1 \sim \mathcal{N}(\tilde{\mu}_{\hat{p}_1}, \tilde{\sigma}^2_{\hat{p}_1})$ with $\tilde{\mu}_{\hat{p}_1} = \boldsymbol{\mu}_2 + \boldsymbol{\mu}_3$, and $\tilde{\sigma}^2_{\hat{p}_1} = \boldsymbol{\Sigma}_{22} + \boldsymbol{\Sigma}_{33} + 2\boldsymbol{\Sigma}_{23}$ which is aligned with the fact that $\hat{p}_1 = 1 - \hat{p}_0$.

To validate our statistical model, we use Eqn. 5 and 6 to computed statistical distribution and compare it against the empirical results. Our results in Fig. 2 show that the statistical model correlates well with the empirical results on both ResNet-18 and VGG-16, on the `Gender` attribute. Please see Supp. B.2 for more details and comprehensive results on validating the statistical model. Note that based on this model, for a perfect classifier ($\alpha_i = 1$, *i.e.* accuracy $= 100\%$), the $\tilde{\mu}_{\hat{p}_0}$ (and therefore μ_{Base}) converges to GT value of p_0^*, resulting in $e_{\mu_{\text{Base}}} \to 0$. This suggests that *the major reason for fairness measurement error is classifier inaccuracy*.

In summary, in this section, we showed that there is a significant error in the current framework for fairness measurement. This error stems from attribute classifier inaccuracy and exists along different attribute classifiers (even highly accurate ones with accuracy $\approx 97\%$) measuring various attributes of generative models, and it can prevent proper evaluation and comparison of generative models. Since training a perfect attribute classifier is not practical due to several reasons such as lack of an appropriate dataset and task hardness, in the next section, we propose a simple method that compensates for classifier inaccuracy and mitigates the fairness measurement error.

Algorithm 1: Computing point and interval estimates using CLEAM.

Require: accuracy of attribute classifier $\boldsymbol{\alpha}$.

Compute classifier output $\hat{p} : \{\hat{p}^1, \ldots, \hat{p}^s\}$ for s batches of generated data.

Compute sample mean $\ddot{\mu}_{\hat{p}}$ and sample variance $\ddot{\sigma}^2_{\hat{p}}$ using (7) and (8).

Use (9) to compute point estimate $\mu_{\texttt{CLEAM}}$.

Use (11) to compute interval estimate $\rho_{\texttt{CLEAM}}$.

4 CLEAM FOR ACCURATE FAIRNESS MEASUREMENT

In this section, we propose a new estimation method in fairness measurement that considers the inaccuracy of the attribute classifier. For this, we use the statistical model, introduced in Sec 3.2, to compute a more accurate estimation of p^* using attribute classifier output, $\hat{p}$. Specifically, we first propose a Point Estimate (PE) by approximating the maximum likelihood value of p^*. Then, using the confidence interval for the mean value of observed data, we propose an Interval Estimate (IE) for p^*.

POINT ESTIMATE FOR p^*. Suppose that we have access to s samples of $\hat{p}$ denoted by $\{\hat{p}^1, \ldots, \hat{p}^s\}$, *i.e.* attribute classification results on s batches of generated data. We can then use the proposed statistical model to approximate the p^*. In the previous section, we showed that we can model $\hat{p}^i_j$ using a Gaussian distribution. Considering this, first, we use the available samples to calculate sample-based statistics including the mean and variance of the $\hat{p}$ samples, as follows:

$$\ddot{\mu}_{\hat{p}_j} = \frac{1}{s}\sum_{i=1}^{s} \hat{p}^i_j \tag{7}$$

$$\ddot{\sigma}^2_{\hat{p}_j} = \frac{1}{s}\sum_{i=1}^{s} (\hat{p}^i_j - \ddot{\mu}_{\hat{p}_j})^2 \tag{8}$$

In a Gaussian distribution, the Maximum Likelihood Estimate (MLE) of the population mean is its sample mean $\ddot{\mu}_{\hat{p}_j}$. Given that s is large enough (*e.g.* $s = 30$), we can assume that $\ddot{\mu}_{\hat{p}_j}$ is a good approximation of the population mean, and equate it to the statistical population mean $\tilde{\mu}_{\hat{p}_j}$ in Eqn. 5. With that, we get the maximum likelihood approximation of p^*, which we call the CLEAM's point estimate $\mu_{\texttt{CLEAM}}$ (more details in Supp. A.2):

$$\mu_{\texttt{CLEAM}}(p^*_0) = \frac{\ddot{\mu}_{\hat{p}_0} - \alpha'_1}{\alpha_0 - \alpha'_1} \quad , \quad \mu_{\texttt{CLEAM}}(p^*_1) = 1 - \mu_{\texttt{CLEAM}}(p^*_0) \tag{9}$$

INTERVAL ESTIMATE FOR p^*. In the previous part, we proposed a PE for p^* using the statistical model, and sample-based mean $\ddot{\mu}_{\hat{p}_0}$. Since we use only s samples of $\hat{p}$, the $\ddot{\mu}_{\hat{p}_0}$ may not capture the exact value of the population mean, which we approximate as $\tilde{\mu}_{\hat{p}_0}$. This adds some degree of inaccuracy into $\mu_{\texttt{CLEAM}}$. In fact, in our framework, $\ddot{\mu}_{\hat{p}_0}$ equals $\tilde{\mu}_{\hat{p}_0}$ when $s \to \infty$. One solution for a more accurate estimation of $\tilde{\mu}_{\hat{p}_0}$ is to use a large s (see Supp. B.1). However, this increases the computational complexity, as each $\hat{p}$ requires n data samples, generated by G_θ. As discussed earlier, $\hat{p}_0$ follows a Gaussian distribution, hence as an alternative solution, to have a better estimation for p^*, we find an approximated 95% confidence interval (CI) for $\tilde{\mu}_{\hat{p}_0}$:

$$\ddot{\mu}_{\hat{p}_0} - 1.96\frac{\ddot{\sigma}_{\hat{p}_0}}{\sqrt{s}} \leq \tilde{\mu}_{\hat{p}_0} \leq \ddot{\mu}_{\hat{p}_0} + 1.96\frac{\ddot{\sigma}_{\hat{p}_0}}{\sqrt{s}} \tag{10}$$

Applying Eqn. 5 to Eqn. 10 gives the lower and upper bounds of approximated 95% CI for p^*_0:

$$\mathcal{L}(p^*_0), \mathcal{U}(p^*_0) = (\ddot{\mu}_{\hat{p}_0} \mp 1.96(\ddot{\sigma}_{\hat{p}_0}/\sqrt{s}) - \alpha'_1)/(\alpha_0 - \alpha'_1) \tag{11}$$

This gives us the CLEAM's interval estimate, $\rho_{\texttt{CLEAM}}(p^*_0) = [\mathcal{L}(p^*_0), \mathcal{U}(p^*_0)]$, a range of values that we can be approximately 95% confident to contain the true p^*_0. The range of possible values for p^*_1 can be simply derived considering $p^*_1 = 1 - p^*_0$. The overall procedure of CLEAM for calculating the point and interval estimates is summarized in Alg. 1.

5 EXPERIMENTS

In this section, we evaluate the performance of CLEAM in fairness measurement. First, in Sec. 5.1, we evaluate CLEAM on measuring the fairness of the SOTA GANs. The results show that CLEAM is able to more accurately approximate p^* compared to previously proposed methods. Then, in Sec. 5.2, we apply CLEAM to re-evaluate the performance of previously proposed bias mitigation algorithms, where the results suggest that the bias reported previously may have been underestimated. Finally, in Sec. 5.3, we use a *pseudo-generator* to vary the degree of bias *i.e.* p^* and demonstrate CLEAM's performance in an extended range of bias.

To the best of our knowledge, there is no similar work in the literature for improving fairness measurements in generative models. Therefore, we compare **CLEAM** with the two most related works: a) the **Baseline** used in previous works (Choi et al., 2020; Tan et al., 2020; Frankel & Vendrow, 2020) b) **Diversity** (Keswani & Celis, 2021) which computes disparity within a dataset. Unless specified otherwise, we repeat the experiments with $s = 30$ batches of images from the generators, and we use $n = 400$ samples in each batch. For a fair comparison, all three algorithms use the exact same inputs. However, while Baseline and Diversity ignore inaccuracy in the attribute classifier, CLEAM makes good use of it to rectify the error made by the classifier. We repeat each experiment 5 times and report the mean value for each test point. Additional results and analysis are in the Supp.

5.1 CLEAM FOR FAIRNESS MEASUREMENT OF GENERATORS

In this setup, we estimate the bias of SOTA GAN with CLEAM and compare with GT obtained via manual labeling. To fairly compare the different methods, we first compute s samples of $\hat{p}$, one for each batch of n images. For Baseline, we use the mean $\hat{p}$ value as the PE (denoted by $\mu_{\texttt{Base}}$), and the 95% confidence interval as IE ($\rho_{\texttt{Base}}$). With the same s samples of $\hat{p}$, we apply Alg. 1 to obtain $\mu_{\texttt{CLEAM}}$ and $\rho_{\texttt{CLEAM}}$. Next, for Diversity, following the original source code, a controlled dataset with uniform sensitive attribute representation is randomly selected from a held-out dataset of CelebA-HQ (Lee et al., 2020). Then, as per Keswani & Celis (2021), we use a VGG-16 (Simonyan & Zisserman, 2014) as a feature extractor and compute Diversity δ. With δ we find $\hat{p}_0 = (\delta + 1)/2$ and subsequently $\mu_{\texttt{Div}}$ and $\rho_{\texttt{Div}}$ from the mean and 95% CI (see Supp C.2 for more details). Then, we compute $e_{\mu_{\texttt{CLEAM}}}(p_0^*)$ and $e_{\mu_{\texttt{Div}}}(p_0^*)$ with Eqn 1, and $e_{\rho_{\texttt{CLEAM}}}(p_0^*)$ and $e_{\rho_{\texttt{Div}}}(p_0^*)$ with Eqn 2, by replacing the Baseline estimates with CLEAM and Diversity respectively.

As discussed earlier, our results in Tab. 1, show that the baseline experiences significantly large errors *i.e.* $4.98\% \leq e_{\mu_{\texttt{Base}}} \leq 17.13\%$, due to a lack of consideration for the classifier inaccuracies. We note that this problem is prevalent throughout classifier architectures, even at higher capacity classifiers *e.g.* ResNet-34. Diversity, a method similarly unaware of the inaccuracies of the classifier, presents a similar issue with $9.49\% \leq e_{\mu_{\texttt{Diversity}}} \leq 14.33\%$. On the other hand, CLEAM dramatically reduces the error for all classifier architectures. Specifically, when compared against the Baseline, the average error rate is reduced from more than 8.23% (with the Baseline) to less than 1.24%, in both StyleGAN2 and StyleSwin. The interval estimate presents similar results, where in most cases $\rho_{\texttt{CLEAM}}$ is able to bound the GT value of p^*, see Supp. Tab. 9 for the results.

5.2 RE-EVALUATING BIAS MITIGATION USING CLEAM

Importance-weighting (Choi et al., 2020) is a simple and very effective method for bias mitigation in generative models. However, its performance on fairness improvement is measured by the Baseline which could be erroneous. In this section, we re-evaluate the performance of importance-weighting with CLEAM, which has demonstrated to provide more accurate estimate in the previous section.

Following Choi et al. (2020), we utilize the original source code to train two BIGGANs (Brock et al., 2019) on the CelebA dataset(Liu et al., 2015): for the first GAN, without applying any bias mitigation (Unweighted), while the second we apply importance re-weighting (Weighted). We do this for the originally proposed attribute `Gender`, and extend the experiment to `BlackHair`. For fair comparison, we follow Choi et al. (2020) and similarly use a ResNet-18 with reasonably high average accuracy of 88% and 97% for attribute `BlackHair` and `Gender`. See Supp. D for more details on training. Our results in Tab. 2 show that the baseline measures a $\mu_{\texttt{Base}}(p_0^*)$ of 0.727 and 0.680 for Unweighted and Weighted, with `Gender` attribute (similar to reported results in Choi et al. (2020)). Note that $\mu_{\texttt{Base}}(p_0^*) = 0.5$ is considered as a fair generator. Meanwhile, CLEAM's results

Table 3: Comparing the ***point estimates*** of Baseline Choi et al. (2020), Diversity Keswani & Celis (2021) and proposed CLEAM measurement frameworks in estimating different p^* of a pseudo-generator, based on the CelebA dataset. The $\hat{p}$ is computed with a ResNet-18 classifier and the mean error is reported using Eqn. 1. We repeat this on both `Gender` and `BlackHair` attributes.

GT	**Baseline**Choi et al. (2020)		**Diversity**Keswani & Celis (2021)		**CLEAM (Ours)**	
	$\mu_{\text{Base}}(\hat{p}_0)$	$e_\mu(p_0^*)(\downarrow)$	$\mu_{\text{Div}}(\hat{p}_0)$	$e_\mu(p_0^*)(\downarrow)$	$\mu_{\text{CLEAM}}(\hat{p}_0)$	$e_\mu(p_0^*)(\downarrow)$
$\boldsymbol{\alpha}$=[0.976,0.979], `Gender`						
p_0^*=0.9	0.880	2.22%	0.950	5.55%	0.899	**0.11%**
p_0^*=0.8	0.783	2.10%	0.785	1.88%	0.798	**0.25%**
p_0^*=0.7	0.691	1.29%	0.709	1.29%	0.701	**0.14%**
p_0^*=0.6	0.592	1.33%	0.591	1.50%	0.597	**0.50%**
p_0^*=0.5	0.501	**0.20%**	0.481	3.80%	0.502	0.40%
	Avg Error	1.43%	Avg Error	2.80%	Avg Error	**0.27%**
$\boldsymbol{\alpha}$=[0.881,0.887], `BlackHair`						
p_0^*=0.9	0.803	10.77%	0.803	10.77%	0.899	**0.11%**
p_0^*=0.8	0.723	9.63%	0.699	12.63%	0.796	**0.50%**
p_0^*=0.7	0.654	6.57%	0.661	5.57%	0.705	**0.71%**
p_0^*=0.6	0.575	4.17%	0.609	1.50%	0.602	**0.33%**
p_0^*=0.5	0.500	**0.0%**	0.521	4.20%	0.504	0.8%
	Avg Error	6.23%	Avg Error	6.93%	Avg Error	**0.49%**

show that $\mu_{\text{CLEAM}}(p_0^*) > \mu_{\text{Base}}(p_0^*)$, implying that previous work could have underestimated the bias of the GANs, which could lead to an erroneous evaluation of a bias mitigation technique, or even comparing between different bias mitigation techniques. See Supp. 11 for IE results.

5.3 CLEAM FOR MEASURING THE VARYING DEGREES OF BIAS

In previous experiments, we show the performance of different methods in measuring the fairness of generators and evaluating the bias mitigation techniques. Another interesting analysis would be to see how these methods act in different degrees of bias, *i.e.* different values of p^*. A challenge of this analysis is that we cannot control the training dynamics of the GANs to obtain an exact value of p^*. Therefore, we introduce a new setup and instead, make use of a *pseudo-generator*.

In this setup, we utilize the CelebA dataset (Liu et al., 2015) to construct different modified datasets that follow different values of p^* *w.r.t.* the sensitive attribute. Then, the pseudo-generator works by random sampling from these modified datasets. Note that the samples in the modified dataset are unseen to the classifier. For example, for `BlackHair` attribute, when $p^* = \{0.9, 0.1\}$, the modified dataset contains 4880 `BlackHair` and 542 `Non-BlackHair` samples. For our experiment, we use different GT values, $p^* = \{p_0^*, p_1^*\}$, where $p_0^* \in \{0.9, 0.8, 0.7, 0.6, 0.5\}$, and $p_1^* = 1 - p_0^*$. Then, to calculate each value of $\hat{p}$ for a particular GT value of p^*, a batch of n samples is randomly drawn from the corresponding dataset and fed into the $C_{\mathbf{u}}$ for classification. We utilize the same ResNet-18 classifiers as per Sec. 5.2, to evaluate our pseudo-generator. Results in Tab.3 for p_0^* demonstrate that CLEAM is effective for different degrees of bias, reducing the average error of the Baseline from 1.43% to 0.27% and 6.23% to 0.49% for `Gender` and `BlackHair` attributes respectively. Note how measurement error in Baseline and Diversity increases by increasing the bias in the data. See Supp. C.3 and C.4 for analysis with different attributes and classifiers.

6 CONCLUSION

In this work, first, we show that existing fairness measurement framework suffers from considerable measurement errors. To reveal this problem, as generated samples are typically unlabeled, we create two new datasets by manually labeling the sensitive attributes of $\sim$9K generated images each, from two SOTA GANs. We discover that this problem arises from ignoring classification inaccuracy. Thus, to mitigate this problem, we propose CLEAM, a more accurate fairness measurement method that considers classification inaccuracies using a statistical model for classifier output. The proposed CLEAM consistently achieves improvement in fairness measurement over extensive experiments, including real generators and controlled setups. Related work, details on the new datasets, hyperparameters, additional analysis with different sensitive attributes and classifiers, and anonymous links to the code and dataset for both pseudo and SOTA generator experiments are in the supplementary.

REFERENCES

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large Scale GAN Training for High Fidelity Natural Image Synthesis. *arXiv:1809.11096 [cs, stat]*, February 2019.

Elisa Celis, Vijay Keswani, Damian Straszak, Amit Deshpande, Tarun Kathuria, and Nisheeth Vishnoi. Fair and Diverse DPP-Based Data Summarization. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 716–725. PMLR, July 2018.

Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair Clustering Through Fairlets. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Kristy Choi, Aditya Grover, Trisha Singh, Rui Shu, and Stefano Ermon. Fair Generative Modeling via Weak Supervision. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 1887–1898. PMLR, November 2020.

J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, June 2009. doi: 10.1109/CVPR.2009.5206848.

Michael Feldman, Sorelle Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. *arXiv:1412.3756 [cs, stat]*, July 2015.

Eric Frankel and Edward Vendrow. Fair Generation Through Prior Modification. *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*, June 2020.

Charles J Geyer. Stat 5101 Notes: Brand Name Distributions. *University of Minnesota*, Stat 5101: 25, January 2010.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. *arXiv:1706.04599 [cs]*, August 2017.

Moritz Hardt, Eric Price, and Nathan Srebro. Equality of Opportunity in Supervised Learning. *arXiv:1610.02413 [cs]*, October 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, June 2016.

Ahmed Imtiaz Humayun, Randall Balestriero, and Richard Baraniuk. MaGNET: Uniform Sampling from Deep Generative Network Manifolds Without Retraining. In *International Conference on Learning Representations*, January 2022.

Ben Hutchinson and Margaret Mitchell. 50 Years of Test (Un)fairness: Lessons for Machine Learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, pp. 49–58, New York, NY, USA, January 2019. Association for Computing Machinery. ISBN 978-1-4503-6125-5. doi: 10.1145/3287560.3287600.

Harsh Jaykumar Jalan, Gautam Maurya, Canute Corda, Sunny Dsouza, and Dakshata Panchal. Suspect Face Generation. In *2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA)*, pp. 73–78, April 2020. doi: 10.1109/CSCITA47329.2020.9137812.

Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, June 2019.

Harry Kesten and Norman Morse. A property of the multinomial distribution. *The Annals of Mathematical Statistics*, 30(1):120–127, 1959.

Vijay Keswani and L. Elisa Celis. Auditing for Diversity Using Representative Examples. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 860–870, Virtual Event Singapore, August 2021. ACM. ISBN 978-1-4503-8332-5. doi: 10.1145/3447548.3467433.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *ICLR 2015*, January 2017.

Agostina J. Larrazabal, Nicolás Nieto, Victoria Peterson, Diego H. Milone, and Enzo Ferrante. Gender imbalance in medical imaging datasets produces biased classifiers for computer-aided diagnosis. *Proceedings of the National Academy of Sciences*, 117(23):12592–12594, June 2020. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1919012117.

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. *Proceedings of International Conference on Computer Vision (ICCV)*, September 2015.

Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. PULSE: Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2437–2445, June 2020.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Otto Nyberg and Arto Klami. Reliably calibrated isotonic regression. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 578–589. Springer, 2021.

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, 2017.

John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

C Radhakrishna Rao. Maximum likelihood estimation for the multinomial distribution. *Sankhyā: The Indian Journal of Statistics (1933-1960)*, 18(1/2):139–148, 1957.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, Salt Lake City, UT, June 2018. IEEE. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00474.

Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. September 2014.

Shuhan Tan, Yujun Shen, and Bolei Zhou. Improving the Fairness of Deep Generative Models without Retraining. *arXiv:2012.04842 [cs]*, December 2020.

Christopher T. H. Teo and Ngai-Man Cheung. Measuring Fairness in Generative Models. *38th International Conference on Machine Learning (ICML) Workshop*, July 2021.

Kiran K Thekumparampil, Ashish Khetan, Zinan Lin, and Sewoong Oh. Robustness of conditional GANs to noisy labels. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Bowen Zhang, Shuyang Gu, Bo Zhang, Jianmin Bao, Dong Chen, Fang Wen, Yong Wang, and Baining Guo. Styleswin: Transformer-based gan for high-resolution image generation, 2021.

Please find the respective code and Dataset in the following anonymous Google-drive.

- Code: `https://drive.google.com/drive/folders/1g0ChBXQla0wfn5MAGc2EfJq5Ek8gOCqH?usp=sharing`.
- Dataset: `https://drive.google.com/drive/folders/1ENslNLyK6EEG2qj5YLZ3Qu3rFijJWEqB?usp=sharing`

A Details on Modelling

A.1 Details of Theoretical Modelling

As discussed, we model our CLEAM framework, in terms of a multinomial distribution, $N_{\mathbf{C}} \sim Multinormial(n, \mathbf{p})$.

$$\mathbf{C} = \begin{bmatrix} C_{0|0} \\ C_{1|0} \\ C_{1|1} \\ C_{0|1} \end{bmatrix} \mathbf{p} = \begin{bmatrix} p_0^* \alpha_0 \\ p_0^* \alpha_0' \\ p_1^* \alpha_1 \\ p_1^* \alpha_1' \end{bmatrix}$$

Assumptions:

1. **Classifiers are reasonably accurate.** We state that, given the advancement in classifiers architecture, and the assumption that the classifier is trained with proper training procedures, it is a reasonable assumption that it achieves reasonable accuracy and hence, $\alpha_0 \neq 0$ and $\alpha_1 \neq 0$. Similarly, we assume that it is highly unlikely to have a perfect classifier and as such $\alpha_0' \neq 0$ and $\alpha_1' \neq 0$.
2. **Generators are not completely biased.** Given that a generator is trained on a reliable dataset with the availability of all classes of a given sensitive attribute, coupled with the advancement in generator's architecture, it is a fair assumption that the generator would learn some representation of each class in the sensitive attribute and not be completely biased, as such $p_0^* \neq 0$ and $p_1^* \neq 0$.

Given that $p_i^*, \alpha_i, \alpha_i' \neq 0, \ \forall i \in \{0, 1\}$, then $0 < \mathbf{p} < 1$ and is not near the boundaries of the parameter space. Hence, we can approximate the multinormial distribution as a Gaussian, $N_{\mathbf{C}} \sim \mathcal{N}(n\mathbf{p}, n\mathbf{M})$, where

$$\mathbf{M} = \begin{bmatrix} p_0^* \alpha_0 & 0 & 0 & 0 \\ 0 & p_0^* \alpha_0' & 0 & 0 \\ 0 & 0 & p_1^* \alpha_1 & 0 \\ 0 & 0 & 0 & p_1^* \alpha_1' \end{bmatrix} - \begin{bmatrix} (p_0^* \alpha_0)^2 & (p_0^*)^2 \alpha_0 \alpha_0' & p_0^* p_1^* \alpha_0 \alpha_1 & p_0^* p_1^* \alpha_0 \alpha_1' \\ (p_0^*)^2 \alpha_0 \alpha_0' & (p_0^* \alpha_0')^2 & p_0^* p_1^* \alpha_0' \alpha_1 & p_0^* p_1^* \alpha_0' \alpha_1' \\ p_0^* p_1^* \alpha_0 \alpha_1 & p_0^* p_1^* \alpha_0' \alpha_1 & (p_1^* \alpha_1)^2 & (p_1^*)^2 \alpha_1 \alpha_1' \\ p_0^* p_1^* \alpha_0 \alpha_1' & p_0^* p_1^* \alpha_0' \alpha_1' & (p_1^*)^2 \alpha_1 \alpha_1' & (p_1^* \alpha_1')^2 \end{bmatrix}$$

With that, we solve for the marginal distribution of N_{C_0} and N_{C_1}

$$N_{C_0} = N_{C_{0|0}} + N_{C_{0|1}} \sim \mathcal{N}(\tilde{\mu}_{C_0}, \tilde{\sigma}_{C_0}^2)$$
$$\tilde{\mu}_{C_0} = n(p_0^* \alpha_0 + p_1^* \alpha_1')$$
$$\tilde{\sigma}_{C_0}^2 = n(p_0^* \alpha_0 - (p_0^* \alpha_0)^2) + n(p_1^* \alpha_1' - (p_1^* \alpha_1')^2) + 2np_0^* p_1^* \alpha_0 \alpha_1'$$

$$N_{C_1} = N_{C_{1|0}} + N_{C_{1|1}} \sim \mathcal{N}(\tilde{\mu}_{C_1}, \tilde{\sigma}_{C_1}^2)$$
$$\tilde{\mu}_{C_1} = n(p_0^* \alpha_0' + p_1^* \alpha_1)$$
$$\tilde{\sigma}_{C_1}^2 = n(p_0^* \alpha_0' - (p_0^* \alpha_0')^2) + n(p_1^* \alpha_1 - (p_1^* \alpha_1)^2) + 2np_0^* p_1^* \alpha_0' \alpha_1$$

We then normalise N_{C_i}, to get $\hat{p}_i = \frac{1}{n} N_{C_i}, \ i \in \{0, 1\}$

$$\hat{p}_0 \sim \mathcal{N}(\tilde{\mu}_{\hat{p}_0}, \tilde{\sigma}_{\hat{p}_0}^2)$$
$$\tilde{\mu}_{\hat{p}_0} = p_0^* \alpha_0 + p_1^* \alpha_1'$$
$$\tilde{\sigma}_{\hat{p}_0}^2 = \frac{1}{n}(p_0^* \alpha_0 - (p_0^* \alpha_0)^2) + \frac{1}{n}(p_1^* \alpha_1' - (p_1^* \alpha_1')^2) + \frac{2}{n} p_0^* p_1^* \alpha_0 \alpha_1'$$

$$\hat{p}_1 \sim \mathcal{N}(\mu_{\hat{p}_1}, \sigma^2_{\hat{p}_1})$$
$$\tilde{\mu}_{\hat{p}_1} = (p^*_0\alpha'_0 + p^*_1\alpha_1)$$
$$\tilde{\sigma}^2_{\hat{p}_1} = \frac{1}{n}(p^*_0\alpha'_0 - (p^*_0\alpha'_0)^2) + \frac{1}{n}(p^*_1\alpha_1 - (p^*_1\alpha_1)^2) + \frac{2}{n}p^*_0p^*_1\alpha'_0\alpha_1$$

A.2 Additional Details on CLEAM Algorithm

In the following, we show that the maximum likelihood estimate (MLE) of the population mean for a Gaussian distribution, is it's sample mean:

$$\frac{\partial}{\partial\tilde{\mu}_{\hat{p}_0}} \prod_{i=1}^{s} ln(\frac{1}{\tilde{\sigma}_{\hat{p}_0}\sqrt{2\pi}} e^{\frac{-(\hat{p}^i_0 - \tilde{\mu}_{\hat{p}_0})^2}{2\tilde{\sigma}^2_{\hat{p}_0}}}) = 0$$
$$\frac{1}{\tilde{\sigma}^2_{\hat{p}_0}} \sum_{i=0}^{s} (\hat{p}^i_0 - \tilde{\mu}_{\hat{p}_0}) = 0$$
$$\tilde{\mu}_{\hat{p}_0} = \frac{1}{s}\sum_{i}^{s} \hat{p}^i_0 = \ddot{\mu}_{\hat{p}_0}$$

This proof shows that, given a Gaussian distribution with population mean $\tilde{\mu}_{\hat{p}_0}$ and standard deviation $\tilde{\sigma}_{\hat{p}_0}$, we can first find the joint probability distribution from the product of each probabilistic outcome (we introduce the natural log as a monotonic function, for ease of calculation). Then, to find the MLE of $\tilde{\mu}_{\hat{p}_0}$, we take the partial derivative of this joint distribution *w.r.t.* $\tilde{\mu}_{\hat{p}_0}$, and solve for its the maximum value. We find this maximum value to be equal to the sample mean, $\ddot{\mu}_{\hat{p}_0}$.

From this, given that s is large, we assume that the sample mean is a good approximation for the population mean and w equate them:

$$\ddot{\mu}_{\hat{p}_0} = \tilde{\mu}_{\hat{p}_0} = p^*_0\alpha_0 + (1 - p^*_0)\alpha'_1$$

We then solve for the maximum likelihood point estimate of p^*, which we denoted with $\mu_{\text{CLEAM}}(p^*)$.

$$\mu_{\text{CLEAM}}(p^*_0) = \frac{\ddot{\mu}_{\hat{p}_0} - \alpha'_1}{\alpha_0 - \alpha'_1} = \frac{\ddot{\mu}_{\hat{p}_0} - 1 + \alpha_1}{\alpha_0 - 1 + \alpha_1}$$
$$\mu_{\text{CLEAM}}(p^*_1) = 1 - \mu_{\text{CLEAM}}(p^*_0)$$

However, we acknowledge that there exist other statistically probable solutions for p^* that could output the s $\hat{p}$ samples, other than the Maximum likelihood point estimate of p^*. We thus propose the following approximation for the 95% confidence interval of p^*. Since $\hat{p}$ distribution is assumed Gaussian, we can propose the following equation. Recall that the notations $\ddot{\mu}_{\hat{p}_0}$ and $\ddot{\sigma}_{\hat{p}_0}$ are the sample mean and standard deviation respectively.

$$Pr(-z_{\frac{\delta}{2}} \leq \frac{\ddot{\mu}_{\hat{p}_0} - \tilde{\mu}_{\hat{p}_0}}{\frac{\ddot{\sigma}_{\hat{p}_0}}{\sqrt{s}}} \leq \mathbf{z}_{\frac{\delta}{2}}) = 1 - \delta$$

$$, where \quad \ddot{\mu}_{\hat{p}_0} = \frac{1}{s}\sum_{i}^{s} \hat{p}^i_0 \quad and \quad \ddot{\sigma}_{\hat{p}_0} = \sqrt{\frac{\sum_{i=1}^{s}(\hat{p}^i_0 - \ddot{\mu}_{\hat{p}_0})}{s}}$$

Solving for $\tilde{\mu}_{\hat{p}}$, we get

$$Pr(-z_{\frac{\delta}{2}} \leq \frac{\ddot{\mu}_{\hat{p}_0} - \tilde{\mu}_{\hat{p}_0}}{\frac{\ddot{\sigma}_{\hat{p}_0}}{\sqrt{s}}} \leq \mathbf{z}_{\frac{\delta}{2}}) = 1 - \delta$$
$$Pr(\ddot{\mu}_{\hat{p}_0} + z_{\frac{\delta}{2}}(\frac{\ddot{\sigma}_{\hat{p}_0}}{\sqrt{s}}) \geq \tilde{\mu}_{\hat{p}_0} \geq \ddot{\mu}_{\hat{p}_0} - z_{\frac{\alpha}{2}}\frac{\ddot{\sigma}_{\hat{p}_0}}{\sqrt{s}}) = 1 - \delta$$

Then, given that $\tilde{\mu}_{\hat{p}} = p^*_0\alpha_0 + p^*_1\alpha'_1 = p^*_0(\alpha_0 - \alpha'_1) + \alpha'_1$ we formulate the following:

$$Pr(\frac{\ddot{\mu}_{\hat{p}_0} + z_{\frac{\delta}{2}}(\frac{\ddot{\sigma}_{\hat{p}_0}}{\sqrt{s}}) - \alpha'_1}{\alpha_0 - \alpha'_1} \geq p^*_0 \geq \frac{\ddot{\mu}_{\hat{p}_0} - z_{\frac{\alpha}{2}}\frac{\ddot{\sigma}_{\hat{p}_0}}{\sqrt{s}} - \alpha'_1}{\alpha_0 - \alpha'_1}) = 1 - \delta \tag{12}$$

As such when $\delta = 0.05$, we can determine that the 95% approximated confidence interval of p^*_0 is :

$$\rho_{\text{CLEAM}}(p^*_0) = [\mathcal{L}(p^*_0), \mathcal{U}(p^*_0)] = [\frac{\ddot{\mu}_{\hat{p}_0} - 1.96(\frac{\ddot{\sigma}_{\hat{p}_0}}{\sqrt{s}}) - \alpha'_1}{\alpha_0 - \alpha'_1} \quad , \quad \frac{\ddot{\mu}_{\hat{p}_0} + 1.96\frac{\ddot{\sigma}_{\hat{p}_0}}{\sqrt{s}} - \alpha'_1}{\alpha_0 - \alpha'_1}]$$

A.3 DETAILS ON FAIRNESS METRIC

Fairness in generative models is defined as *Equal Representation* meaning that the generator is supposed to generate an equal number of samples for each element of an attribute, e.g., an equal number of generated `Male` and `Female` samples when the sensitive attribute is `Gender`. Therefore, the expected distribution for a fair generator is usually a uniform distribution denoted by $\bar{p}$. Considering this, the fairness discrepancy (FD) metric (Choi et al., 2020) measures the L2 norm between $\bar{p}$ and the estimated class probability of the generator by the sensitive attribute classifier $C_{\mathbf{u}}$, as follows:

$$f = |\bar{p} - \mathbb{E}_{z \sim p_z(z)}[C_{\mathbf{u}}(\mathbf{G}(\mathbf{z}))]|_2 \tag{13}$$

where $C_{\mathbf{u}}(G(z))$ is the one-hot vector for the classified label of the generated sample, $G(z)$. z is sampled from a Gaussian noise distribution $p_z(z)$. Note that for a fair generator the fairness discrepancy f would be zero, which also indicates zero bias.

B VALIDATING STATISTICAL MODEL FOR CLASSIFIER OUTPUT

B.1 VALIDATION OF SAMPLE-BASED ESTIMATE VS MODEL-BASED ESTIMATE

As described in the main paper, we utilise the sample-based estimate, $\ddot{\mu}_{\hat{p}_0}$, $\ddot{\sigma}^2_{\hat{p}_0}$ as an approximate for the model-based estimate $\tilde{\mu}_{\hat{p}_0}$, $\tilde{\sigma}^2_{\hat{p}_0}$. As discussed in Sec. A.2, $\ddot{\mu}_{\hat{p}_0}$ allows us to find the maximum likelihood approximate of p^* and $\ddot{\sigma}^2_{\hat{p}_0}$ allows us to ease computation.

To validate this approximation, we use a batch-size $s = 30$ and sample size $n = 400$ in each batch to generate s different $\hat{p}$ values from the pseudo-generators (Sec. 5.3 of the main paper) , with different GT p^*. Then we calculate the sample-based estimates as given in Eqn. 7, 8 of the main paper. As the GT p^* is known, we also calculate the model-based estimates as given in Eqn. 5, 6 and compare it against the sample-based estimates.

Our results in Tab. 4 shows that both the sample and theoretical means and standard deviations are close approximate to one another. Thus, we can utilise the sample statistics as a close approximation in our proposed method, CLEAM.

Additional results for different values of batch-sizes (s) and sample-sizes (n) are tabulated in Tab. 5, 6 and 7. Notice that a reduction in s and n values contributed to increased errors between the sample-based and model-based estimates. While making s very large ($s = 200$), results in the sample based estimate almost a perfectly approximating the model based estimates.

Table 4: **Comparing sample-based estimates ($\ddot{\boldsymbol{\mu}}_{\hat{\boldsymbol{p}}_0}$, $\ddot{\boldsymbol{\sigma}}_{\hat{\boldsymbol{p}}_0}$) against model-based estimates ($\tilde{\boldsymbol{\mu}}_{\hat{\boldsymbol{p}}_0}$, $\tilde{\boldsymbol{\sigma}}_{\hat{\boldsymbol{p}}_0}$).** The results show that sample-based estimates are close to model-based estimates. Furthermore, note the discrepancy between p^*_0 and $\ddot{\mu}_{\hat{p}_0}$, and that between p^*_0 and $\tilde{\mu}_{\hat{p}_0}$, highlighting the issue of using $\hat{p}_0$ directly to estimate p^*_0 and the need to compensate for classifier error as we discussed.

GT	Sampled-based estimates		Model-based estimates	
	$\ddot{\mu}_{\hat{p}_0}$	$\sqrt{\ddot{\sigma}^2_{\hat{p}_0}}$	$\tilde{\mu}_{\hat{p}_0}$	$\sqrt{\tilde{\sigma}^2_{\hat{p}_0}}$
Gender, $\boldsymbol{\alpha}$=[0.976,0.979]				
$p^*_0 = 0.9$	0.881	0.0101	0.881	0.0106
$p^*_0 = 0.8$	0.781	0.0133	0.785	0.0135
$p^*_0 = 0.7$	0.692	0.0149	0.690	0.0152
$p^*_0 = 0.6$	0.590	0.0165	0.594	0.0162
$p^*_0 = 0.5$	0.503	0.0164	0.499	0.0164
$\boldsymbol{\alpha}$=[0.881,0.887], Black-Hair				
$p^*_0 = 0.9$	0.802	0.0130	0.804	0.0139
$p^*_0 = 0.8$	0.723	0.0151	0.727	0.0162
$p^*_0 = 0.7$	0.653	0.0169	0.650	0.0177
$p^*_0 = 0.6$	0.580	0.0180	0.574	0.0186
$p^*_0 = 0.5$	0.502	0.0180	0.497	0.0189

Table 5: We repeat the same experiment as Tab.4 with $s = 20$ and $n = 400$ samples.

GT	Sampled-based estimates		Model-based estimates	
	$\ddot{\mu}_{\hat{p}_0}$	$\sqrt{\ddot{\sigma}^2_{\hat{p}_0}}$	$\tilde{\mu}_{\hat{p}_0}$	$\sqrt{\tilde{\sigma}^2_{\hat{p}_0}}$
Gender, $\boldsymbol{\alpha}$=[0.976,0.979]				
$p^*_0 = 0.9$	0.855	0.0201	0.881	0.0106
$p^*_0 = 0.8$	0.774	0.0211	0.785	0.0135
$p^*_0 = 0.7$	0.672	0.0219	0.690	0.0152
$p^*_0 = 0.6$	0.580	0.0181	0.594	0.0162
$p^*_0 = 0.5$	0.510	0.0230	0.499	0.0164
$\boldsymbol{\alpha}$=[0.881,0.887], Black-Hair				
$p^*_0 = 0.9$	0.768	0.180	0.804	0.0139
$p^*_0 = 0.8$	0.712	0.210	0.727	0.0162
$p^*_0 = 0.7$	0.658	0.190	0.650	0.0177
$p^*_0 = 0.6$	0.554	0.230	0.574	0.0186
$p^*_0 = 0.5$	0.508	0.242	0.497	0.0189

Table 6: We repeat the same experiment as per Tab.4 with $s = 30$ and $n = 200$ samples.

GT	Sampled-based estimates		Model-based estimates	
	$\ddot{\mu}_{\hat{p}_0}$	$\sqrt{\ddot{\sigma}^2_{\hat{p}_0}}$	$\tilde{\mu}_{\hat{p}_0}$	$\sqrt{\tilde{\sigma}^2_{\hat{p}_0}}$
Gender, $\boldsymbol{\alpha}$=[0.976,0.979]				
$p^*_0 = 0.9$	0.860	0.0232	0.881	0.0149
$p^*_0 = 0.8$	0.780	0.0286	0.785	0.0191
$p^*_0 = 0.7$	0.710	0.0294	0.690	0.0215
$p^*_0 = 0.6$	0.578	0.0380	0.594	0.0228
$p^*_0 = 0.5$	0.520	0.0321	0.499	0.0233
$\boldsymbol{\alpha}$=[0.881,0.887], Black-Hair				
$p^*_0 = 0.9$	0.742	0.0312	0.804	0.0197
$p^*_0 = 0.8$	0.740	0.0332	0.727	0.0229
$p^*_0 = 0.7$	0.610	0.0291	0.650	0.0250
$p^*_0 = 0.6$	0.582	0.350	0.574	0.0262
$p^*_0 = 0.5$	0.542	0.388	0.497	0.0267

Table 7: We repeat the same experiment as per Tab.4 with $s = 200$ and $n = 400$ samples.

GT	Sampled-based estimates		Model-based estimates	
	$\ddot{\mu}_{\hat{p}_0}$	$\sqrt{\ddot{\sigma}^2_{\hat{p}_0}}$	$\tilde{\mu}_{\hat{p}_0}$	$\sqrt{\tilde{\sigma}^2_{\hat{p}_0}}$
Gender, $\boldsymbol{\alpha}$=[0.976,0.979]				
$p^*_0 = 0.9$	0.881	0.0104	0.881	0.0106
$p^*_0 = 0.8$	0.784	0.0133	0.785	0.0135
$p^*_0 = 0.7$	0.690	0.0153	0.690	0.0152
$p^*_0 = 0.6$	0.594	0.0160	0.594	0.0162
$p^*_0 = 0.5$	0.500	0.0164	0.499	0.0164
$\boldsymbol{\alpha}$=[0.881,0.887], Black-Hair				
$p^*_0 = 0.9$	0.804	0.0137	0.804	0.0139
$p^*_0 = 0.8$	0.726	0.0160	0.727	0.0162
$p^*_0 = 0.7$	0.650	0.0179	0.650	0.0177
$p^*_0 = 0.6$	0.573	0.0185	0.574	0.0186
$p^*_0 = 0.5$	0.498	0.0191	0.497	0.0189

B.2 Goodness-Of-Fit Test: $\hat{p}$ from the Real GANs with Our Theoretical Model

Table 8: **Validating theoretical model on GAN:** KS-test on $s = 30$ and $\delta = 0.05$ with $D_{crit} = 0.24$. As seen from the table, since $\eta < D_{crit}$, all of the generated samples by GANs are statistically similar to the respective Gaussian at a 95% confidence of the K-S test.

Model Type	Sensitive Attribute	η
StyleGAN2	Gender	0.1048
StyleSwin	Gender	0.1509
StyleGAN2	Blackhair	0.1065
StyleSwin	Blackhair	0.1079

In order to make sure that our proposed theoretical model in Eqn. 5 and Eqn. 6 of the main paper, is also a good representation of the $\hat{p}$ distribution when using a real GAN as generator, here we perform a goodness of fit test between proposed model for distribution of $\hat{p}$, and sample data generated by a GAN. To do this, we first obtain $s = 30$ values of $\hat{p}$ from framework shown in Fig. 1 of the main paper, and use StyleGANv2 and StyleSwin as the generative model. Then using the known classifier's $\boldsymbol{\alpha}$ and GAN's GT p^*, as discussed in Sec. 3.1 of the main paper, we form the theoretical models Gaussian distribution, $\mathcal{N}(\tilde{\mu}_{\hat{p}_0}, \tilde{\sigma}^2_{\hat{p}_0})$.

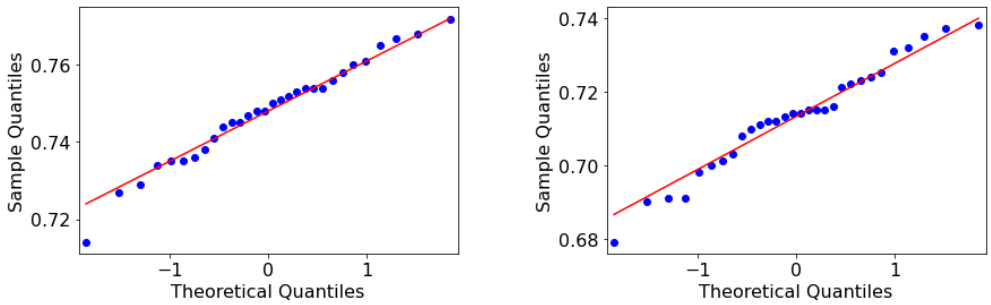
Now with both our model distribution and the GAN samples, we utilise the Kolmogorov-Smirnov goodness of fit test (K-S test) to determine if the samples distribution is statistically similar to the proposed Gaussian model. We thus propose the following hypothesis test for the samples $\hat{p}^i_j, i \in \{1, \cdots, s\}$:

$\mathbf{H_0}$: *the samples $\hat{p}_j^i$ belong to the modelled distribution.*

$\mathbf{H_1}$: *at least one of the samples $\hat{p}_j^i$ does not match the modelled distribution.*

The K-S test then measures a D-statistic (η) and compares it against a D_{crit} for a given s. As we use $s = 30$, and a significance level $\delta = 0.05$ in our setup, we have $D_{crit} = 0.24$. As seen from Tab. 8, all of the measured η values are below D_{crit}, thus we cannot reject the null hypothesis at a 95% confidence with the K-S test. Therefore, we conclude that the distribution of the obtained samples from the framework (by GANs as generator) are statistically similar to the proposed Gaussian distribution. As a result, we can utilise CLEAM to approximate the p^* range in the presence of a real GAN as the generator.

We further perform a Quantile-Quantile(QQ) analysis to provide a more visual representation. In particular, we plot the Quantile-Quantile(QQ) plot between the $\hat{p}$ samples (produced for the data generated by the GAN) and proposed model. As seen in Fig. 3, the $\hat{p}$ samples from GAN correlate tightly with the standardised line (in red), a line indicating a perfect correlation between theoretical and sample quantiles. This analysis supports our claim that the $\hat{p}$ samples from a real generator (GAN) follow the distribution estimated by the proposed model.



(a) StyleGAN2, Gender.

(b) StyleSwin2, Gender

(c) StyleGAN2, Blackhair

(d) StyleSwin, BlackHair

Figure 3: Quartile-Quartile(QQ) plot between $s = 30$ $\hat{p}$ samples calculated for StyleGAN2 (Karras et al., 2019) and StyleSwin (Zhang et al., 2021) generators and proposed theoretical model for $\hat{p}$

C Additional Experimental Results

C.1 Fairness Measurement Results with Interval Estimate of CLEAM

This section contains the interval estimates (IE) results for the experimental in Sec. 5. As discussed in the main manuscript, the point estimate can still contain some approximation error, as a result we suggest that in addition to the PE, the IE should be reported. Our results show that in our experiments where the GT p^* is known CLEAM's IE, in most cases, demonstrates being able to to bound the p^*. Additionally, when analyzing bias mitigation techniques (as per Tab. 11), since the IE of unweighted and weighted GANs do not overlap, we are provided some statistical grantees that the bias mitigation techniques is indeed effective.

Table 9: Comparing **interval values** of Baseline (Choi et al., 2020), Diversity (Keswani & Celis, 2021) and our proposed CLEAM measurement framework in evaluating the p^* of datasets sampled from (A) StyleGAN2 (Karras et al., 2019) and (B) StyleSwim (Zhang et al., 2021). We utilize four different classifier Resnet-18/34 (He et al., 2016), MobileNetv2 (Sandler et al., 2018) and VGG-16 (Simonyan & Zisserman, 2014) to classify the SA, `Gender` and `Blackhair`, and measure their distribution. We then compared the measured values against the GT p^* of the datasets and reported the mean errors with Eqn. 2.

(A) StyleGAN2							
Classifier	**GT**	**Baseline**(Choi et al., 2020)		**Diversity**(Keswani & Celis, 2021)		**(Ours) CLEAM**	
		$\rho_{\text{Base}}(p_0^*)$	$e_\rho(p_0^*)(\downarrow)$	$\rho_{\text{Div}}(p_0^*)$	$e_\rho(p_0^*)(\downarrow)$	$\rho_{\text{CLEAM}}(p_0^*)$	$e_\rho(p_0^*)(\downarrow)$
`Gender`							
ResNet-18	$p_0^* = 0.642$	[0.602, 0.618]	6.1%	−.−	−. − %	[0.629, 0.646]	**1.9**%
ResNet-34		[0.589, 0.599]	8.1%	−.−	−. − %	[0.628, 0.638]	**2.2**%
MobileNetv2		[0.602, 0.612]	6.2%	−.−	−. − %	[0.632, 0.643]	**1.6**%
VGG-16		[0.526, 0.538]	18.0%	[0.536 , 0.564]	16.5%	[0.628, 0.644]	**2.2**%
		Avg Error	9.6%	Avg Error	16.5%	Avg Error	**2.0**%
`Blackhair`							
ResNet-18	$p_0^* = 0.643$	[0.591, 0.607]	8.0%	−.−	−. − %	[0.631, 0.652]	**2.1**%
ResNet-34		[0.561, 0.572]	12.7%	−.−	−. − %	[0.637, 0.651]	**1.4**%
MobileNetv2		[0.574, 0.584]	10.7%	−.−	−. − %	[0.632, 0.647]	**1.7**%
VGG-16		[0.597, 0.608]	5.9%	[0.568, 0.596]	11.7%	[0.632, 0.648]	**1.9**%
		Avg Error	9.3%	Avg Error	11.7%	Avg Error	**1.8**%
(B) StyleSwin							
`Gender`							
ResNet-18	$p_0^* = 0.656$	[0.612, 0.629]	6.9%	−.−	−. − %	[0.639, 0.658]	**2.7**%
ResNet-34		[0.605, 0.615]	8.0%	−.−	−. − %	[0.643, 0.654]	**2.0**%
MobileNetv2		[0.618, 0.629]	5.9%	−.−	−. − %	[0.649, 0.661]	**1.3**%
VGG-16		[0.549, 0.560]	16.3%	[0.548, 0.576]	16.4%	[0.660, 0.675]	**2.7**%
		Avg Error	9.3%	Avg Error	16.4%	Avg Error	**2.2**%
`Blackhair`							
ResNet-18	$p_0^* = 0.668$	[0.605, 0.620]	9.5%	−.−	−. − %	[0.649, 0.670]	**2.9**%
ResNet-34		[0.576, 0.586]	11.8%	−.−	−. − %	[0.656, 0.669]	**1.9**%
MobileNetv2		[0.591, 0.600]	12.7%	−.−	−. − %	[0.652, 0.666]	**2.4**%
VGG-16		[0.620, 630]	7.7%	[0.590, 0.626]	11.7%	[0.670, 0.684]	**2.4**%
		Avg Error	10.4%	Avg Error	11.7%	Avg Error	**2.4**%

Table 10: Comparing the ***interval estimates*** of Baseline Choi et al. (2020), Diversity Keswani & Celis (2021) and proposed CLEAM measurement frameworks in measuring different p^* of a pseudo-generator, based on the CelebA dataset. The $\hat{p}$ is assess with a ResNet-18 classifier and the mean error is reported using Eqn. 1. We repeat this on both `Gender` and `BlackHair` attributes.

(A) Pseudo-Generator						
GT	**Baseline**(Choi et al., 2020)		**Diversity**(Keswani & Celis, 2021)		**(Ours) CLEAM**	
	$\rho_{\text{Base}}(p_0^*)$	$e_\rho(p_0^*)(\downarrow)$	$\rho_{\text{Div}}(p_0^*)$	$e_\rho(p_0^*)(\downarrow)$	$\rho_{\texttt{CLEAM}}(p_0^*)$	$e_\rho(p_0^*)(\downarrow)$
$\boldsymbol{\alpha}$=[0.976,0.979], `Gender`						
$p_0^* = 0.9$	$[0.876, 0.884]$	2.7%	$[0.913, 0.986]$	9.6%	$[0.895, 0.904]$	$\mathbf{0.5}\%$
$p_0^* = 0.8$	$[0.778, 0.788]$	2.8%	$[0.762, 0.809]$	4.8%	$[0.794, 0.803]$	$\mathbf{0.8}\%$
$p_0^* = 0.7$	$[0.687, 0.695]$	1.9%	$[0.696, 0.722]$	3.1%	$[0.697, 0.707]$	$\mathbf{0.1}\%$
$p_0^* = 0.6$	$[0.586, 0.598]$	2.3%	$[0.581, 0.612]$	3.3%	$[0.591,0.603]$	$\mathbf{1.5}\%$
$p_0^* = 0.5$	$[0.495, 0.507]$	$\mathbf{1.0}\%$	$[0.473, 0.490]$	5.5%	$[0.497, 0.508]$	1.6%
	Avg Error	2.1%	Avg Error	5.26%	Avg Error	$\mathbf{0.90}\%$
$\boldsymbol{\alpha}$=[0.881,0.887], `BlackHair`						
$p_0^* = 0.9$	$[0.800, 0.806]$	11.0%	$[0.791, 0.815]$	12.1%	$[0.893, 0.905]$	$\mathbf{0.8}\%$
$p_0^* = 0.8$	$[0.719, 0.727]$	10.1%	$[0.686, 0.713]$	14.2%	$[0.790, 0.803]$	$\mathbf{1.3}\%$
$p_0^* = 0.7$	$[0.648, 0.660]$	7.4%	$[0.643, 0.68]$	8.2%	$[0.698, 0.712]$	$\mathbf{1.6}\%$
$p_0^* = 0.6$	$[0.564, 0.586]$	5.8%	$[0.604, 0.614]$	2.3%	$[0.599, 0.606]$	$\mathbf{1.0}\%$
$p_0^* = 0.5$	$[0.495, 0.505]$	$\mathbf{1.0}\%$	$[0.506, 0.536]$	7.2%	$[0.497, 0.511]$	1.9%
	Avg Error	7.06%	Avg Error	8.8%	Avg Error	$\mathbf{1.32}\%$

Table 11: Re-evaluating the **interval estimates** of previously proposed bias mitigation method, importance-weighting (imp-weighting) (Choi et al., 2020) with CLEAM. To do this, we first evaluate the bias of a BIGGAN (Brock et al., 2019) with and without implementing imp-weighting *i.e.* unweighted and weighted, with the Baseline. Then, we apply CLEAM to obtain a more accurate measurements, which we use to compare against the Baseline. We do this for both `Gender` and `BlackHair` attributes.

(A) Pseudo-Generator						
GT	**Baseline**(Choi et al., 2020)		**Diversity**(Keswani & Celis, 2021)		**(Ours) CLEAM**	
	$\rho_{\text{Base}}(p_0^*)$	$e_\rho(p_0^*)(\downarrow)$	$\rho_{\text{Div}}(p_0^*)$	$e_\rho(p_0^*)(\downarrow)$	$\rho_{\texttt{CLEAM}}(p_0^*)$	$e_\rho(p_0^*)(\downarrow)$
(B) Importance Re-weighting (Choi et al., 2020) GAN						
$\boldsymbol{\alpha}$=[0.976,0.979], `Gender`						
Unweighted	$[0.721, 0.732]$	-	$[0.697, 0.722]$	-	$[0.733, 0.744]$	-
Weighted	$[0.674, 0.685]$	-	$[0.658, 0.684]$	-	$[0.686, 0.693]$	-
$\boldsymbol{\alpha}$=[0.881,0.887], `BlackHair`						
Unweighted	$[0.725, 0.733]$	-	$[0.704, 0.729]$	-	$[0.798, 0.809]$	-
Weighted	$[0.710, 0.722]$	-	$[0.696, 0.716]$	-	$[0.778, 0.792]$	-

C.2 EXPERIMENTAL SETUP FOR DIVERSITY(KESWANI & CELIS, 2021)

In this section, we describe our experimental setup for Diversity (Keswani & Celis, 2021), as utilised in the main paper. Recall that in Keswani & Celis (2021) a VGG-16 (Simonyan & Zisserman, 2014) model pre-trained on ImageNet (Deng et al., 2009) is utilised as a feature extractor. Then, this feature extractor is applied on both the unknown (generator's data) and the controlled dataset. Finally, the unknown sample's features are compared against the controlled one's via a similarity algorithm to compute diversity, δ.

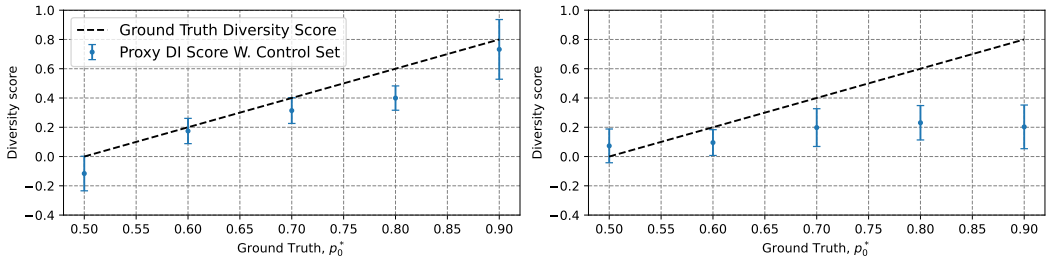
From our initial results, on the pseudo-generator's setup in Fig. 4a , we recognise that the original implementation with VGG-16 trained on ImageNet works well on the `Gender` attribute. To demonstrate this, we compare our measured proxy diversity score against the GT diversity score shown in Eqn. 14, as per Keswani & Celis (2021).

$$GT\ Diversity = p_0^* - p_1^* \tag{14}$$

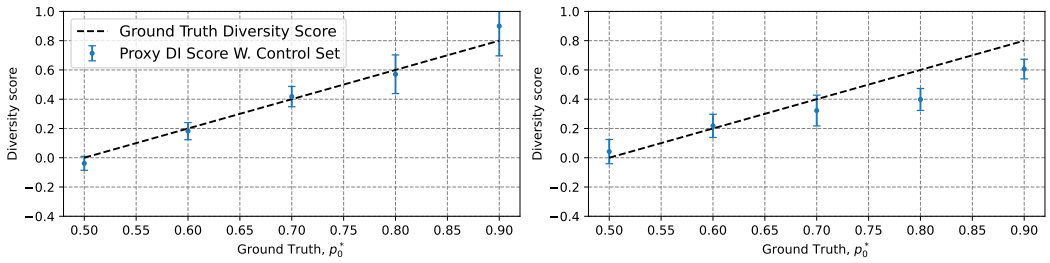
However, when evaluating the harder `BlackHair` attribute, we observed significant error between the GT Diversity scores and the proxy Diversity scores. This error was especially prevalent in the larger biases *e.g.* $p_0^* = 0.9$. We theorised that, this was due to the differences between the domains of the feature extractor and the generated/controlled images *i.e.* ImageNet versus CelebA/CelebA-HQ.

To verify this, we retrained the VGG-16 model on the CelebA dataset with the respective sensitive attribute. Then we removed the last fully connected layer of the classifier model, and utilise the 4096 feature vector for the diversity measurement, as per Keswani & Celis (2021). From our results in Fig. 4b, we see significant improvement in our implementation on both `Gender` and `BlackHair`.

However, we recognise certain limitations still exist in the Diversity measure when used on more ambiguous and harder attributes *e.g.* `Young` and `Attractive`, even though the re-trained classifier measured an accuracy of 78.44% and 84.41% for `Young` and `Attractive`, respectively. Regardless, given the improvement seen on the `BlackHair` attribute, we utilized our improved VGG-16 feature extractor in the main paper, in place of the erroneous pre-trained VGG-16 (ImageNet) .

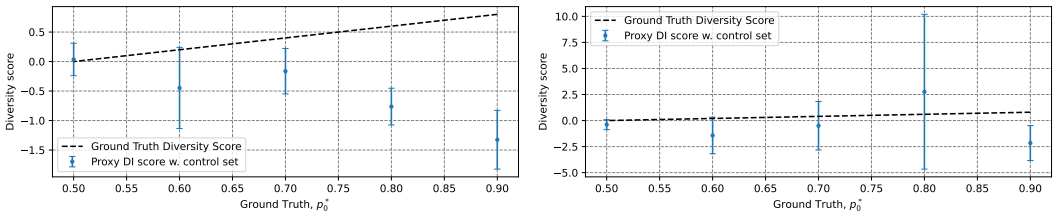


(a) VGG-16 trained on ImageNet



(b) VGG-16 trained on CelebA

Figure 4: **Improvement in Diversity by fine-tuning the VGG-16, as a feature extractor**: **(a)** Diversity implementation by Keswani & Celis (2021) with VGG-16 pre-trained on ImageNet as the feature extractor testing on the pseudo-generator's with $p^* = \{0.9, 0.8, 0.7, 0.6, 0.5\}$ for sensitive attribute `Gender`(Left) and `BlackHair`(Right). **(b)** We re-implemented VGG-16 and retrained with CelebA as the feature extractor. We observed significant improvement in predicting the GT p^*



(a) VGG-16(CelebA) on Attractive

(b) VGG-16(CelebA) on Young

Figure 5: **Limitations Of Diversity algorithm.** Our implementation of VGG-16 trained on CelebA on sensitive attribute Attractive and Young. VGG-16 Classifier achieved an accuracy of 78.44% and 84.1% for sensitive attribute Attractive and Young. However, the same VGG-16 performs poorly on the diversity metric, demonstrating the limitations of the diversity framework.

C.3 MEASURING VARYING DEGREES OF BIAS WITH ADDITIONAL SENSITIVE ATTRIBUTES

In Sec. 5.3 of the main paper, we demonstrate CLEAM's ability to improve accuracy in approximating p^* for the sensitive attribute, Gender and BlackHair. In this section, we extend the experiment on harder (lower $\boldsymbol{\alpha}$) sensitive attributes *i.e.* Young, and Attractive. We further demonstrate that the Baseline is also sensitive to $skew_{\boldsymbol{\alpha}} = (\alpha_1 - \alpha_0)$.

In this setup, we repeat the same experiment with the pseudo-generator, but instead on the Young and Attractive attributes from the CelebA dataset. Given that both sensitive attribute classifiers trained on either Young or Attractive have similar average accuracy, $\alpha_{Avg} = \frac{\alpha_0+\alpha_1}{2}$ of 0.801 and 0.794 but different $skew_{\boldsymbol{\alpha}}$ of 0.103 and 0.027, we are able to investigate the effects of $skew_{\boldsymbol{\alpha}}$ on both CLEAM and Baseline. We did not include Diversity in this study, due to its poor performance on harder sensitive attribute, as discussed in C.2.

From our results in Tab.5, we observe that as the $skew_{\boldsymbol{\alpha}}$ increases from sensitive attribute Attractive to Young, the error becomes much more significant in the baseline method. The average Baseline e_μ increases from 12.3% to 17.6%. On the other hand, CLEAM's e_μ remains below 1%, further emphasising CLEAM's effectiveness. Similar observation can be made for the interval estimates in Tab.13.

Furthermore, we observe that $skew_{\boldsymbol{\alpha}}$ has influence on the errors observed in the lower biases *e.g.* $p^* = [0.5, 0.5]$. Unlike Gender and Blackhair, who have relatively negligible skew, Young and Attractive observes a significantly larger error on $p^* = [0.5, 0.5]$. We attribute this difference in performance at $p_0^* = 0.5$ due to Gender and Blackhair setups having a specific combination of (i) Generator producing almost perfectly unbias data with $p^* = [0.5, 0.5]$ (ii) sensitive attribute classifier with almost perfectly uniform inaccuracies $\boldsymbol{\alpha}' = 1 - \boldsymbol{\alpha}$ *i.e.* $skew_{\boldsymbol{\alpha}} \approx 0$, thereby leading to uniform misclassification and giving rise to the false impression of better accuracy by the baseline method, at $p^* = [0.5, 0.5]$.

Table 12: Comparing ***point estimate values*** of Baseline (Choi et al., 2020), and proposed CLEAM measurement framework on (A) pseudo-generator with sensitive attribute {Young, Attractive}.

Pseudo-Generator				
Test	**Baseline**(Choi et al., 2020)		**(Ours) CLEAM**	
	$\mu_{\text{Base}}(p_0^*)$	$e_\mu(p_0^*)$	$\mu_{\text{CLEAM}}(p_0^*)$	$e_\mu(p_0^*)$
$\boldsymbol{\alpha}$=[0.749,0.852], Young				
$p_0^* = 0.9$	0.690	23.3%	0.905	$\mathbf{0.5}\%$
$p_0^* = 0.8$	0.630	21.2%	0.804	$\mathbf{0.5}\%$
$p_0^* = 0.7$	0.570	18.6%	0.698	$\mathbf{0.2}\%$
$p_0^* = 0.6$	0.510	15.0%	0.595	$\mathbf{0.8}\%$
$p_0^* = 0.5$	0.450	10.0%	0.506	$\mathbf{1.2}\%$
	Avg Error	17.6%	Avg Error	$\mathbf{0.64}\%$
$\boldsymbol{\alpha}$=[0.780,0.807], Attractive				
$p_0^* = 0.9$	0.730	18.8%	0.908	$\mathbf{0.9}\%$
$p_0^* = 0.8$	0.670	16.3%	0.804	$\mathbf{0.5}\%$
$p_0^* = 0.7$	0.60	14.3%	0.696	$\mathbf{0.6}\%$
$p_0^* = 0.6$	0.54	10.0%	0.592	$\mathbf{1.3}\%$
$p_0^* = 0.5$	0.480	4.0%	0.493	$\mathbf{1.4}\%$
	Avg Error	12.3%	Avg Error	$\mathbf{0.94}\%$

Table 13: Comparing ***interval estimate values*** of Baseline (Choi et al., 2020) and proposed CLEAM measurement framework on (A) pseudo-generator with sensitive attribute {Young, Attractive}.

Pseudo-Generator				
Test	**Baseline**(Choi et al., 2020)		**(Ours) CLEAM**	
	$\rho_{\text{Base}}(p_0^*)$	$e_\rho(p_0^*)$	$\rho_{\text{CLEAM}}(p_0^*)$	$e_\rho(p_0^*)$
$\boldsymbol{\alpha}$=[0.749,0.852], Young				
$p_0^* = 0.9$	$[0.684, 0.695]$	24.0%	$[0.890, 0.920]$	$\mathbf{2.2}\%$
$p_0^* = 0.8$	$[0.625, 0.635]$	21.9%	$[0.790, 0.810]$	$\mathbf{1.3}\%$
$p_0^* = 0.7$	$[0.565, 0.575]$	19.2%	$[0.690, 0.710]$	$\mathbf{1.4}\%$
$p_0^* = 0.6$	$[0.505, 0.515]$	15.8%	$[0.590, 0.600]$	$\mathbf{1.6}\%$
$p_0^* = 0.5$	$[0.445, 0.455]$	11.0%	$[0.490, 0.500]$	$\mathbf{2.0}\%$
	Avg Error	18.38%	Avg Error	$\mathbf{1.7}\%$
$\boldsymbol{\alpha}$=[0.780,0.807], Attractive				
$p_0^* = 0.9$	$[0.724, 0.736]$	19.5%	$[0.900, 0.920]$	$\mathbf{2.2}\%$
$p_0^* = 0.8$	$[0.665, 0.675]$	16.9%	$[0.790, 0.810]$	$\mathbf{1.3}\%$
$p_0^* = 0.7$	$[0.594, 0.606]$	15.1%	$[0.690, 0.710]$	$\mathbf{1.4}\%$
$p_0^* = 0.6$	$[0.534, 0.546]$	11.0%	$[0.580, 0.600]$	$\mathbf{3.3}\%$
$p_0^* = 0.5$	$[0.475, 0.485]$	5.0%	$[0.490, 0.540]$	$\mathbf{2.0}\%$
	Avg Error	13.5%	Avg Error	$\mathbf{2.0}\%$

C.4 Measuring Varying Degrees of Bias with Additional Attribute Classifier

In this section, we validate CLEAM's versatility with different classifier architecture. In our setup, we utilise MobileNetV2 (Sandler et al., 2018) as in Frankel & Vendrow (2020). Then similar to Sec. 5.3 of the main paper, we utilize a pseudo-Generator with known GT p^* for the Gender and BlackHair attribute, to evaluate CLEAMs effectiveness at determining bias.

As seen in our results in Tab. 14,15, MobileNetV2 demonstrates similar results, on the Baseline and CLEAM, to ResNet-18 model in the main paper. In these results, we observed a significantly large e_μ for the baseline of 1.4% and 9.4% for the Gender and BlackHair attributes, respectively. Whereas, CLEAM reported an e_μ of 0.8% and 0.3%. The same can be observed in e_ρ. We thus demonstrates CLEAM's versatility and ability to be deployed as a post-processing method (without retraining), on models of varying architecture.

Table 14: Comparing ***point estimate values*** of Baseline (Choi et al., 2020), Diversity (Keswani & Celis, 2021) and proposed CLEAM measurement framework on (A) pseudo-generator with sensitive attribute {Gender, BlackHair} and MobileNetV2(Sandler et al., 2018).

(A) Pseudo-Generator						
Test	**Baseline**(Choi et al., 2020)		**Diversity**(Keswani & Celis, 2021)		**(Ours) CLEAM**	
	$\mu_{\text{Base}}(p_0^*)$	$e_\mu(p_0^*)$	$\mu_{\text{Div}}(p_0^*)$	$e_\mu(p_0^*)$	$\mu_{\text{CLEAM}}(p_0^*)$	$e_\mu(p_0^*)$
$\boldsymbol{\alpha}$=[0.980,0.986], Gender						
$p_0^* = 0.9$	0.882	2.0	0.950	5.6%	0.899	$\mathbf{0.1}\%$
$p_0^* = 0.8$	0.786	1.7%	0.785	1.9%	0.800	$\mathbf{0.0}\%$
$p_0^* = 0.7$	0.689	1.6%	0.709	1.2%	0.699	$\mathbf{0.1}\%$
$p_0^* = 0.6$	0.593	1.2%	0.591	1.5%	0.600	$\mathbf{0.0}\%$
$p_0^* = 0.5$	0.497	0.4%	0.481	3.8%	0.502	$\mathbf{0.3}\%$
	Avg Error	1.4%	Avg Error	2.8%	Avg Error	$\mathbf{0.8}\%$
$\boldsymbol{\alpha}$=[0.861,0.916], BlackHair						
$p_0^* = 0.9$	0.782	13.0%	0.803	10.7%	0.899	$\mathbf{0.1}\%$
$p_0^* = 0.8$	0.705	11.8%	0.699	12.6%	0.800	$\mathbf{0.0}\%$
$p_0^* = 0.7$	0.623	10.3%	0.661	5.5%	0.700	$\mathbf{0.0}\%$
$p_0^* = 0.6$	0.550	8.3%	0.609	1.5%	0.600	$\mathbf{0.0}\%$
$p_0^* = 0.5$	0.478	3.8%	0.521	4.2%	0.506	$\mathbf{1.2}\%$
	Avg Error	9.4%	Avg Error	6.9%	Avg Error	$\mathbf{0.3}\%$

Table 15: Comparing ***Interval estimate values*** of Baseline (Choi et al., 2020), Diversity (Keswani & Celis, 2021) and proposed CLEAM measurement framework on (A) pseudo-generator with sensitive attribute $\{\texttt{Gender}, \texttt{BlackHair}\}$ and MobileNetV2(Sandler et al., 2018)

(A) Pseudo-Generator						
Test	**Baseline**(Choi et al., 2020)		**Diversity**(Keswani & Celis, 2021)		**(Ours) CLEAM**	
	$\rho_{\text{Base}}(p_0^*)$	$e_\rho(p_0^*)$	$\rho_{\text{Div}}(p_0^*)$	$e_\rho(p_0^*)$	$\rho_{\text{CLEAM}}(p_0^*)$	$e_\rho(p_0^*)$
$\boldsymbol{\alpha}$=[0.980,0.986], Gender						
$p_0^* = 0.9$	$[0.879, 0.885]$	2.3%	$[0.913, 0.986]$	9.6%	$[0.995, 0.902]$	$\mathbf{0.5}\%$
$p_0^* = 0.8$	$[0.782, 0.790]$	2.3%	$[0.762, 0.809]$	4.8%	$[0.794, 0.804]$	$\mathbf{0.6}\%$
$p_0^* = 0.7$	$[0.685, 0.693]$	2.1%	$[0.696, 0.722]$	3.1%	$[0.694, 0.704]$	$\mathbf{0.8}\%$
$p_0^* = 0.6$	$[0.585, 0.597]$	2.5%	$[0.581, 0.612]$	3.3%	$[594, 0.605]$	$\mathbf{1.0}\%$
$p_0^* = 0.5$	$[0.491, 0.502]$	1.8%	$[0.473, 0.490]$	5.5%	$[495, 0.507]$	$\mathbf{1.1}\%$
	Avg Error	2.2%	Avg Error	5.3%	Avg Error	$\mathbf{0.8}\%$
$\boldsymbol{\alpha}$=[0.861,0.916], BlackHair						
$p_0^* = 0.9$	$[0.777, 0.787]$	13.6%	$[0.791, 0.815]$	12.1%	$[0.893, 0.900]$	$\mathbf{0.7}\%$
$p_0^* = 0.8$	$[0.699, 0.710]$	12.6%	$[0.686, 0.713]$	14.2%	$[0.793, 0.807]$	$\mathbf{0.9}\%$
$p_0^* = 0.7$	$[0.618, 0.628]$	11.7%	$[0.643, 0.68]$	8.2%	$[0.694, 0.706]$	$\mathbf{0.9}\%$
$p_0^* = 0.6$	$[0.544, 0.556]$	9.3%	$[0.604, 0.614]$	2.3%	$[0.593, 0.608]$	$\mathbf{1.3}\%$
$p_0^* = 0.5$	$[0.472, 0.484]$	5.6%	$[0.506, 0.536]$	7.2%	$[0.498, 0.514]$	$\mathbf{2.4}\%$
	Avg Error	10.6%	Avg Error	8.8%	Avg Error	$\mathbf{1.2}\%$

D DETAILS OF HYPER-PARAMETER SETTINGS

Attribute Classifier C_u. In our experiments, we utilized a Resnet-18/34 (He et al., 2016), MobileNetv2 (Sandler et al., 2018) and VGG-16. The respective datasets *i.e.* CelebA dataset (Liu et al., 2015) and CelebA-HQ dataset (Lee et al., 2020) are then segment into {Train,Test,Validate} with respect to the ratio {80%,10%,10%}, where each segmentation of the dataset contains uniform distribution *w.r.t.* the queried sensitive attribute. The classifiers are then trained with the training datasets and the α are evaluated with the validation dataset. Each classifier is trained with an Adam optimiser(Kingma & Ba, 2017) with a learning rate=$1e^{-3}$, Batch size=64 and input dim=64x64 from the CelebA dataset (Liu et al., 2015) and dim=128x128 from the CelebA-HQ dataset (Lee et al., 2020).

Generator G_ϕ. As mentioned in the main paper, we utilised Choi et al. (2020) setup[1] for the training of our imp-weighted and unweighted GANs. With this, we replicate their hyperparameter selection of 64 x 64 celebA (Liu et al., 2015) images with a learning rate=$2e^{-4}$, $\beta_1 = 0$, $\beta_2 = 0.99$ and four discriminator steps per generator step. We utilise a single RTX3090 for the training of our models.

Batch Size s. In our experiments, where we generated $n = 400$ samples with our pseudo-generator setup for s batches, followed by applying CLEAM to approximate p^*, we found that the batch size $s = 30$ to be suitable to approximate the known GT p^*. Increasing s did not lead to noticeable improvement in the error. However, from our results in Tab. 17 and 18, decreasing s or n did result in some degradation in both the Baseline and CLEAM. In particular, we attribute this increase in error to two reasons. Firstly, the deviation between the statistical estimate and the model estimate, as discussed in Sec. B.1. Secondly, the poorer estimate of $\mu_{\texttt{base}}$, which then in turn influences $\mu_{\texttt{CLEAM}}$.

Computational Time. In our main paper, we note that CLEAM is a lightweight correction to the existing baseline method, that requires no additional parameter to be computed during evaluation. To support this, we evaluated the computational time for the Baseline, Diversity and our proposed CLEAM. Our results in Tab. 16 shows that there is only a small difference in computational time (3-4s) between the Baseline and our proposed CLEAM. This difference is solely to facilitate the computation of Algo. 1.

Table 16: Computation time for estimating p^* with hyper-parameters s=30 and n=400 for the Baseline Choi et al. (2020), Diversity Keswani & Celis (2021) and our proposed CLEAM. For hardware, we utilize a single RTX3090 and reported the average computational time, in seconds, of 5 consecutive runs.

	Baseline Choi et al. (2020)	Diversity Keswani & Celis (2021)	CLEAM (Ours)
Computational Time on **CelebA, 64x64** , s	97	600	100
Computational Time on **CelebA-HQ, 128x128**, s	132	820	136

[1] https://github.com/ermongroup/fairgen

Table 17: As per Tab. 1 of tha main paper, we compare the **point estimates** of Baseline (Choi et al., 2020), Diversity (Keswani & Celis, 2021) and our proposed CLEAM measurement framework in evaluating the p^* of datasets sampled from (A) StyleGAN2 (Karras et al., 2019) and (B) StyleSwin (Zhang et al., 2021). In this setup, each $\hat{p}$ instead utilizes $n = 400$ samples and is evaluated for a batch-size of $s = 20$. We repeat this for 5 experimental runs and report the mean error rate, per Eqn. 1.

(A) StyleGAN2							
Classifier	**GT**	**Baseline**(Choi et al., 2020)		**Diversity**(Keswani & Celis, 2021)		**CLEAM (Ours)**	
		$\mu_{\text{Base}}(\hat{p}_0)$	$e_\mu(p_0^*)(\downarrow)$	$\mu_{\text{Div}}(\hat{p}_0)$	$e_\mu(p_0^*)(\downarrow)$	$\mu_{\text{CLEAM}}(\hat{p}_0)$	$e_\mu(p_0^*)(\downarrow)$
Gender							
ResNet-18		0.605	5.7%	—	—	0.633	**1.4%**
ResNet-34	p_0^*=0.642	0.589	8.2%	—	—	0.628	**2.2%**
MobileNetv2		0.601	6.4%	—	—	0.631	**1.7**%
VGG-16		0.528	17.8%	0.547	14.8%	0.632	**1.6**%
		Avg Error	9.5%	Avg Error	14.8%	Avg Error	**1.7**%
BlackHair							
ResNet-18		0.595	7.5%	—	—	0.637	**0.9%**
ResNet-34	p_0^*=0.643	0.562	12.6%	—	—	0.640	**0.5**%
MobileNetv2		0.577	10.3%	—	—	0.637	**0.9**%
VGG-16		0.600	6.7%	0.581	9.6%	0.637	**0.9**%
		Avg Error	9.3%	Avg Error	9.6%	Avg Error	**0.8**%
(B) StyleSwin							
Gender							
ResNet-18		0.617	5.9%	—	—	0.645	**1.7%**
ResNet-34	p_0^*=0.656	0.606	7.6%	—	—	0.645	**1.7**%
MobileNetv2		0.620	5.5%	—	—	0.652	**0.6**%
VGG-16		0.552	15.9%	0.560	14.6%	0.665	**1.4**%
		Avg Error	8.7%	Avg Error	14.6%	Avg Error	**1.4**%
BlackHair							
ResNet-18		0.610	8.7%	—	—	0.657	**1.6%**
ResNet-34	p_0^*=0.668	0.577	13.6%	—	—	0.658	**1.5**%
MobileNetv2		0.594	11.1%	—	—	0.657	**1.6**%
VGG-16		0.621	7.0%	0.606	9.3%	0.672	**0.6**%
		Avg Error	10.1%	Avg Error	9.3%	Avg Error	**1.3**%

Table 18: As per Tab. 1 of tha main paper, we compare the **point estimates** of Baseline (Choi et al., 2020), Diversity (Keswani & Celis, 2021) and our proposed CLEAM measurement framework in evaluating the p^* of datasets sampled from (A) StyleGAN2 (Karras et al., 2019) and (B) StyleSwin (Zhang et al., 2021). In this setup, each $\hat{p}$ instead utilizes $n = 200$ samples and is evaluated for a batch-size of $s = 30$. We repeat this for 5 experimental runs and report the mean error rate, per Eqn. 1.

(A) StyleGAN2							
Classifier	**GT**	**Baseline**(Choi et al., 2020)		**Diversity**(Keswani & Celis, 2021)		**CLEAM (Ours)**	
		$\mu_{\text{Base}}(\hat{p}_0)$	$e_\mu(p_0^*)(\downarrow)$	$\mu_{\text{Div}}(\hat{p}_0)$	$e_\mu(p_0^*)(\downarrow)$	$\mu_{\text{CLEAM}}(\hat{p}_0)$	$e_\mu(p_0^*)(\downarrow)$
Gender							
ResNet-18		0.601	6.4%	—	—	0.629	**2.0%**
ResNet-34	p_0^*=0.642	0.590	8.1%	—	—	0.629	**2.0%**
MobileNetv2		0.601	6.4%	—	—	0.631	**1.7**%
VGG-16		0.527	18.0%	0.551	14.2%	0.631	**1.7**%
		Avg Error	9.7%	Avg Error	14.2%	Avg Error	**1.9**%
BlackHair							
ResNet-18		0.598	7.0%	—	—	0.640	**0.5%**
ResNet-34	p_0^*=0.643	0.560	12.9%	—	—	0.640	**0.5**%
MobileNetv2		0.572	11.0%	—	—	0.632	**1.7**%
VGG-16		0.605	6.0%	0.580	9.8%	0.642	**1.6**%
		Avg Error	9.2%	Avg Error	9.8%	Avg Error	**0.8**%
(B) StyleSwin							
Gender							
ResNet-18		0.615	6.3%	—	—	0.643	**2.0%**
ResNet-34	p_0^*=0.656	0.612	6.7%	—	—	0.651	**0.7**%
MobileNetv2		0.624	4.9%	—	—	0.656	**0.0**%
VGG-16		0.551	16.0%	0.572	12.8%	0.665	**1.4**%
		Avg Error	8.5%	Avg Error	12.8%	Avg Error	**1.0**%
BlackHair							
ResNet-18		0.608	9.0%	—	—	0.655	**1.9%**
ResNet-34	p_0^*=0.668	0.578	13.5%	—	—	0.659	**1.3**%
MobileNetv2		0.592	11.4%	—	—	0.655	**1.9**%
VGG-16		0.621	7.0%	0.608	9.0%	0.673	**0.7**%
		Avg Error	10.2%	Avg Error	9.0%	Avg Error	**1.5**%

E RELATED WORK

FAIRNESS IN GENERATIVE MODELS. Fairness in machine learning is mostly studied for discriminative learning, where usually the objective is to handle a classification task independent of a sensitive attribute in the input data, *e.g.* making a hiring decision independent of the applicant `Gender`. However, the definition of fairness is quite different for generative learning, where it is considered as equal representation/generation probability *w.r.t.* a sensitive attribute. Because of this difference, the conventional fairness metrics used for classification, like Equalised Odds, Equalised Opportunity (Hardt et al., 2016) and Demographic Parity (Feldman et al., 2015), cannot be applied to generative models. Instead, the similarity between the probability distribution of the generated sample *w.r.t.* a sensitive attribute (p^*) and a target distribution $\bar{p}$ (a uniform distribution) (Choi et al., 2020) is utilized as fairness metric. See Supp. A.3 for details.

EXISTING WORKS ON FAIR GENERATIVE MODELS. Existing works focus on ***bias mitigation*** in generative models. The importance reweighting algorithm is proposed in (Choi et al., 2020) where a re-weighting algorithm favours a reference fair dataset *w.r.t.* the sensitive attribute in-place of a larger biased dataset. Frankel & Vendrow (2020) introduces the concept of prior modification, where an additional smaller network is added to modify the prior of a GAN to achieve a fairer output. Tan et al. (2020) learns the latent input space *w.r.t.* the sensitive attribute, which they can later sample accordingly to achieve a fair output. MaGNET (Humayun et al., 2022) demonstrates that enforcing uniformity in the latent feature space of a GAN, through a sampling process, improves fairness. In all of these works, the focus is on improving fairness of the generative model (where the performance of the model is measured with a framework, in which the inaccuracies in the attribute classifier has been ignored). However, our proposed CLEAM method focuses on improving ***fairness measurement***, by compensating for the inaccuracies in the attribute classifier through a statistical model. Therefore, it can be used to evaluate the bias mitigation algorithms more accurately.

EQUAL REPRESENTATION. Some literature also use a similar notion of equal representation (used in generative models) to address fairness. In Chierichetti et al. (2017), fair clustering is proposed by enforcing the clusters to represent each attribute equally. Celis et al. (2018) proposes fair data summarization to mitigate the bias in creating a representative subset for a given dataset, while handling the trade-offs between fairness and diversity during sampling. However, unlike our setup, these works assume to have access to the attribute labels. Meanwhile, in data mining, a similar problem was recently studied. Given a large dataset of unlabelled mined data, the objective is to evaluate the disparity of the dataset *w.r.t.* an attribute. To do this, an evaluation framework called diversity (Keswani & Celis, 2021) was introduced. To measure this, a pre-trained classifier is used as a feature extractor. The unlabelled dataset is then compared against a controlled reference dataset (with known labels) via a similarity algorithm.

CLASSIFIER CALIBRATION. The proposed CLEAM can be seen from a classifier calibration point of view as it refines the output of the classifier. However, CLEAM should not be mistaken with conventional calibration algorithms, *e.g.* temperature scaling (Guo et al., 2017), Platt Scaling (Platt et al., 1999) and Isotonic regression (Nyberg & Klami, 2021). Unlike these algorithms that concern themselves with the confidence of prediction, CLEAM focuses on sensitive attribute distribution, thereby making these algorithms ineffective.

More specifically, conventional classifier calibration methods usually work on soft labels (probabilities). Note that in our framework, the $argMax$ is applied to the output probabilities to determine the hard label. Therefore, in our application that deals with hard labels, regular classification techniques are less effective. To investigate this, we conduct a few calibration experiment by applying some popular classifier calibration techniques; temperature scaling(T-scaling) (Guo et al., 2017), Isotonic Regression(Nyberg & Klami, 2021) and Platt Scaling(Platt et al., 1999) on a pre-trained ResNet-18(He et al., 2016) senstive attribute classifier. In Fig. 6, we see that T-scaling is the most effective in correcting the calibration curve to the ideal Ref line. Note that, this Ref line indicates that the classifier is perfectly calibrated *w.r.t.* the soft labels.

Next, using the pseudo-generator from Sec. 5 of the main paper, we utilised the calibrated classifiers earlier and compare them against CLEAM (which was applied on an uncalibrated model). **In our results,** seen in Fig. 7, we observe that these traditional calibration methods are less effective in correcting the sensitive attribute distribution error. In fact, methods like Platt scaling worsen the error, and T scaling —which is shown in (Guo et al., 2017) and our experiment to be one of the

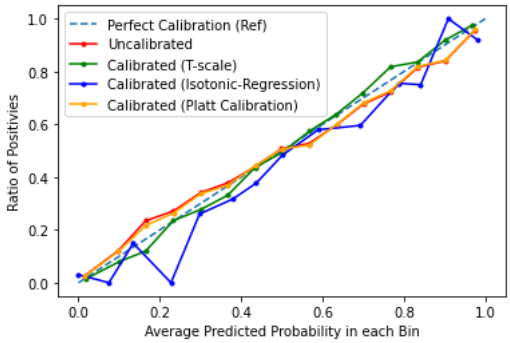


Figure 6: Calibration Curve on ResNet-18 for `Attractive` attribute. We observe that the T-scaling is the most effective technique in improving soft label calibration and Isotonic regression the worst. However, this same trend does not follow in the hard label errors of Fig 7.

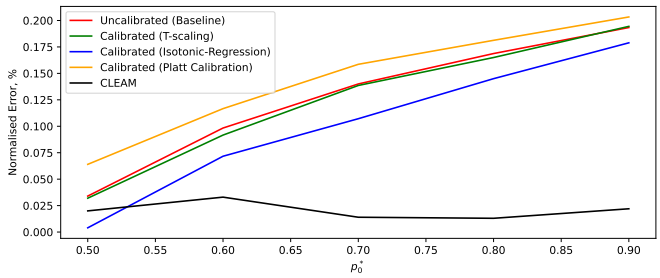


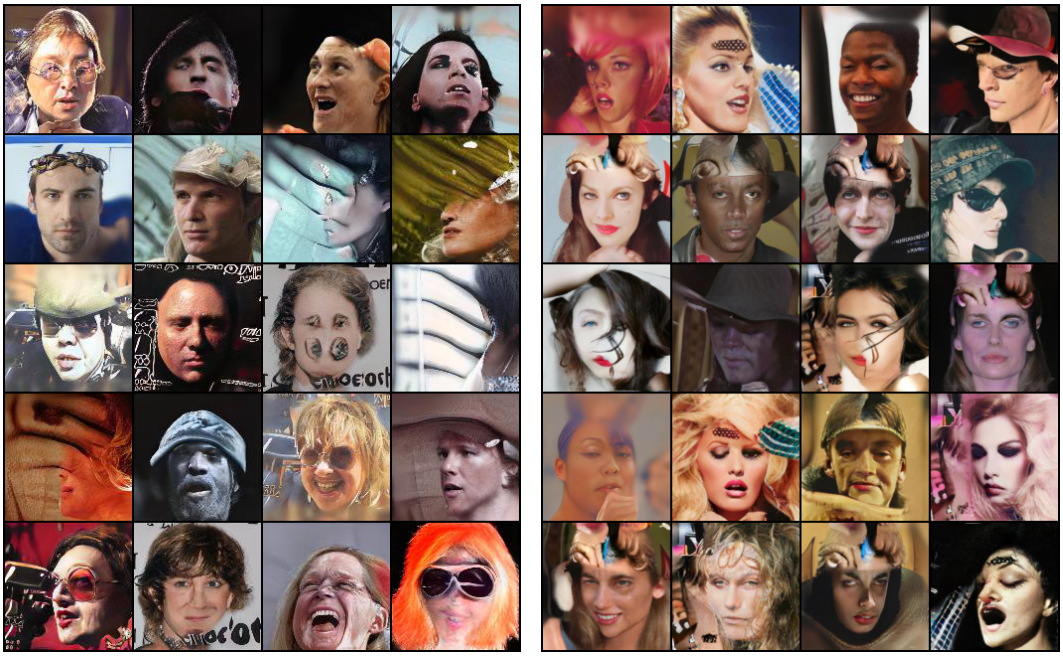
Figure 7: **Comparing Calibration Techniques**: Using the pseudo-generator, we compare CLEAM against well known calibration techniques, overall we observe that previous techniques are significantly less effective, achieving an average error of; T-Scaling: 12.4%, Isotonic Regression: 10.1%, Platt Calibration: 14.5% and uncalibrated (baseline): 12.4% against CLEAM: 2.0%

most effective traditional calibration methods— does not change class predictions (hard labels), but merely perturb the soft labels. This demonstrates that traditional calibration technique are not direct correlation to hard label calibration, which CLEAM aims to address.

F DETAILS OF THE NEW DATASET: LABELED GAN IMAGES

In this section, we provide more information on our new dataset, containing labeled samples from StyleGAN2[2] (Karras et al., 2019) and StyleSwin [3] (Zhang et al., 2021) trained on CelebA-HQ (Lee et al., 2020). More specifically, our dataset contains ≈9k randomly generated samples based on the original saved weights and codes of the respective GANs. These samples are then hand labeled *w.r.t.* the sensitive attribute `Gender` and `Blackhair`, which is perceived to be consistent with human perception. Then with these labeled datasets, we can approximate the ground truth sensitive attribute distribution, p^*, of the respective GANs.

DATASET LABELING PROTOCOL. To ensure high-quality samples and labels in our dataset, we passed out the dataset through Amazon Mechanical Turk, where labelers were given detailed guidelines and examples in identifying the individual sensitive attributes. In addition to the sensitive attribute option *e.g.* `Gender(Male)` or `Gender(Female)`, labelers were also given an "unidentifiable" option which they were instructed to select for low quality samples, as per Fig, 8. We repeated this process for 3 runs *s.t.* each sample had the opinions of three independent labelers. Finally, each sample was assigned the label that the majority had selected, this for example includes male, female, or unidentifiable, for the attribute `Gender`. We discard the samples that had been labeled unidentifiable and were left with a high-quality dataset as per Fig. 9 and 10.



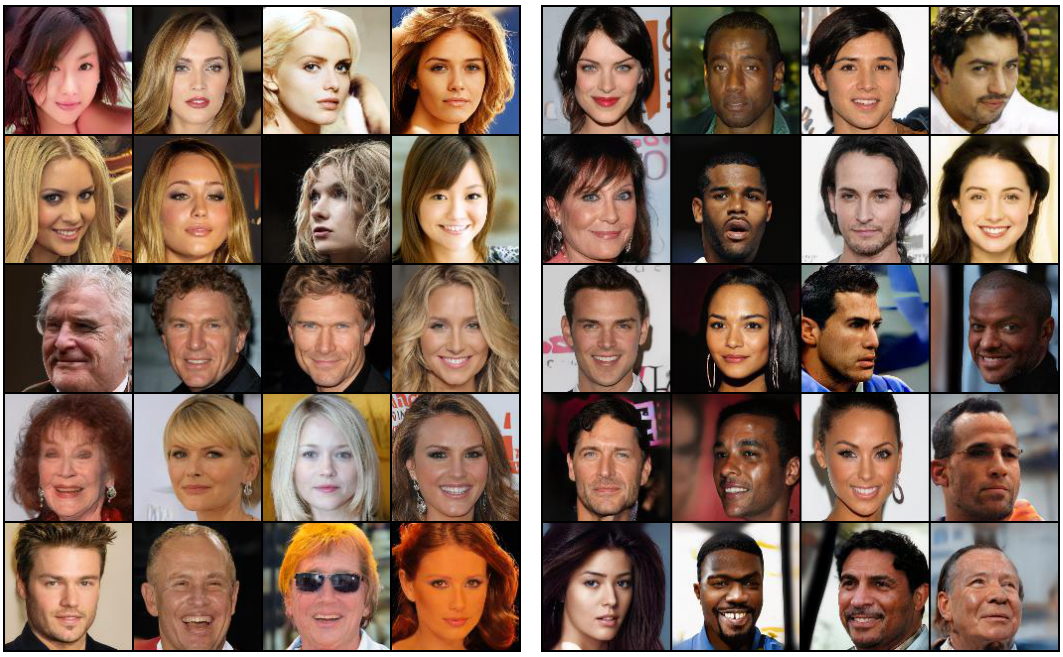
(a) StyleGAN2 (b) StyleSwin

Figure 8: Examples of rejected samples during hand-labeling

[2]https://github.com/NVlabs/stylegan2-ada-pytorch
[3]https://github.com/microsoft/StyleSwin

(a) Gender (Female) Samples (b) Gender (Male) Samples

Figure 9: Examples of samples *w.r.t.* Gender attribute



(a) no-BlackHair Samples (b) BlackHair Samples

Figure 10: Examples of samples *w.r.t.* BlackHair attribute