
AugGen: Synthetic Augmentation Can Improve Discriminative Models

Parsa Rahimi^{1,2} Damien Teney² Sébastien Marcel^{2,3}

Abstract

The increasing dependence on large-scale datasets in machine learning introduces significant privacy and ethical challenges. Synthetic data generation offers a promising solution; however, most current methods rely on external datasets or pre-trained models, which add complexity and escalate resource demands. In this work, we introduce a novel self-contained synthetic augmentation technique that strategically samples from a conditional generative model trained exclusively on the target dataset. This approach eliminates the need for auxiliary data sources. Applied to face recognition datasets, our method achieves 1–12% performance improvements on the IJB-C and IJB-B benchmarks. It outperforms models trained solely on real data and exceeds the performance of state-of-the-art synthetic data generation baselines. Notably, these enhancements often surpass those achieved through architectural improvements, underscoring the significant impact of synthetic augmentation in data-scarce environments. These findings demonstrate that carefully integrated synthetic data not only addresses privacy and resource constraints but also substantially boosts model performance. The project page is available at: <https://parsa-ra.github.io/auggen>.

1. Introduction

As machine learning increasingly relies on application-specific data, the demand for high-quality, accurately labeled datasets poses significant challenges. Privacy, legal, and ethical concerns amplify these difficulties, particularly in sensitive areas like human face images. A popular solution is synthetic data generation (Wood et al., 2021; Azizi et al., 2023; Rahimi et al., 2024), (DeAndres-Tame et al., 2024; Bae et al., 2023), which leverages methods such as

¹EPFL, Switzerland ²Idiap, Switzerland ³UNIL, Switzerland. Correspondence to: Parsa Rahimi <parsa.rahiminoshanagh@epfl.ch>.

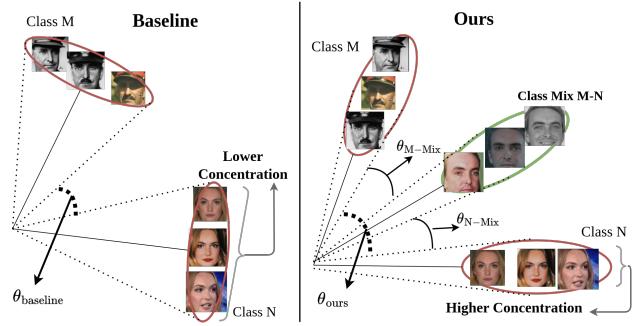


Figure 1. Core idea of AugGen. AugGen boosts the model’s overall discriminative capabilities without requiring external datasets or pre-trained networks. To achieve this, we propose a novel sampling strategy using a conditional diffusion model—trained exclusively on the discriminator’s **original data**—this enables the generation of **synthetic** “mixes” of source classes. Incorporating these **synthetic** samples into the discriminator’s training, results in higher intra-class compactness and greater inter-class separation ($\theta_{\text{ours}} > \theta_{\text{baseline}}$) than models trained solely on the **original data**.

3D-rendering graphics and generative models (e.g., GANs and diffusion models). Notably, synthetic data can surpass real data in model performance, as shown by (Wood et al., 2021), where 3D-rendered face models with precise labels outperformed real-data-based models in tasks like face landmark localization and segmentation, highlighting the advantages of data synthesis, especially for tasks requiring dense annotations. Image generative models remain underutilized despite rapid advances in VAEs (Kingma, 2013), GANs (Goodfellow et al., 2020; Karras et al., 2019; 2021), and Diffusion models (Song et al., 2020; Karras et al., 2022; 2024; Hoogeboom et al., 2023; Gu et al., 2024). Comparisons of generative models often use metrics like Fréchet Distance (FD) (Stein et al., 2023; Heusel et al., 2017), which measure similarity to training data, or subjective user preferences for text-to-image tasks (Esser et al., 2024).

Currently, synthetic data generation involves training large-scale generative models (Rombach et al., 2022) on datasets such as LAION-5B (Schuhmann et al., 2022), then refining them via fine-tuning, prompt engineering, or textual inversion (Azizi et al., 2023; Trabucco et al., 2024). This approach also applies to Face Recognition (FR), where synthetic data aims to mitigate privacy and ethical concerns.

However, most methods still rely on large face datasets (which carry their own privacy issues) and auxiliary models, offering no clear advantage over existing real datasets. For instance, DCFace (Kim et al., 2023) generates diverse face images from multiple identities and uses robust FR systems and auxiliary networks to filter and balance samples. It remains unclear whether performance gains stem from the datasets, the generative models, or other factors—though larger, more diverse data typically improves results. Contrary to current trends, we advocate using generative models as an augmentation tool for FR training rather than replacing real datasets. Two key factors motivate this stance:

1. Synthetic datasets generated by diffusion models often leak training data (Li et al., 2024; Carlini et al., 2023; Shahreza & Marcel, 2024), offering no clear benefit over existing priors (Kim et al., 2023; Boutros et al., 2023; Melzi et al., 2023; Sun et al., 2024; Xu et al., 2024).
2. Since responsible FR datasets are scarce and difficult to collect, we aim to boost performance with limited real data, thereby narrowing the gap between small-scale and large-scale training sets.

For doing this, as shown in Figure 1, we aim to create mixed classes that blend features from two or more source classes while preserving distinct identities. Our method can be considered a general formulation of MixUp (Zhang, 2017) and CutMix (Yun et al., 2019), but instead of cropping or blending images in label and image space, we are using a generator to create a new novel class. This approach improves margin-based losses used to train SOTA FR systems. To realize this, we train a generative model ($G = p(\mathbf{X} | \mathbf{y})$) on a single real dataset and leverage a discriminative model ($M = p(\mathbf{y} | \mathbf{X})$) trained on the same dataset. We introduce a simple yet novel sampling strategy to synthesize new samples. Experimental results show that augmenting real data with these synthetic samples enhances the discriminator’s performance. As a general augmentation technique, our method can be particularly beneficial in critical applications where data is scarce or difficult to acquire.

Our main contribution is to validate this hypothesis in the context of face recognition (FR):

H1: A generative model can boost the performance of a downstream discriminative model with an appropriate informed sampling, and augmenting the resulting data with the original data that was used for training the generative and discriminative models.

Our contributions are summarized as follows.

- We propose a simple yet effective sampling technique that strategically conditions a generative model to pro-

duce beneficial samples, enhancing the discriminator’s training process (Subsection 3.3).

- We show that mixing our AugGen data with real samples **often surpasses even architectural-level improvements**, underscoring that **synthetic dataset generation can be as impactful as architectural advances** (Section 4).
- We demonstrate that AugGen training can be as effective as adding up to **1.7×** real samples, **reducing** the need for more face images while preserving performance (Subsection 4.3).
- We show that current generative metrics (e.g., FD, KD) are poorly correlated with downstream discriminative performance, emphasizing the need for improved proxy metrics (Appendix F).

To the best of our knowledge, this is the first demonstration of generative image models effectively enhancing augmentation at this scale without relying on auxiliary models or external datasets.

2. Related Work

Synthetic Data in Computer Vision. For a smaller number of class variations, (e.g., 2 or 3 classes for classification target), authors in (Frid-Adar et al., 2018) train separate generative models. This approach is not scalable for a higher number of classes and variations of our target (e.g., we have thousands of classes for training an FR system). In (Azizi et al., 2023), the authors fine-tuned pre-trained diffusion models on ImageNet classes after training on large text-image datasets, demonstrating improved performance on this benchmark through the synthesis of new samples. Authors in (Wood et al., 2021) leveraged 3D rendering engines and computer graphics. Here as they have access to the underlying 3D Morphable Face Model (3DMM) (Blanz & Vetter, 1999) and closed-form back projection to the image plane, the authors introduced a Face Dataset for landmark detection, localization and also semantic segmentation task. By design, as the method has access to accurate labels in such 3D rendered datasets authors demonstrated a slight advantage on the models trained on their proposed dataset when it is evaluated against real-world datasets.

Synthetic Data for Face Recognition. SynFace (Qiu et al., 2021) employs DiscoFaceGAN (Deng et al., 2020) for controllable identity mixup (Zhang, 2017), training with a FR network on MS-Celeb1M (Guo et al., 2016), 3DMM, keypoint matching, and other priors. DCFace (Kim et al., 2023) uses dual-condition latent diffusion models (LDMs)—one for style and one for identity—trained on CASIA-WebFace (Yi et al., 2014), then filters generated images with auxiliary demographic classifiers and a strong

FR system. In (Sevastopolskiy et al., 2023), a StyleGAN2-ADA (Karras et al., 2020) is pre-trained on a large unlabeled multi-ethnic dataset, and an encoder transfers latent-space mappings to an FR network to mitigate bias.

GANDiffFace (Melzi et al., 2023) combines StyleGAN3 (Karras et al., 2021) and Stable Diffusion (Rombach et al., 2022) (trained on LAION-5B (Schuhmann et al., 2022)), along with DreamBooth (Ruiz et al., 2023), for increased intra-class variation. IDiff-face (Boutros et al., 2023) conditions a latent diffusion model on FR embeddings from a network trained on MS1Mv2 (Deng et al., 2019). ID³ (Xu et al., 2024) similarly conditions a diffusion model on face attributes and an FR network trained on MS1Mv2, using both CASIA-WebFace and FFHQ (Karras et al., 2019) for training. Unlike DCFace’s post-processing, ID³ incorporates identity/attribute information directly into the generation process. Note that using MS1Mv2 yields higher FR performance than CASIA-WebFace (Deng et al., 2019).

DigiFace1M (Bae et al., 2023) generates diverse 3D-rendered faces with varied poses, expressions, and lighting. In (Rahimi et al., 2024), off-the-shelf image-to-image translation (Theiss et al., 2022; Zhou et al., 2022) further boosts DigiFace1M’s performance despite lacking explicit identity information. Additional prior work is discussed in Appendix A.

3. Methodology

Figure 2 illustrates our approach, where a discriminative model M_{orig} and a generator G are trained on the same dataset. By strategically sampling from G , we generate synthetic images forming new classes, augmenting the original dataset. We first define the problem for the discriminator and generator in Subsection 3.1 and Subsection 3.2, then introduce our key contribution: generating new classes (Finding Weights, Figure 2(c)) to complement real datasets with synthetic images.

3.1. Discriminative Model

Assume a dataset $\mathbf{D}_{\text{orig}} = \{(\mathbf{X}_i, y_i)\}_{i=0}^{k-1}$, where each $\mathbf{X}_i \in \mathbb{R}^{H \times W \times 3}$ and $y_i \in \{0, \dots, l-1\}$ ($l < k$). The goal is to learn a discriminative model $f_{\theta_{\text{dis}}} : \mathbf{X} \rightarrow \mathbf{y}$ that estimates $p(\mathbf{y}|\mathbf{X})$ (e.g., on ImageNet (Russakovsky et al., 2015) or CASIA-WebFace (Yi et al., 2014)). Typically, similar images have closer features under a measure m (e.g., cosine distance). We train $f_{\theta_{\text{dis}}}$ via empirical risk minimization:

$$\theta_{\text{dis}}^* = \arg \min_{\theta_{\text{dis}} \in \Theta_{\text{dis}}} \mathbb{E}_{(\mathbf{X}, y) \sim \mathbf{D}_{\text{orig}}} [\mathcal{L}_{\text{dis}}(f_{\theta_{\text{dis}}}(\mathbf{X}), y)], \quad (1)$$

where \mathcal{L}_{dis} is typically cross-entropy, and \mathbf{h}_{dis} denotes hyperparameters (e.g., learning rates). The resulting model $M_{\text{orig}} = f_{\theta_{\text{dis}}^*}$ is shown in Figure 2(a).

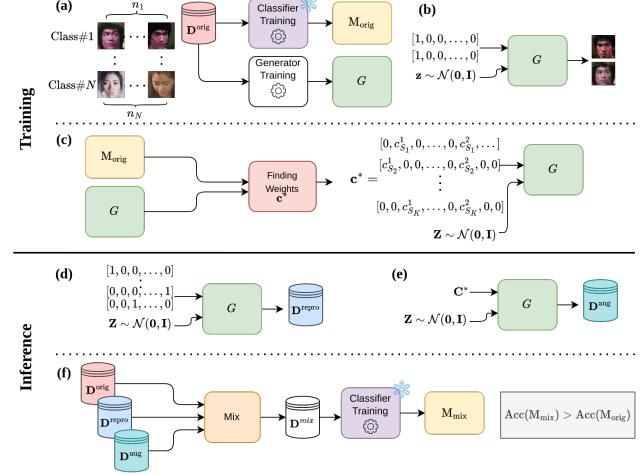


Figure 2. Overview diagram of AugGen: (a) A labeled dataset, \mathbf{D}^{orig} , is used to train a class-conditional generator, $G(\mathbf{Z}, \mathbf{c})$, and a discriminative model, M_{orig} . (b,d) Reproduced dataset, $\mathbf{D}^{\text{repro}}$, closely mimics \mathbf{D}^{orig} under the original conditions. (c) We find new condition vectors, \mathbf{C}^* , to generate an augmented dataset, \mathbf{D}^{aug} , using the generator. (f) Augmenting \mathbf{D}^{orig} with \mathbf{D}^{aug} boosts M_{orig} performance without auxiliary datasets or models.

3.2. Generative Model

Generative models seek to learn the data distribution, enabling the generation of new samples. We use diffusion models (Song et al., 2020; Anderson, 1982), which progressively add noise to data and train a denoiser S . Following (Karras et al., 2022; 2024), S is learned in two stages. First, for a given noise level σ , we add noise \mathbf{N} to $E_{\text{VAE}}(\mathbf{X})$ (or \mathbf{X} directly in pixel-based diffusion) and remove it via:

$$\mathcal{L}(S_{\theta_{\text{den}}}; \sigma) = \mathbb{E}_{(\mathbf{X}, y) \sim \mathbf{D}^{\text{orig}}, \mathbf{N} \sim \mathcal{N}(0, \sigma)} [\|S_{\theta_{\text{den}}}(E_{\text{VAE}}(\mathbf{X}) + \mathbf{N}; \mathbf{c}(y), \sigma) - \mathbf{X}\|_2^2], \quad (2)$$

where $\mathbf{c}(y)$ denotes the class condition, and $E_{\text{VAE}}(\cdot)$ and $D_{\text{VAE}}(\cdot)$ are optional VAE encoder and decoder. In the second stage, we sample different noise levels and minimize:

$$\theta_{\text{den}}^* = \arg \min_{\theta_{\text{den}} \in \Theta_{\text{den}}} \mathbb{E}_{\sigma \sim \mathcal{N}(\mu, \sigma^2)} [\lambda_\sigma \mathcal{L}(S_{\theta_{\text{den}}}; \sigma)], \quad (3)$$

where λ_σ weights each noise scale. Latent diffusion (Rombach et al., 2022) conducts denoising in a compressed latent space, reducing computational cost for high-resolution data.

3.3. Class Mixing

In our formulation, \mathbf{c} is one-hot encoded for each label in \mathbf{D}^{orig} , then mapped to the denoiser’s condition space. After training the conditional denoiser $S_{\theta_{\text{den}}}$ (Figure 2, (c)) via Equation 3, we can sample from the generator in two ways:

1. Use the same one-hot vectors as in training, producing samples similar to \mathbf{D}^{orig} . As an example, when pass-

ing the one-hot vector for the first class, the generator synthesizes samples that resemble this class (Figure 2, (d)), collectively forming D^{repro} .

2. Apply novel condition vectors \mathbf{c}^* different from those used during training.

We explore combining known conditions to synthesize entirely new classes, aiming to increase inter-class separation and feature compactness as presented in Figure 1. By leveraging the previously trained M_{orig} , these additional samples can make M_{mix} (*i.e.*, discriminator trained on the mix of real and generated data) better across diverse benchmarks. Given two classes i and j with one-hot vectors \mathbf{c}^i and \mathbf{c}^j , we construct a new class condition via

$$\mathbf{c}^* = \alpha \mathbf{c}^i + \beta \mathbf{c}^j, \quad (4)$$

We denote the trained denoiser’s generation process by G , so $\mathbf{X}^i = G(\mathbf{Z}, \mathbf{c}^i)$ uses noise $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and condition \mathbf{c} to iteratively denoise the input. To find suitable α and β , we formulate the problem as a grid search, aiming for dissimilarity to classes i and j while preserving class coherence for repeated samples from $G(\mathbf{Z}, \mathbf{c}^*)$. We set the α and β to some possible combinations in a linear space of the values between 0.1 to 1.1. For example, possible combinations would be $\alpha = 0.3, \beta = 0.5$ or $\alpha = 1.1, \beta = 0.4$. We denote \mathbb{W} , the set which contains possible values of α and β . We also select some subset of \mathbb{L} and call it \mathbb{L}_s , for the set to contain some specific classes. Then we randomly select two values from the \mathbb{L}_s namely i and j . Later for each $(\alpha, \beta) \in \mathbb{W}$ we apply the Equation 4, to get the \mathbf{c}^* . We generate three types of images. The first two is the reproduction dataset, D^{repro} as before by setting the conditions to \mathbf{c}^i and \mathbf{c}^j , to get $\mathbf{X}^i = G(\mathbf{Z}, \mathbf{c}^i)$ and $\mathbf{X}^j = G(\mathbf{Z}, \mathbf{c}^j)$. Finally the third one is $\mathbf{X}^* = G(\mathbf{Z}, \mathbf{c}^*)$. By passing the generated images to the $f_{\theta_{\text{dis}}^*}$ (*i.e.*, our discriminator which was trained on the D^{orig}) we get the features, $\mathbf{e}^i, \mathbf{e}^j$ and \mathbf{e}^* respectively. We seek to maximize the dissimilarity between generated images so that we can treat the new sample \mathbf{X}^* as a new class. For this, we use a dissimilarity measure, m_d which the *higher* the absolute value it produces the more dissimilar the inputs are. We calculate this measure for each of the reproduced images of the existing classes with respect to the new class, $d_i = m_d(\mathbf{e}^i, \mathbf{e}^*)$ and $d_j = m_d(\mathbf{e}^j, \mathbf{e}^*)$, and we define the total dissimilarity between the reproduced classes and the newly generated class as $m_d^{\text{total}} = |d_i| + |d_j|$. We repeat this process K times, this means that we get K different \mathbf{X}^* for the same \mathbf{c}^i and \mathbf{c}^j). We also want each $K \mathbf{X}^*$ to be as similar as possible to each other so we can assign the same label/class to them for a fixed α and β . To this end, we also calculate a similarity measure, m_s , in which the higher the absolute output of this measure is the the more similar their input is. We define the total similarity between the K generated \mathbf{X}^* as m_s^{total} . We hypothesize

and verify later with our experiments that the good candidates for α and β are the ones that have a high value of the $m^{\text{total}} = m_s^{\text{total}} + m_d^{\text{total}}$. This search for α and β is outlined in the Algorithm 1.

Algorithm 1 Grid search for α and β

```

Require: Search range for  $\alpha, \beta \in [0.1, 1.1], \mathbb{L}_s \subseteq \mathbb{L}, K$ : Number of iterations.
Require:  $G(., .)$ : Class-conditional Generator trained on  $D^{\text{orig}}$ 
Require:  $f_{\theta_{\text{dis}}^*}$ : Discriminator trained on  $D^{\text{orig}}$ 
Output:  $\alpha^*$  and  $\beta^*$ 

1: Create set  $\mathbb{W} = \{(\alpha, \beta) \mid \alpha, \beta \in [0.1, 1.1]\}$ 
2: Randomly select two values  $i$  and  $j$  from  $\mathbb{L}_s$ 
3: Create empty set  $\mathbb{M}$ .
4: for each  $(\alpha, \beta) \in \mathbb{W}$  do
5:    $\mathbf{c}^* = \alpha \mathbf{c}^i + \beta \mathbf{c}^j$ 
6:   Create empty set  $\mathbb{F}$ .
7:   for  $k = 1, \dots, K$  do
8:     Get Repro Images :  $\mathbf{X}^i = G(\mathbf{Z}, \mathbf{c}^i), \mathbf{X}^j = G(\mathbf{Z}, \mathbf{c}^j)$ 
9:     Get Interpolated Images :  $\mathbf{X}^* = G(\mathbf{Z}, \mathbf{c}^*)$ 
10:    Get Repro Features:  $\mathbf{e}^i, \mathbf{e}^j = f_{\theta_{\text{dis}}^*}(\mathbf{X}^i), f_{\theta_{\text{dis}}^*}(\mathbf{X}^j)$ 
11:    Get Interpolated Feature:  $\mathbf{e}^* = f_{\theta_{\text{dis}}^*}(\mathbf{X}^*)$ 
12:    Add  $\mathbf{e}^*$  to  $\mathbb{F}$ 
13:    Dissimilarity measures:  $d_i = m_d(\mathbf{e}^i, \mathbf{e}^*)$   

            $d_j = m_d(\mathbf{e}^j, \mathbf{e}^*)$ 
14:    Total dissimilarity:  $m_d^{\text{total}} = |d_i| + |d_j|$ 
15:  end for
16:   $m_s^{\text{total}} = 0$ 
17:   $\forall p, q \in \mathbb{F} | p \neq q$  Calculate  $m_s(\mathbf{e}^p, \mathbf{e}^q)$  and add it to  $m_s^{\text{total}}$ 
18:  Final measure:  $m^{\text{total}} = m_s^{\text{total}} + m_d^{\text{total}}$  and add it to  $\mathbb{M}$ .
19: end for
20: Return  $\alpha^*$  and  $\beta^*$  that the  $m^{\text{total}}$ , in  $\mathbb{M}$  is high.

```

After finding candidate values for α and β , by randomly selecting classes from \mathbb{L} , and calculating \mathbf{c}^* , we can generate images that represent a hypothetically new class. The output of this process is what we call generated augmentations of the D^{orig} , or D^{aug} as depicted in the Figure 2 (e) and presented in Algorithm 2. As shown in Figure 1, the newly generated classes are similar within themselves but distinct from their mixed classes, retaining source-class cues to aid discrimination by design. Training with the mix of D^{orig} and D^{aug} (Figure 2(f)) benefits the discriminator, as demonstrated in Section 4.

4. Experiments

We demonstrate the effectiveness of our proposed augmentation method for the problem of Face Recognition (FR). Large datasets are usually required for modern FR systems,

Algorithm 2 Generating D^{aug}

Require: α^* and β^* from Algorithm 1, $\mathbb{L}_s \subseteq \mathbb{L}$, C : Number of mixed classes, N : Number of samples per class.

Require: $G(\cdot, \cdot)$: Class-conditional Generator trained on D^{orig}

Output: D^{aug}

- 1: Create empty set D^{aug} .
- 2: **for** $n = 1, \dots, C$ **do**
- 3: Randomly select two values i and j from \mathbb{L}_s
- 4: $c^* = \alpha^* c^i + \beta^* c^j$
- 5: Create empty set T
- 6: **for** $n = 1, \dots, N$ **do**
- 7: $\mathbf{X}^* = G(\mathbf{Z}, c^*)$
- 8: Add \mathbf{X}^* to T .
- 9: **end for**
- 10: Add T to D^{aug} .
- 11: **end for**
- 12: Return D^{aug}

so improving performance with limited data is crucial.

4.1. Experimental Setup

Training Data. We evaluate our approach using two real-world datasets, D^{orig} : CASIA-WebFace (Yi et al., 2014) and a subset of WebFace4M (Zhu et al., 2021). The WebFace4M subset, referred in this work to as WebFace160K, was selected to include approximately 10,000 identities (*i.e.*, like CASIA-WebFace), each represented by 11 to 24 samples, resulting in a total of 160K face images. More details about the datasets are presented in the Appendix B.

Discriminative Model. To ensure a fair comparison across different methods during the training of the discriminator, we adopted a standardized baseline. This baseline employed a face recognition (FR) system consisting of a IR50 backbone, modified according to the ArcFace’s implementation (Deng et al., 2019), paired with the AdaFace head (Kim et al., 2022) to incorporate margin loss. Furthermore, when analyzing architectural improvements at the network level, we explored training solely with real data versus mixed data. For this analysis, we used IR101 due to its increased parameterization, which is expected to enhance its ability to generalize.

Each real or mixed dataset was trained multiple times with identical hyperparameters but different seed values. More details are outlined in Appendix C. For comparisons, we repeated these procedures using several synthetic datasets from the literature: the original DigiFace1M (3D graphics), its RealDigiFace translations (Rahimi et al., 2024) (Hybrid, 3D, and post-processed), and two diffusion-based datasets, DCFace (Kim et al., 2023) and IDiff-Face (Boutros et al., 2023). Additionally, standard augmentations for face recognition tasks were applied to all models. These augmenta-

tions included photometric transformations, cropping, and low-resolution adjustments to simulate common variations encountered in real-world scenarios.

Generative Model. To train our generative model, we used a variant of the diffusion formulation (Karras et al., 2022; 2024). For the D^{orig} CASIA-WebFace we used the latent-based formulation in which, as depicted in Equation 2 we employed a VAE to encode the image to a compressed space and decode it back to the image space. For WebFace160K we used the pixel space variant for better coverage of different diffusion models. Furthermore, we set the one-hot condition vectors $c^{\sim 10K}$, have a size of $\sim 10,000$, corresponding to the number of classes in D^{orig} . We train two versions of the latent diffusion model (LDM) from scratch, labeled small and medium, to analyze the impact of network size and training iterations on the final performance, following the size presets outlined in the original papers (Karras et al., 2024; 2022). For the pixel-space diffusion model, we mainly used the small variant. Details are presented in Appendix C.

Grid Search. As presented in the Algorithm 1 we need to find an appropriate α and β for generating useful augmentations based on the generator trained in the previous section. For the D^{orig} , CASIA-WebFace which has the long-tail distribution of samples per class, we set the \mathbb{L}_s to the classes from the generator that are presented more than the median number of samples per class. Naturally, we empirically observed that these classes are better reproduced when we were generating D^{repro} . For the case of WebFace160K the \mathbb{L}_s is all the classes. Later we set the \mathbb{W} to $\{0.1, 0.2, \dots, 1.0, 1.1\}$ for searching α and β to calculate the new condition vector c^* . Closely related to how the FR models are being trained, especially the usage of the margin loss (*i.e.*, AdaFace (Kim et al., 2022) or ArcFace (Deng et al., 2019)), we set the measure for dissimilarity between the features of the two sample images, \mathbf{X}^1 and \mathbf{X}^2 , using cosine similarity to $m_d = 1 - \frac{\mathbf{e}^1 \cdot \mathbf{e}^2}{\|\mathbf{e}^1\| \|\mathbf{e}^2\|}$. Note that the \mathbf{e} s were calculated using a discriminator that was trained solely on the D^{orig} . We treat the values of the measure in such a way that the higher the output of the measure the more it reflects its functionality (*i.e.*, the larger the measure for dissimilarity is the more dissimilar the inputs are). Accordingly, we set the similarity measure to $m_s = \frac{\mathbf{e}^1 \cdot \mathbf{e}^2}{\|\mathbf{e}^1\| \|\mathbf{e}^2\|}$, which again reflects that the inputs are more similar if the output of this measure is closer to 1. We iterate multiple choices of the i and j and average our m^{total} for each of the choices. A sample of the output of this process is depicted in 4a. Here we observe that by increasing the α and β from $(0.1, 0.1)$ to between $(0.7, 0.7)$ and $(0.8, 0.8)$ the measure increases and after that, it will decrease when we go toward $(1.1, 1.1)$, specifically, we are interested in the $\alpha = \beta$ line as we do not want to include any bias regarding the classes that we **randomly choose**. We consider three

sets of values for (α, β) , $(0.5, 0.5)$, $(0.7, 0.7)$ and $(1.0, 1.0)$ corresponding to the m^{total} of 1.48, **1.58** and 1.53 respectively. Then the (α^*, β^*) respectively from the Algorithm 1 for CASIA-WebFace is $(0.7, 0.7)$.

Based on our observations, for the WebFace160K dataset, we performed a coarser parameter search with a higher concentration in the range of 0.5 to 0.9. The total metric value, m^{total} , for WebFace160K is illustrated in 4b. Using this approach, we evaluated m^{total} for the parameter pairs (α, β) at specific points: $(0.5, 0.5)$, $(0.7, 0.7)$, $(0.8, 0.8)$, and $(1.0, 1.0)$. The corresponding m^{total} values were 0.6068, 0.7256, **0.7390**, and 0.7230, respectively. Based on these results, the (α^*, β^*) pair for WebFace160K was determined to be $(0.8, 0.8)$, as it achieved the highest m^{total} value of 0.7390. In Appendix G we quantitatively demonstrated the effectiveness of this measure in the final performance of the discriminator when we trained it on the synthetically generated dataset using various α and β .

Synthetic Dataset. For generating the reproduction dataset D^{repro} , we set the condition for each of the $\sim 10,000$ classes in the original CASIA-WebFace and WebFace160K dataset to the generator. The number of samples per class is 20 unless mentioned otherwise. For generating D^{aug} we randomly sampled $10,000 - 50,000$ combination of the \mathbb{L}_s , $\binom{\text{Card}(\mathbb{L}_s)}{2}$, (samples with more than the median number of sample/class in case of CASIA-WebFace as the D^{orig}), and fixed them for all the experiments. Later by setting the α and β to candidate values found in the previous section, (*i.e.*, like $(0.7, 0.7)$ for CASIA), we generated 10 to 50 sample per mixed of selected classes. In Figure 3, some samples of the generated images are shown where the first and last columns depict examples of the two classes in the D^{orig} . The second and 4-th columns are the reproduction of the same identities from the first and last column respectively, D^{repro} . Each line is generated using the same seed (source of randomness in the generator), and finally, the middle column (3rd from left) is the D^{aug} which is generated by $\mathbf{X}^* = G(\mathbf{Z}, \mathbf{c}^*)$ when we calculate the \mathbf{c}^* by optimum α and β . We can observe that the middle column’s identity is slightly different from the source classes while being coherent when we generate multiple examples of this new identity. **By design, these classes can be considered as hard examples for the discriminator.** This subtle difference is one of the reasons why our augmentation is improving the final performance. In the Appendix I more samples are presented.

4.2. Face Recognition Benchmarks

We show that our synthetic augmentation is boosting the performance of a model trained with the real dataset in all of the studied public FR benchmarks. For this purpose, we evaluated against two sets of FR benchmarks. The first set consists of LFW (Huang et al., 2008), CFPFP (Sengupta et al., 2016), CPLFW (Zheng & Deng, 2018), CALFW (Zheng et al., 2017), AgeDB (Moschoglou et al., 2017),



Figure 3. Randomly sampled images. From left to right: The first column shows variations of a randomly selected identity (ID 1) from D^{orig} . The second column presents the reproduction of the same ID using the generator, conditioned on the corresponding one-hot vector $G(\mathbf{Z}, \mathbf{c}_1)$. The third and fourth columns follow the same process for a different ID, with the middle column representing a newly synthesized identity generated by conditioning the generator on $G(\mathbf{Z}, \mathbf{c}^*)$. The samples above the red line are from CASIA-WebFace, while the lower part corresponds to WebFace160K.

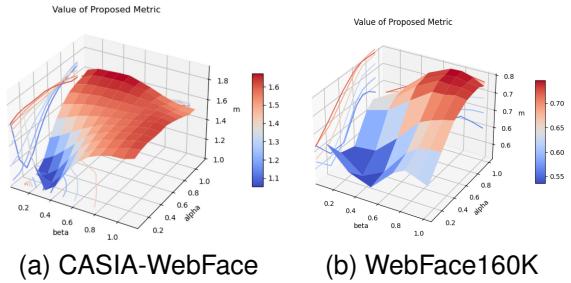


Figure 4. The value of the proposed measure m^{total} for setting the candidate values of α (x axis) and β (y axis). Here for each α and β and our 100 combination of \mathbb{L}_s we calculated the m^{total} by setting the K in Algorithm 1 to 10.

which includes mainly high-quality images with various lighting, poses, and ages the average of these benchmarks presented in Table 1 as **Avg-H**. The second set involves benchmarks consisting of medium to low-quality images from a realistic and more challenging FR scenario (NIST IJB-B/C) (Maze et al., 2018; Whitelam et al., 2017) and TinyFace (Cheng et al., 2019). For evaluation, we report verification accuracy (*i.e.*, True Acceptance Rate (TAR)), where the thresholds are set using cross-validation in the high-quality benchmarks, and TARs at different thresholds determined by fixed False Match Rates (FMR) in IJB-B/C. Specifically for the latter, we are mainly interested in the verification accuracy for two thresholds that are usually used

in real-world scenarios when the FR systems are being deployed, namely TAR@FPR=1e-06 and TAR@FPR=1e-05 for both IJB-B and IJB-C. In the Table 1 the **Aux** column depicts that if the method under study used any auxiliary model for the generation of the dataset other than the D^{orig} . The ideal value for this column is **N** which refers to not using any auxiliary model/datasets. The n^s and n^r depict the number of synthetic and real images used for training the discriminative model. The final values for the benchmarks are reported as the mean and std of the observed numbers when we are changing only the seed as discussed before. Details about the benchmarks, including High-Quality benchmarks and TAR at additional thresholds, are provided in Appendix D. Table 1 is divided into two sections, separated by a triple horizontal line. The upper section compares AugGen, using the CASIA-WebFace dataset as the source, and the lower part is when we set the D^{orig} to WebFace160K. For each, we considered fully synthetic face recognition, FR_{syn} , data, and a combination of synthetic and real data, (distinguished by a double horizontal line) FR_{mix} . This comparison evaluates their performance relative to the original source dataset (*i.e.*, fully real, FR_{real}) and relevant works, including synthetic data from three approaches: the proposed AugGen, DCFace (Kim et al., 2023), and IDiffFace (Boutros et al., 2023). The triple horizontal line segmentation is primarily due to the use of CASIA-WebFace and among other data/models in the latter two methods’ generation pipelines. For *each* part of the table, **bold** and underline numbers are presenting best and second best respectively. In the second part, in case augmentation with the real CASIA-WebFace is performing better than solely training with the CASIA-WebFace (*i.e.*, middle part of both tables) the cell is shaded in **gray**. We are observing inconsistencies in different benchmarks for other methods. For instance, for IJB-B/C DCFace is not performing better than CASIA-WebFace alone and IDiffface is not outperforming FR_{real} in thresholds set to low FPRs (*i.e.*, TAR@FPR=1e-6). In the case of FR_{real} training, we additionally used the **IR101** network depicted as \dagger . This is done to **demonstrate the introduced augmentation samples can be as important as architectural-level improvements**. As in most cases the less parametrized network (*i.e.*, IR50) trained with the AugGen samples is outperforming the more parametrized network, **IR101**, solely trained on the original samples, D^{orig} . This is in conjunction with the fact that in most cases using the **IR101** FR_{real} training outperforms the simpler IR50 model. Additionally, in case our augmentations also perform better than architectural improvements we shade the corresponding cell to **green**. For the less challenging benchmarks presented by Avg-H in Table 1, we observe that although our method consists of a smaller number of samples and does not use any auxiliary model/data we are performing competitively with other state-of-the-art (SOTA) methods/datasets. In the second part of this table we are observing mainly all the

methods that we combined with the CASIA-WebFace are boosting the discriminator which is solely trained on the CASIA-WebFace. For IJB/C we demonstrate better performance being the best in most FPRs although our datasets were generated for augmentation by design. By observing the results after the augmentation (second part of the table), AugGen is the only method that consistently performs better than the baseline. One interesting finding was the performance drop of the model when it was combined with the CASIA-WebFace. But we are observing that *consistently in all of the benchmarks, our augmentation methodology is boosting the baseline*. We demonstrate that although we did not use any auxiliary model/data our synthetic dataset performed competitively with other state-of-the-art methods or even outperformed them in some cases.

The lower part of the triple horizontal line reports results with AugGen samples using our WebFace160K as the D^{orig} . The observations remain the same, as in all cases we are performing even better than architectural improvements.

As shown in Figure 1, the discriminator’s feature space exhibits reduced intra-class variation and increased inter-class separation, with further details in Appendix H.

4.3. Gains over Additional Real Data

In this section, we aim to address a critical question: *How much additional **real** (non-generated) data would it take to achieve the same performance improvement as our synthetic augmentation?* This experiment is vital because the primary goal is to maximize the accuracy of the face recognition (FR) system using the existing dataset. To evaluate this, we used our *WebFace160K* subset as a baseline and incrementally added data from the *WebFace4M* dataset. This process allows us to determine how the performance boost achieved through *AugGen* compares to the addition of real data, providing a clear measure of its effectiveness. In Table 2, the **Ratio** represents the proportion of additional real samples added to *WebFace160K* (*e.g.*, 160K + 110K with a Ratio of 1.69). Remarkably, adding approximately 600K AugGen samples delivers performance gains comparable to including 110K real images. This highlights that AugGen achieves equivalent performance improvements with significantly fewer **real** images.

5. Conclusions

In this work, we introduced a novel yet simple sampling approach that carefully conditions a generator using a discriminative model, both trained on a single real dataset, to generate augmented samples. By combining these augmented samples with the original real dataset for training, we enhance the performance of discriminative models without relying on auxiliary data or pre-trained networks. Our proposed AugGen method significantly im-

Table 1. Comparison of the FR_{syn} training (upper part), FR_{real} training (middle), and FR_{mix} training (bottom) using CASIA-WebFace/WebFace160K, when the models are evaluated in terms of accuracy against standard FR benchmarks. **Avg-H** depicts the average accuracy of all high-quality benchmarks including, LFW, CFP-FP, CPLFW, AgeDB, and CALFW. Here n^s and n^r depict the number of Synthetic and Real Images respectively and Aux depicts whether the method for generating the dataset uses an auxiliary information network for generating their datasets (**Y**) or not (**N**). the \dagger denotes network trained on IR101 if not the model trained using the IR50. The numbers under columns labeled like C/B-1e-6 indicate TAR for IJB-C/B at FPR of 1e-6. TR1 depicts the rank-1 accuracy for the TinyFace benchmark.

Method/Data	Aux	n^s	n^r	B-1e-6	B-1e-5	C-1e-6	C-1e-5	TR1	Avg-H
DigiFace1M	N/A	1.22M	0	15.31 \pm 0.42	29.59 \pm 0.82	26.06 \pm 0.77	36.34 \pm 0.89	32.30 \pm 0.21	78.97 \pm 0.44
RealDigiFace	Y	1.20M	0	21.37 \pm 0.59	39.14 \pm 0.40	36.18 \pm 0.19	45.55 \pm 0.55	42.64 \pm 1.70	81.34 \pm 0.02
IDiff-face	Y	1.2M	0	26.84\pm2.03	50.08\pm0.48	41.75\pm1.04	51.93\pm0.89	45.98\pm0.61	84.68 \pm 0.05
DCFace	Y	1.2M	0	22.48 \pm 4.35	47.84 \pm 6.10	35.27 \pm 10.78	58.22 \pm 7.50	45.94 \pm 0.01	91.56\pm0.09
D ^{aug} (Ours)	N	0.6M	0	29.40\pm1.36	54.54\pm0.59	45.15\pm1.04	61.52\pm0.47	52.33\pm0.03	88.78 \pm 0.06
D ^{repro} (Ours)	N	0.6M	0	15.71 \pm 3.12	45.97 \pm 4.64	31.54 \pm 6.65	58.61 \pm 3.89	53.61\pm0.47	90.64 \pm 0.07
CASIA-WebFace	N/A	0	0.5M	1.02 \pm 0.26	5.06 \pm 1.70	0.73 \pm 0.19	5.37 \pm 1.41	58.12 \pm 0.31	94.21 \pm 0.09
CASIA-WebFace \dagger	N/A	0	0.5M	0.74 \pm 0.31	3.94 \pm 1.62	0.38 \pm 0.13	3.92 \pm 1.96	59.64 \pm 0.49	94.84 \pm 0.07
IDiff-face	Y	1.2M	0.5M	0.89 \pm 0.07	5.80\pm0.63	0.70 \pm 0.11	7.46 \pm 2.08	59.32 \pm 0.34	94.86\pm0.02
DCFace	Y	0.5M	0.5M	0.26 \pm 0.11	1.59 \pm 0.51	0.18 \pm 0.07	1.54 \pm 0.59	56.60 \pm 0.41	94.72 \pm 0.09
D ^{aug} (Ours)	N	0.6M	0.5M	2.61\pm0.91	15.74\pm3.20	4.36\pm1.41	18.58\pm3.99	59.82\pm0.13	94.66 \pm 0.03
WebFace160K	N/A	0	0.16M	32.13 \pm 1.87	72.18 \pm 0.18	70.37 \pm 0.75	78.81 \pm 0.32	61.51 \pm 0.16	92.50 \pm 0.02
WebFace160K \dagger	N/A	0	0.16M	34.84 \pm 0.49	74.10 \pm 0.24	72.56 \pm 0.02	81.26 \pm 0.14	62.59 \pm 0.01	93.32 \pm 0.12
D ^{aug} (Ours)	N	0.6M	0.16M	36.62\pm0.77	78.32\pm0.33	78.58\pm0.15	85.02\pm0.15	61.60\pm0.38	94.17 \pm 0.08

Table 2. Effect of adding more real samples from WebFace4M to WebFace160K in comparison to adding more synthetic images. The backbone for all models is IR50. Here **Avg-H** depicts the average accuracy of all high-quality benchmarks including, LFW, CFP-FP, CPLFW, AgeDB, and CALFW. **Ratio** depicts the ratio number of **real** samples used over the number of samples in WebFace160K. The numbers under columns labeled like C/B-1e-6 indicate TAR for IJB-C/B at FPR of 1e-6.

Syn #Class \times #Sample	n^r	n^s	B-1e-6	B-1e-5	C-1e-6	C-1e-5	Avg-H	Ratio
0	160K	0	32.13 \pm 1.87	72.18 \pm 0.18	70.37 \pm 0.75	78.81 \pm 0.32	92.50 \pm 0.02	1
(10K x 20)	160K	200K	34.93 \pm 0.50	76.15 \pm 0.20	75.18 \pm 0.22	83.06 \pm 0.11	93.77 \pm 0.04	1
(20K x 20)	160K	400K	36.54 \pm 1.27	78.00 \pm 0.23	78.48 \pm 0.55	84.40 \pm 0.07	93.96 \pm 0.01	1
(25K x 20)	160K	500K	36.35\pm0.70	77.87\pm0.52	78.61\pm0.42	84.49 \pm 0.01	94.10 \pm 0.08	1
(30K x 20)	160K	600K	36.62\pm0.77	78.32\pm0.33	78.58\pm0.15	85.02\pm0.15	94.17\pm0.08	1
0	160K + 80K	0	33.78 \pm 1.11	77.29 \pm 0.12	77.38 \pm 0.10	83.50 \pm 0.04	93.85 \pm 0.02	1.5
0	160K + 110K	0	33.53 \pm 1.47	78.26\pm0.05	78.49 \pm 0.54	85.02\pm0.01	94.19\pm0.01	1.69
0	800K	0	38.12 \pm 0.00	87.68 \pm 0.00	87.11 \pm 0.00	92.27 \pm 0.00	96.46 \pm 0.00	5.0

proves discriminative model performance across multiple FR benchmarks, consistently outperforming baseline models and, in many cases, exceeding architectural-level enhancements—highlighting its potential to compete with architectural-level improvements. We further demonstrate that training with AugGen-augmented datasets is as effective as using 1.7 \times more real samples, emphasizing its impact on alleviating data collection challenges. Additionally, we identify inconsistencies in CASIA-WebFace-based evaluations and recommend alternative datasets for more reliable benchmarking on IJB-B/C. Our findings underscore the potential of augmentation-based approaches for improving discriminative models.

Limitations. The primary limitation of our method is its computational intensity. We anticipate that improved generator architectures will further enhance performance, as demonstrated by our comparison between medium-sized and small-sized generators.

Future work. A promising research direction is reformulating margin losses in FR to be compatible with soft labels. By establishing a correlation between target soft labels and c^* (e.g., with $\alpha, \beta = 0.7$ increasing m^{total} , a natural choice for soft target labels would be 0.5, 0.5 for corresponding source classes), future studies can explore whether treating a class as a soft-class or a new one yields better performance.

References

- Anderson, B. D. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.
- Azizi, S., Kornblith, S., Saharia, C., Norouzi, M., and Fleet, D. J. Synthetic data from diffusion models improves imagenet classification. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL <https://openreview.net/forum?id=DlRs0xjyPm>.
- Bae, G., de La Gorce, M., Baltrušaitis, T., Hewitt, C., Chen, D., Valentin, J., Cipolla, R., and Shen, J. Digiface-1m: 1 million digital face images for face recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3526–3535, 2023.
- Blanz, V. and Vetter, T. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’99*, pp. 187–194, USA, 1999. ACM Press/Addison-Wesley Publishing Co. ISBN 0201485605. doi: 10.1145/311535.311556. URL <https://doi.org/10.1145/311535.311556>.
- Boutros, F., Grebe, J. H., Kuijper, A., and Damer, N. Idiff-face: Synthetic-based face recognition through fuzzy identity-conditioned diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19650–19661, 2023.
- Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramer, F., Balle, B., Ippolito, D., and Wallace, E. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 5253–5270, 2023.
- Cheng, Z., Zhu, X., and Gong, S. Low-resolution face recognition. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*, pp. 605–621. Springer, 2019.
- DeAndres-Tame, I., Tolosana, R., Melzi, P., Vera-Rodriguez, R., Kim, M., Rathgeb, C., Liu, X., Morales, A., Fierrez, J., Ortega-Garcia, J., et al. Frcsyn challenge at cvpr 2024: Face recognition challenge in the era of synthetic data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3173–3183, 2024.
- Deng, J., Guo, J., Xue, N., and Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4690–4699, 2019.
- Deng, Y., Yang, J., Chen, D., Wen, F., and Tong, X. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5154–5163, 2020.
- Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.
- Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. Synthetic data augmentation using gan for improved liver lesion classification. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pp. 289–293, 2018. doi: 10.1109/ISBI.2018.8363576.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.
- Gu, J., Zhai, S., Zhang, Y., Susskind, J. M., and Jaitly, N. Matryoshka diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=tOzCcDdH9O>.
- Guo, Y., Zhang, L., Hu, Y., He, X., and Gao, J. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part III 14*, pp. 87–102. Springer, 2016.
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.
- Ho, J. and Salimans, T. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL <https://openreview.net/forum?id=qw8AKxfYbI>.
- Hoogeboom, E., Heek, J., and Salimans, T. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pp. 13213–13232. PMLR, 2023.
- Huang, G. B., Mattar, M., Berg, T., and Learned-Miller, E. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Workshop on faces in ‘Real-Life’ Images: detection, alignment, and recognition*, 2008.
- Karras, T., Laine, S., and Aila, T. 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, 2019.

- Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8110–8119, 2020.
- Karras, T., Aittala, M., Laine, S., Hrknen, E., Hellsten, J., Lehtinen, J., and Aila, T. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021.
- Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35: 26565–26577, 2022.
- Karras, T., Aittala, M., Lehtinen, J., Hellsten, J., Aila, T., and Laine, S. Analyzing and improving the training dynamics of diffusion models. In *Proc. CVPR*, 2024.
- Kim, M., Jain, A. K., and Liu, X. Adaface: Quality adaptive margin for face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18750–18759, 2022.
- Kim, M., Liu, F., Jain, A., and Liu, X. Dcface: Synthetic face generation with dual condition diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12715–12725, 2023.
- Kingma, D. P. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- Kynkänniemi, T., Karras, T., Laine, S., Lehtinen, J., and Aila, T. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.
- Li, Z., Hong, J., Li, B., and Wang, Z. Shake to leak: Fine-tuning diffusion models can amplify the generative privacy risk. In *2024 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 18–32. IEEE, 2024.
- Maze, B., Adams, J., Duncan, J. A., Kalka, N., Miller, T., Otto, C., Jain, A. K., Niggel, W. T., Anderson, J., Cheney, J., et al. Iarpa janus benchmark-c: Face dataset and protocol. In *2018 international conference on biometrics (ICB)*, pp. 158–165. IEEE, 2018.
- Melzi, P., Rathgeb, C., Tolosana, R., Vera-Rodriguez, R., Lawatsch, D., Domin, F., and Schaubert, M. Gandiff-face: Controllable generation of synthetic datasets for face recognition with realistic variations. *arXiv preprint arXiv:2305.19962*, 2023.
- Moschoglou, S., Papaioannou, A., Sagonas, C., Deng, J., Kotsia, I., and Zafeiriou, S. Agedb: the first manually collected, in-the-wild age database. In *proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 51–59, 2017.
- Naeem, M. F., Oh, S. J., Uh, Y., Choi, Y., and Yoo, J. Reliable fidelity and diversity metrics for generative models. In *International Conference on Machine Learning*, pp. 7176–7185. PMLR, 2020.
- Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.
- Oquab, M., Dariseti, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.
- Qiu, H., Yu, B., Gong, D., Li, Z., Liu, W., and Tao, D. Synface: Face recognition with synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10880–10890, 2021.
- Rahimi, P., Razeghi, B., and Marcel, S. Synthetic to authentic: Transferring realism to 3d face renderings for boosting face recognition. *arXiv preprint arXiv:2407.07627*, 2024.
- Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.
- Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., and Aberman, K. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22500–22510, 2023.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.
- Sajjadi, M. S., Bachem, O., Lucic, M., Bousquet, O., and Gelly, S. Assessing generative models via precision and recall. *Advances in neural information processing systems*, 31, 2018.
- Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35: 25278–25294, 2022.

- Sengupta, S., Chen, J.-C., Castillo, C., Patel, V. M., Chellappa, R., and Jacobs, D. W. Frontal to profile face verification in the wild. In *2016 IEEE winter conference on applications of computer vision (WACV)*, pp. 1–9. IEEE, 2016.
- Sevastopolskiy, A., Malkov, Y., Durasov, N., Verdoliva, L., and Nießner, M. How to boost face recognition with stylegan? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20924–20934, 2023.
- Shahreza, H. O. and Marcel, S. Unveiling synthetic faces: How synthetic datasets can expose real identities. *arXiv preprint arXiv:2410.24015*, 2024.
- Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- Stein, G., Cresswell, J. C., Hosseinzadeh, R., Sui, Y., Ross, B. L., Villecroze, V., Liu, Z., Caterini, A. L., Taylor, E., and Loaiza-Ganem, G. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=08zf7kTOoh>.
- Sun, Z., Song, S., Patras, I., and Tzimiropoulos, G. Cemi-face: Center-based semi-hard synthetic face generation for face recognition. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.
- Theiss, J., Leverett, J., Kim, D., and Prakash, A. Unpaired image translation via vector symbolic architectures. In *European Conference on Computer Vision*, pp. 17–32. Springer, 2022.
- Trabucco, B., Doherty, K., Gurinas, M. A., and Salakhutdinov, R. Effective data augmentation with diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=ZWzUA9zeAg>.
- Whitelam, C., Taborsky, E., Blanton, A., Maze, B., Adams, J., Miller, T., Kalka, N., Jain, A. K., Duncan, J. A., Allen, K., et al. Iarpa janus benchmark-b face dataset. In *proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 90–98, 2017.
- Wood, E., Baltrušaitis, T., Hewitt, C., Dziadzio, S., Cashman, T. J., and Shotton, J. Fake it till you make it: face analysis in the wild using synthetic data alone. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3681–3691, 2021.
- Xu, J., Li, S., Wu, J., Xiong, M., Deng, A., Ji, J., Huang, Y., Mu, G., Feng, W., Ding, S., et al. Id³: Identity-preserving-yet-diversified diffusion models for synthetic face recognition. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- Yi, D., Lei, Z., Liao, S., and Li, S. Z. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.
- Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.
- Zhang, H. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.
- Zheng, T. and Deng, W. Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. *Beijing University of Posts and Telecommunications, Tech. Rep*, 5(7), 2018.
- Zheng, T., Deng, W., and Hu, J. Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments. *arXiv preprint arXiv:1708.08197*, 2017.
- Zhou, S., Chan, K. C., Li, C., and Loy, C. C. Towards robust blind face restoration with codebook lookup transformer. In *NeurIPS*, 2022.
- Zhu, Z., Huang, G., Deng, J., Ye, Y., Huang, J., Chen, X., Zhu, J., Yang, T., Lu, J., Du, D., et al. Webface260m: A benchmark unveiling the power of million-scale deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10492–10502, 2021.

Method	Year	Generation Methodology	Aux	FR
SynFace (Qiu et al., 2021)	2021	3DMM & GAN	Y	Y
DigiFace1M (Bae et al., 2023)	2023	3D-Rendering	Y	N
DCFace (Kim et al., 2023)	2023	Diffusion	Y	Y
IDiffFace (Boutros et al., 2023)	2023	Diffusion	N	Y
GANDiffFace (Melzi et al., 2023)	2023	GAN/Diffusion	Y	Y
RealDigiFace (Rahimi et al., 2024)	2024	GAN/Diffusion	Y	N
ID ³ (Xu et al., 2024)	2024	Diffusion	Y	Y
CemiFace (Sun et al., 2024)	2024	Diffusion	N	Y
Ours	-	Diffusion	N	N

Table 3. State-of-the-art Synthetic Face Recognition (SFR) dataset generation methods are compared based on two criteria: the use of Auxiliary Networks (Aux) and External Face Recognition (FR) Systems. Aux indicates whether auxiliary networks are utilized, with Y representing "Yes" and N representing "No." Similarly, FR highlights the use of external face recognition systems beyond those trained solely on the methodology's dataset, using the same Y/N notation.

A. Summary of SOTA methods

Table 3 summarized recent methodologies for synthetic FR dataset generation. Here the *Generation Methodology* refers to which of the main methods (*i.e.*, Diffusion, GAN, 3DMM, ...) were used to generate synthetic data. *Auxiliary Networks (Aux)* refers to the use of additional models (e.g., age estimators, face parsers) or datasets during synthetic data generation. The last column, *FR*, indicates whether a strong pre-trained FR backbone, separate from the dataset used for training, was employed or not. As depicted in Figure 5, unlike most methods that depend on auxiliary datasets or models for improvement, our methods only rely on a single source of real data.

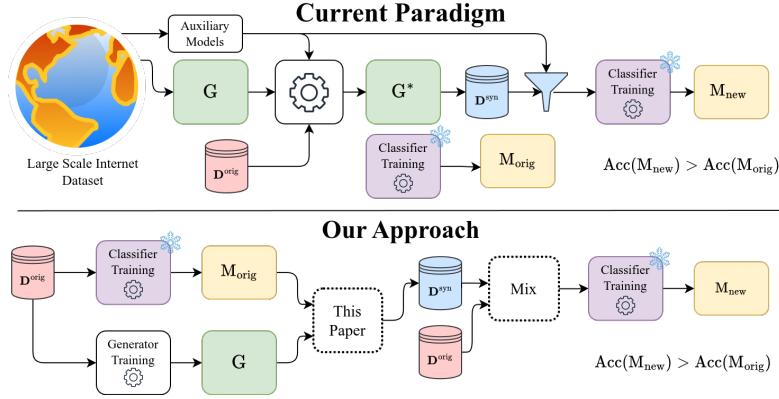


Figure 5. In this paper we explore the use of a generator and discriminator trained on the same dataset to generate useful augmentation of the data that can make the final downstream model more robust (*i.e.*, better performance across diverse benchmarks). This is opposed to the current paradigm which for doing so they use separate auxiliary models and generators trained on large-scale datasets.

B. Original Datasets D^{orig}

Table 4 summarizes the key statistics of CASIA-WebFace, WebFace160K, and the original WebFace4M dataset. Notably, WebFace160K was curated to avoid a long-tail distribution in the number of samples per identity, aligning its statistics more closely to equal presentation while differing from the CASIA-WebFace.

C. Experiment Details

C.1. Discriminator Training

In the Table 5 the most important parameters for training our discriminative models are presented.

AugGen: Synthetic Augmentation Can Improve Discriminative Models

Name	n IDs	n^r	Min	25%	50%	75%	Max
CASIA-WebFace	~10.5K	~490K	2	18	27	48	802
WebFace160K	~10K	~160K	11	13	16	19	24
WebFace4M	~206K	~4,235K	1	6	11	24	1497

Table 4. The middle part of the table presents the datasets used in this paper as D^{orig} , n IDs and n^r representing the number of IDs and real images. The Min and Max present the minimum and maximum number of samples per identity for the corresponding dataset. The number of samples like 25%, 50%, and 75% percentiles are also provided.

C.2. Generator and Its Training

We trained two sizes of generator namely small and medium as in (Karras et al., 2024). The training of the small-sized generator took about 1 NVIDIA H100 GPU day for the generator to see 805M images in different noise levels with a batch size of 2048. For reaching the same number of training images for the medium-sized generator, took about 2 days with a batch size of 1024. We used an Exponential Moving Average (EMA) length of 10%. As observed in literature (Nichol & Dhariwal, 2021), the EMA of model weights plays a crucial role in the output quality of the Image Generators.

For sampling our models we did **not** employ any Classifier Free Guidance (CFG) (Ho & Salimans, 2021).

C.3. Table Details

For the Table 10 we conditioned a medium-sized generator which trained till it saw 805M images in different noise levels (~1500 Epochs). The conditions were set according to the four sets of values of the α and β . This is done for a fixed identity combination from the \mathbb{L}_s for all of them. Later for each of these new conditions c^* we generated 50 images. All other tables were reported from a medium-sized generator when they saw 335M training samples.

Table 5. Details of the Discriminator and its Training

Parameter Name	Discriminator Type 1	Discriminator Type 2
Network type	ResNet 50	ResNet 50
Marin Loss	AdaFace	AdaFace
Batch Size	192	512
GPU Number	4	1
Gradient Acc Step	1 (For every training step)	N/A
GPU Type	Nvidia RTX 3090 Ti	Nvidia H100
Precision of Floating Point Operations	High	High
Matrix Multiplication Precision	High	High
Optimizer Type	SGD	SGD
Momentum	0.9	0.9
Weight Decay	0.0005	0.0005
Learning Rate	0.1	0.1
WarmUp Epoch	1	1
Number of Epochs	26	26
LR Scheduler	Step	Step
LR Milestones	[12, 24, 26]	[12, 24, 26]
LR Lambda	0.1	0.1
Input Dimension	112×112	112×112
Input Type	RGB images	RGB Images
Output Dimension	512	512
Seed	41,2048,10 (In some models)	41,2048

D. FR Benchmark Details

The full tables are presented in this section. Detailed results for the High-Quality benchmarks are presented in Table 6. Results for more thresholds set by various FPRs for IJB-B/C are presented in Table 7 and Table 7 respectively.

Table 6. Comparison of the FR_{syn} training (upper part), FR_{real} training (middle), and FR_{mix} training (bottom) using CASIA-WebFace and our WebFace160K, when the models are evaluated in terms of accuracy against standard FR benchmarks, namely LFW, CFPFP, CPLFW, AgeDB and CALFW with their corresponding protocols. Here n^s and n^r depict the number of Synthetic and Real Images respectively and Aux depicts whether the method for generating the dataset uses an auxiliary information network for generating their datasets (**Y**) or not (**N**). the \dagger denotes network trained on IR101 if not the model trained using the IR50.

Method/Data	Aux	n^s	n^r	LFW	CFP-FP	CPLFW	AgeDB	CALFW	Avg
DigiFace1M	N/A	1.22M	0	92.43 \pm 0.00	74.64 \pm 0.06	82.57 \pm 0.43	75.72 \pm 0.51	69.48 \pm 1.32	78.97 \pm 0.44
RealDigiFace	Y	1.20M	0	93.88 \pm 0.19	76.95 \pm 0.17	85.47 \pm 0.06	77.57 \pm 0.07	72.82 \pm 0.59	81.34 \pm 0.02
IDiff-face	Y	1.2M	0	97.45 \pm 0.05	77.07 \pm 0.34	80.48 \pm 0.63	87.26 \pm 0.05	81.15 \pm 0.61	84.68 \pm 0.05
DCFace	Y	1.2M	0	98.77\pm0.12	84.13\pm0.35	91.19\pm0.01	92.52\pm0.07	91.21\pm0.06	91.56\pm0.09
D ^{aug} (Ours)	N	0.6M	0	98.38 \pm 0.12	83.35 \pm 0.12	87.64 \pm 0.06	89.64 \pm 0.29	84.88 \pm 0.53	88.78 \pm 0.06
D ^{repro} (Ours)	N	0.6M	0	98.60 \pm 0.02	85.26\pm0.14	91.13 \pm 0.14	90.54 \pm 0.16	87.69 \pm 0.19	90.64 \pm 0.07
CASIA-WebFace	N/A	0	0.5M	99.32 \pm 0.02	88.97 \pm 0.27	96.35 \pm 0.06	93.07 \pm 0.13	93.34 \pm 0.14	94.21 \pm 0.09
CASIA-WebFace \dagger	N/A	0	0.5M	99.45 \pm 0.05	89.92 \pm 0.12	97.06 \pm 0.06	93.54 \pm 0.02	94.33 \pm 0.13	94.86 \pm 0.07
IDiff-face	Y	1.2M	0.5M	99.53\pm0.07	89.92 \pm 0.01	96.91\pm0.27	93.64\pm0.16	94.28\pm0.04	94.86\pm0.02
DCFace	Y	0.5M	0.5M	99.43 \pm 0.08	89.44 \pm 0.42	96.67 \pm 0.16	93.82\pm0.04	94.24\pm0.15	94.72\pm0.09
D ^{aug} (Ours)	N	0.5M	0.5M	99.47 \pm 0.07	89.96\pm0.07	96.71 \pm 0.05	93.40 \pm 0.22	93.74 \pm 0.02	94.66 \pm 0.03
WebFace160K	N/A	0	0.16M	99.08 \pm 0.13	87.99 \pm 0.45	93.95 \pm 0.59	92.75 \pm 0.20	90.78 \pm 0.79	92.91 \pm 0.42
WebFace160K \dagger	N/A	0	0.16M	98.97 \pm 0.11	87.54 \pm 0.06	93.40 \pm 0.01	92.55 \pm 0.02	90.01 \pm 0.04	92.50 \pm 0.02
D ^{aug} (Ours)	N	0.6M	0.16M	99.39 \pm 0.03	89.56 \pm 0.08	95.84 \pm 0.29	93.60 \pm 0.10	92.47 \pm 0.17	94.17 \pm 0.08

Table 7. Comparison of the FR_{syn} training, FR_{real} training, and FR_{mix} training, when the models are evaluated against IJB-B with thresholds set by various FPRs in terms of TAR. Here n^s and n^r depict the number of Synthetic and Real Images respectively and Aux depicts whether the method for generating the dataset uses an auxiliary information network for generating their datasets (**Y**) or not (**N**). the \dagger denotes network trained on IR101 if not the model trained with the IR50. The numbers under columns labeled like $B\text{-}1e\text{-}6$ indicate TAR for IJB-B at FPR of 1e-6.

Method/Data	Aux	n^s	n^r	B-1e-6	B-1e-5	B-1e-4	B-1e-3	B-0.01	B-0.1	Avg
DigiFace1M	N/A	1.22M	0	15.31 \pm 0.42	29.59 \pm 0.82	43.53 \pm 0.77	59.89 \pm 0.51	76.62 \pm 0.44	91.01 \pm 0.12	52.66 \pm 0.47
RealDigiFace	Y	1.20M	0	21.37 \pm 0.59	39.14 \pm 0.40	52.61 \pm 0.70	67.68 \pm 0.73	81.30 \pm 0.56	93.15 \pm 0.17	59.21 \pm 0.52
IDiff-face	Y	1.2M	0	26.84\pm2.03	50.08\pm0.48	64.58 \pm 0.32	77.19 \pm 0.41	88.27 \pm 0.15	95.94 \pm 0.05	67.15 \pm 0.50
DCFace	Y	1.2M	0	22.48 \pm 4.35	47.84 \pm 6.10	73.20\pm2.53	86.11\pm0.59	93.55\pm0.16	97.56\pm0.06	70.12\pm2.28
D ^{aug} (Ours)	N	0.6M	0	29.40\pm1.36	54.54\pm0.59	70.93 \pm 0.25	82.95 \pm 0.08	91.67 \pm 0.10	97.05 \pm 0.04	71.09\pm0.11
D ^{repro} (Ours)	N	0.6M	0	15.71 \pm 3.12	45.97 \pm 4.64	73.05 \pm 0.89	85.54 \pm 0.16	93.52 \pm 0.17	97.82\pm0.08	68.60 \pm 1.43
CASIA-WebFace	N/A	0	0.5M	1.02 \pm 0.26	5.06 \pm 1.70	50.37 \pm 4.03	87.13 \pm 0.38	95.36 \pm 0.11	98.36 \pm 0.04	56.22 \pm 0.99
CASIA-WebFace \dagger	N/A	0	0.5M	0.74 \pm 0.31	3.94 \pm 1.62	49.30 \pm 5.75	88.42 \pm 0.69	95.78 \pm 0.16	98.44 \pm 0.09	56.10 \pm 1.42
IDiff-face	Y	1.2M	0.5M	0.89\pm0.07	5.80\pm0.63	54.76\pm2.31	88.33\pm0.49	96.02\pm0.04	98.59\pm0.03	57.40\pm0.56
DCFace	Y	0.5M	0.5M	0.26 \pm 0.11	1.59 \pm 0.51	35.62 \pm 7.89	84.30 \pm 3.52	95.10 \pm 0.46	98.36 \pm 0.08	52.54 \pm 2.08
D ^{aug} (Ours)	N	0.5M	0.5M	2.61\pm0.91	15.74\pm3.20	63.67\pm1.68	89.19\pm0.28	95.78\pm0.02	98.51\pm0.05	60.92\pm1.02
WebFace160K	N/A	0	0.16M	32.13 \pm 1.87	72.18 \pm 0.18	82.96 \pm 0.20	90.37 \pm 0.04	95.66 \pm 0.11	98.75 \pm 0.00	78.67 \pm 0.40
WebFace160K \dagger	N/A	0	0.16M	34.84 \pm 0.49	74.10 \pm 0.24	84.57 \pm 0.41	91.57 \pm 0.12	96.09 \pm 0.12	98.87 \pm 0.03	80.01 \pm 0.24
D ^{aug} (Ours)	N	0.6M	0.16M	36.62 \pm 0.77	78.32 \pm 0.33	87.65 \pm 0.11	93.34 \pm 0.13	96.86 \pm 0.12	99.01 \pm 0.05	81.97 \pm 0.16

E. Mixing Effect

In Table 9, by setting the original dataset to CASIA-WebFace, the effect of increasing the number of samples in our augmented dataset using $(\alpha, \beta) = (0.7, 0.7)$ weights is shown. On average, adding more classes (#Class) and samples per class (#Sample) improves the performance of the final discriminative model. The performance eventually decreases as more samples are added per class. We hypothesize that this is due to the similarity of images generated under the new conditions, c , when sampling $G(\mathbf{Z}, c)$ multiple times. This reduces the intra-class variability necessary for training an effective discriminator. We also observe that we should add an appropriate number of the augmentation dataset (*i.e.*, comparing $10k \times 5$ to without any augmentation) for the final performance to be better than the discriminator trained on the

Table 8. Comparison of the FR_{syn} training, FR_{real} training, and FR_{mix} training, when the models are evaluated against IJB-C with thresholds set by various FPRs in terms of TAR. Here n^s and n^r depict the number of Synthetic and Real Images respectively and Aux depicts whether the method for generating the dataset uses an auxiliary information network for generating their datasets (Y) or not (N). the \dagger denotes network trained on IR101 if not the model trained with the IR50. The numbers under columns labeled like $B\text{-}1e\text{-}6$ indicate TAR for IJB-C at FPR of 1e-6.

Method/Data	Aux	n^s	n^r	C-1e-6	C-1e-5	C-1e-4	C-1e-3	C-0.01	C-0.1	Avg
DigiFace1M	N/A	1.22M	0	26.06 \pm 0.77	36.34 \pm 0.89	49.98 \pm 0.55	65.17 \pm 0.39	80.21 \pm 0.22	92.44 \pm 0.05	58.37 \pm 0.46
RealDigiFace	Y	1.20M	0	36.18 \pm 0.19	45.55 \pm 0.55	58.63 \pm 0.59	72.06 \pm 0.90	84.77 \pm 0.59	94.57 \pm 0.19	65.29 \pm 0.50
IDiff-face	Y	1.2M	0	41.75 \pm 1.04	51.93 \pm 0.89	65.01 \pm 0.63	78.25 \pm 0.39	89.41 \pm 0.19	96.55 \pm 0.05	70.48 \pm 0.47
DCFace	Y	1.2M	0	35.27 \pm 1.08	58.22 \pm 7.50	77.51 \pm 2.89	88.86 \pm 0.69	94.81 \pm 0.09	98.06 \pm 0.06	75.46 \pm 3.65
D ^{aug} (Ours)	N	0.6M	0	45.15 \pm 1.04	61.52 \pm 0.47	74.12 \pm 0.33	85.09 \pm 0.20	93.01 \pm 0.17	97.64 \pm 0.04	76.09 \pm 0.38
D ^{repro} (Ours)	N	0.6M	0	31.54 \pm 6.65	58.61 \pm 3.89	78.11 \pm 0.51	88.51 \pm 0.04	94.79 \pm 0.09	98.17 \pm 0.04	74.96 \pm 1.82
CASIA-WebFace	N/A	0	0.5M	0.73 \pm 0.19	5.37 \pm 1.41	56.76 \pm 2.73	89.44 \pm 0.35	96.16 \pm 0.07	98.61 \pm 0.02	57.84 \pm 0.75
CASIA-WebFace \dagger	N/A	0	0.5M	0.38 \pm 0.13	3.92 \pm 1.96	55.21 \pm 6.21	90.42 \pm 0.76	96.55 \pm 0.19	98.69 \pm 0.10	57.53 \pm 1.54
IDiff-face	Y	1.2M	0.5M	0.70 \pm 0.11	7.46 \pm 2.08	57.43 \pm 4.17	89.89 \pm 0.71	96.63 \pm 0.08	98.77 \pm 0.01	58.48 \pm 1.19
DCFace	Y	0.5M	0.5M	0.18 \pm 0.07	1.54 \pm 0.59	38.17 \pm 8.24	86.18 \pm 3.32	95.88 \pm 0.42	98.59 \pm 0.05	53.42 \pm 2.11
D ^{aug} (Ours)	N	0.5M	0.5M	4.36 \pm 1.41	18.58 \pm 3.99	67.85 \pm 2.18	91.12 \pm 0.38	96.57 \pm 0.07	98.78 \pm 0.05	62.88 \pm 1.35
WebFace160K	N/A	0	\sim 0.16M	70.37 \pm 0.75	78.81 \pm 0.32	86.45 \pm 0.11	92.68 \pm 0.01	96.52 \pm 0.05	99.02 \pm 0.01	87.31 \pm 0.20
WebFace160K \dagger	N/A	0	\sim 0.16M	72.56 \pm 0.02	81.26 \pm 0.14	88.27 \pm 0.23	93.55 \pm 0.07	97.02 \pm 0.07	99.12 \pm 0.00	88.63 \pm 0.08
D ^{aug} (Ours)	N	\sim 0.6M	\sim 0.16M	78.58 \pm 0.15	85.02 \pm 0.15	90.87 \pm 0.09	94.98 \pm 0.09	97.55 \pm 0.05	99.23 \pm 0.01	91.04 \pm 0.04

original dataset.

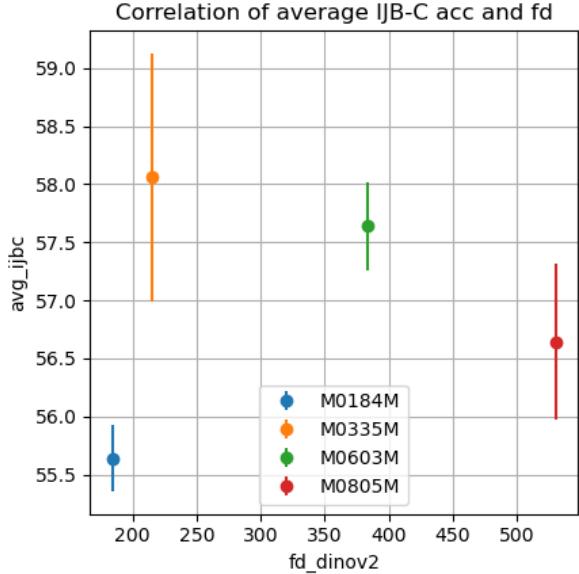
Table 9. Effect of mixing different numbers of classes (#Class) and samples per class (#Sample) with the original data, CASIA-WebFace. For TinyFace Rank-1 and Rank-5 verification accuracies are presented as TR1 and TR5 respectively. The numbers under columns labeled like C/B-1e-6 indicate TAR for IJB-C/B at FPR of 1e-6.

Syn #Class \times #Sample	n^r	B-1e-6	B-1e-5	C-1e-6	C-1e-5	TR1	TR5
0	0.5M	1.16 \pm 0.08	5.61 \pm 1.64	0.83 \pm 0.10	5.86 \pm 1.31	58.01 \pm 0.28	63.47 \pm 0.07
Ours (5k \times 5)	0.5M	0.85 \pm 0.06	5.60 \pm 0.84	0.65 \pm 0.08	6.70 \pm 0.97	58.19 \pm 0.20	63.48 \pm 0.01
Ours (5k \times 20)	0.5M	1.08 \pm 0.16	5.81 \pm 1.01	0.84 \pm 0.12	6.88 \pm 1.38	57.50 \pm 0.13	63.07 \pm 0.33
Ours (5k \times 50)	0.5M	0.63 \pm 0.23	4.56 \pm 0.41	0.46 \pm 0.10	6.55 \pm 0.35	57.39 \pm 0.20	62.55 \pm 0.11
Ours (10K \times 5)	0.5M	0.77 \pm 0.08	4.40 \pm 0.14	0.61 \pm 0.03	4.69 \pm 0.26	58.30 \pm 0.28	63.28 \pm 0.30
Ours (10K \times 20)	0.5M	1.29 \pm 0.01	8.21 \pm 1.38	1.43 \pm 0.22	9.67 \pm 1.01	58.01 \pm 0.50	63.00 \pm 0.71
Ours (10K \times 50)	0.5M	0.62 \pm 0.17	4.29 \pm 0.27	0.64 \pm 0.10	5.98 \pm 0.00	57.51 \pm 0.32	62.77 \pm 0.08

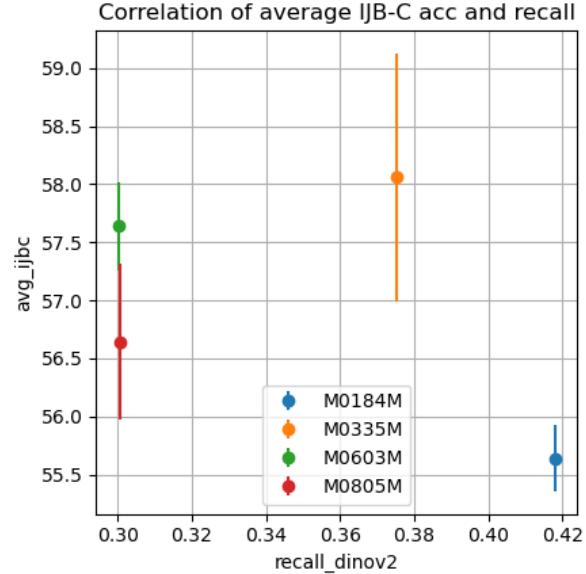
F. Downstream Performance vs Metrics in Generative Models

In this section, we examine whether there is a correlation between common metrics for evaluating generative models and the discriminator’s performance when trained on our augmented dataset. We studied the FD (Heusel et al., 2017) Precision/Recall (Sajjadi et al., 2018; Kynkänniemi et al., 2019) and Coverage (Naeem et al., 2020) which is usually used to quantify the performance of the Generative Models. Calculation of these metrics requires the projection of the images into meaningful feature spaces. For feature extraction, we consider two backbones, Inception-V3 (Szegedy et al., 2016) and DINoV2 (Oquab et al., 2023) which the latter shown effective for evaluating diffusion models (Stein et al., 2023). Both these models were trained using the ImageNet (Russakovsky et al., 2015) in a supervised and semi-supervised manner respectively. Experiments were performed by randomly selecting 100,000 images of both CASIA-WebFace (as the source distribution) and our generated images by the value of α and β using Algorithm 1 (*i.e.*, the same settings as presented in the Section 4). We are reporting four versions of our generated augmentation using a medium-sized generator when it sees 184M, 335M, 603M, and 805M training samples in different noise scales of the original CASIA-WebFace (M for Million). For each of the

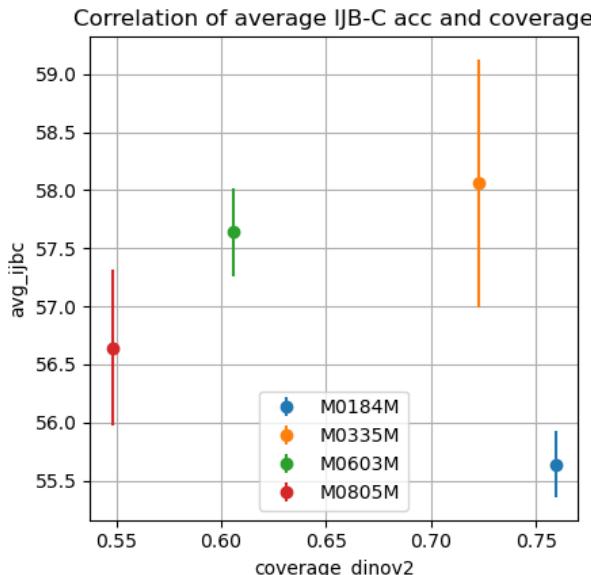
classes generated from these models, we selected 20 samples, based on the previous observation in Table 9. Later by mixing the selected images with the original CASIA-WebFace we train FR for each of them and report the average accuracies for different thresholds in the IJB-C (*i.e.*, similar to **Avg** column in the Table 8). Figure 6 and Figure 7 are showing mentioned metrics for Inception-V3 and DINOv2 feature extractor respectively. We observe no clear correlation between the metrics used to evaluate generative models and the performance of a downstream task. When comparing D^{aug} to D^{orig} for FD, a higher FD (*i.e.*, distinguishable D^{aug} images) should enhance discriminator performance, but that wasn't observed here. This holds when we are augmenting the dataset for training the generator and discriminator with the D^{orig} . This highlights the need to develop new evaluation metrics as a proxy.



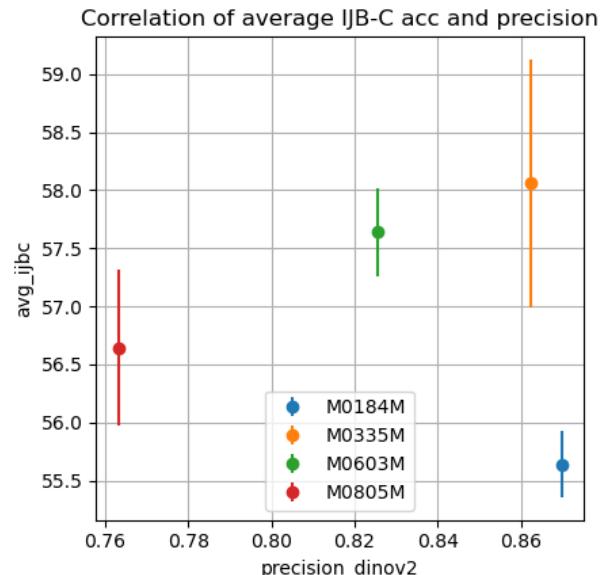
(a) FD



(b) Recall



(c) Coverage



(d) Precision

Figure 6. Correlation between the FD, Recall, Coverage, and Precision for the generated dataset by comparing it with the features of CASIA-WebFace using the DINOv2 extractor.

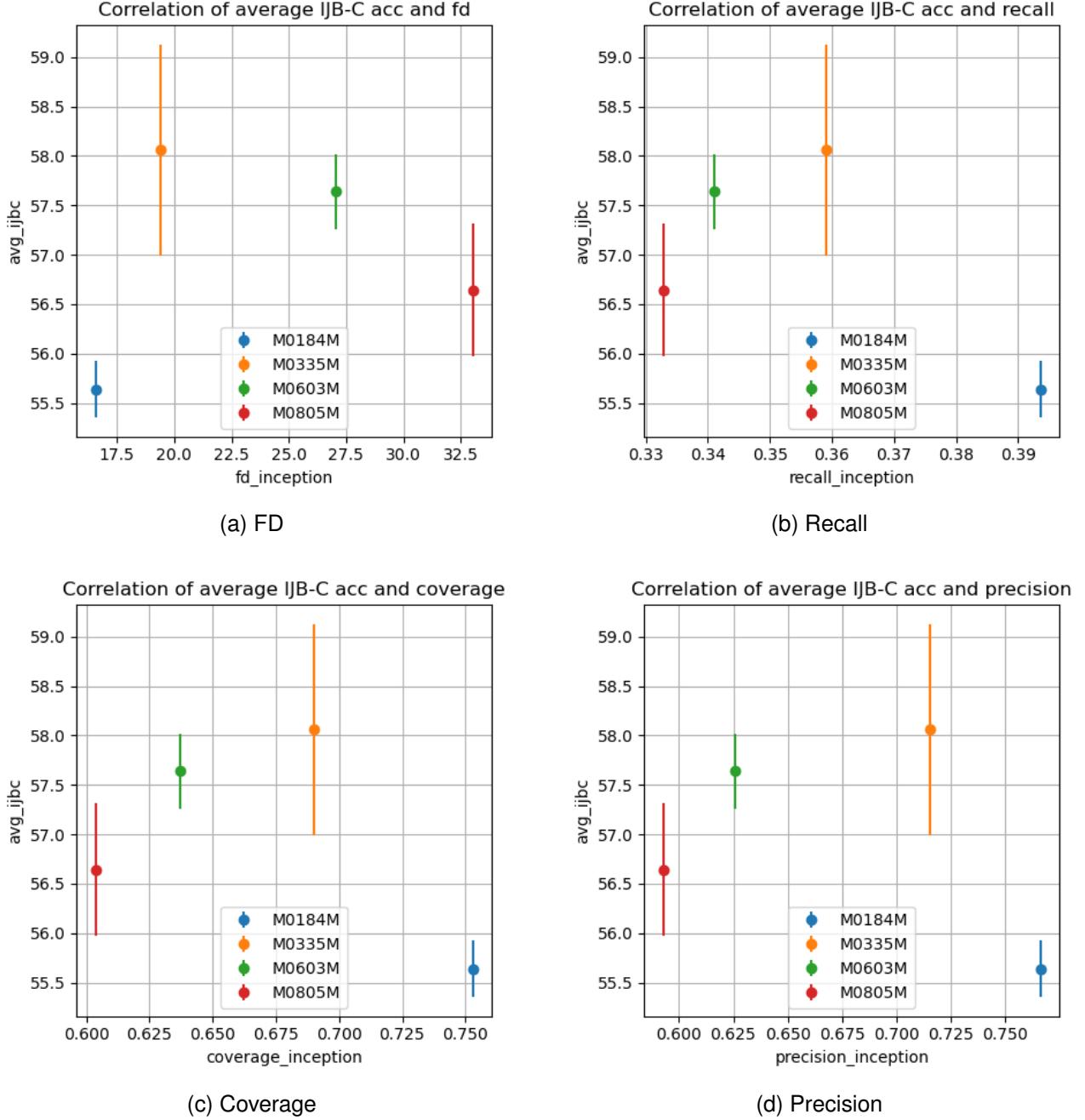


Figure 7. Correlation between the FD, Recall, Coverage, and Precision for the generated dataset by comparing it with the features of CASIA-WebFace using Inception-v3 extractor.

G. Effectiveness of Grid Search

We also study the effectiveness of our proposed method in Algorithm 1 which tries to find the suitable condition weights, α , and β . We compare with four sets of values:

- Rand: α and β were selected randomly for 10,000 mixture of identities from the set of $\{0.1, 0.3, 0.5, 0.7, 0.9, 1.0, 1.1\}$.
- Half: α and β set to 0.5 for all 10,000 random mixture of identities selected from \mathbb{L}_s .

- Full: α and β set to 1 for all 10,000 random mixture of identities selected from \mathbb{L}_s .
- Half++: α and β set to 0.7 according to the Algorithm 1 for the generator and discriminator trained on CASIA-WebFace dataset. This is done for all 10,000 random mixture of identities selected from \mathbb{L}_s

The results for this are shown in the Table 10. We observe on almost all of the benchmarks the D^{aug} generated using α and β values with higher m^{total} are performing better.

Table 10. Effectiveness of our weighting procedure (W/ Half++) in comparison to (W/ Random) or when putting the conditions to 0.5 (W/ Half) and when setting the condition signal to 1 (W/ Full). Best in bold, second best, underlined. TR1 represents the Rank-1 accuracy for the TinyFace benchmark. The numbers under columns labeled like C/B-1e-6 indicate TAR for IJB-C/B at FPR of 1e-6

C Weight Method	n^s	n^r	B-1e-6	B-1e-5	C-1e-6	C-1e-5	TR1	m^{total}
W/ Half	~0.5M	0	8.52±5.61	27.74±6.87	11.59±4.26	35.69±5.23	46.42±0.60	1.48
W/ Full	~0.5M	0	17.63±0.08	32.47±0.47	24.30±0.80	37.45±0.22	45.08±0.17	1.53
W/ Random	~0.5M	0	<u>24.47±1.23</u>	<u>39.83±1.08</u>	<u>30.79±1.39</u>	<u>44.33±0.88</u>	49.34±0.31	N/A
W/ Half++	~0.5M	0	25.44±0.19	46.20±0.12	39.66±0.38	51.47±0.29	<u>47.95±0.09</u>	1.58

H. Verifying the driving Hypothesis

As shown in Figure 1, introducing a new class using Algorithm 2, aims to augment the original dataset with a novel mix of source classes. This approach enforces the network to improve the compactness and separability of class representations. By requiring the network to distinguish the mixed class from its source classes, we strengthen its discriminative power. To validate this approach, we conducted experiments on two models, $f_{\theta_{\text{dis}}}^{\text{Baseline}}$ and $f_{\theta_{\text{dis}}}^{\text{AugGen}}$, trained before and after incorporating AugGen samples, respectively, and evaluated their performance using the following metrics:

1. **Mean** absolute Inter-Class Similarity of samples across all mixed classes. After applying AugGen, we expect that the average similarity of samples from different classes become lower, corresponding to a higher θ_{ours} in Figure 1.
2. **Mean** and standard deviation (*i.e.*, **std**) of Intra-Class Similarity of samples of all mixed classes, (*i.e.*, M-Intra and S-Intra in Table 11). This should indicate if the generated samples for each class, cause the model to boost its compactness.

These metrics are presented in the Table 11. After adding the AugGen samples, we are observing lower M-Inter which reflects that the similarity of the samples between different classes decreased. We are also observing the M-Intra increase reflecting that the networks perceive the images of the same class as more similar.

Table 11. Comparison of models trained with and without AugGen samples: *M-Inter* represents interclass similarity, indicating class separation, while *M-Intra* and *S-Intra* measure the mean and standard deviation of intraclass similarity, reflecting class compactness.

Dataset/Method	n^s	n^r	M-Inter(\downarrow)	M-Intra(\uparrow)	S-Intra(\downarrow)
Baseline	0	0.16M	0.0672	0.49065	0.13499
AugGen	0.2M	0.16M	0.0664	0.54917	0.12807

I. More Samples of D^{aug}

In the following figures, you can find more examples of generated images for Small and Medium-sized generators and also trained for more steps. By comparing Figure 8 (generated result from a small-sized generator trained when it sees 335M images (~ 700 Epochs), **S335M**, as the optimization of score-function, involves multiple noise levels of images), Figure 9 (**M335M**) and Figure 13 (**M805M**) we generally observe that larger generators are producing better images, but training for more steps does not necessarily translate to better image quality. This is especially important as we are exploring the out-of-distribution generation capabilities of an image generator.

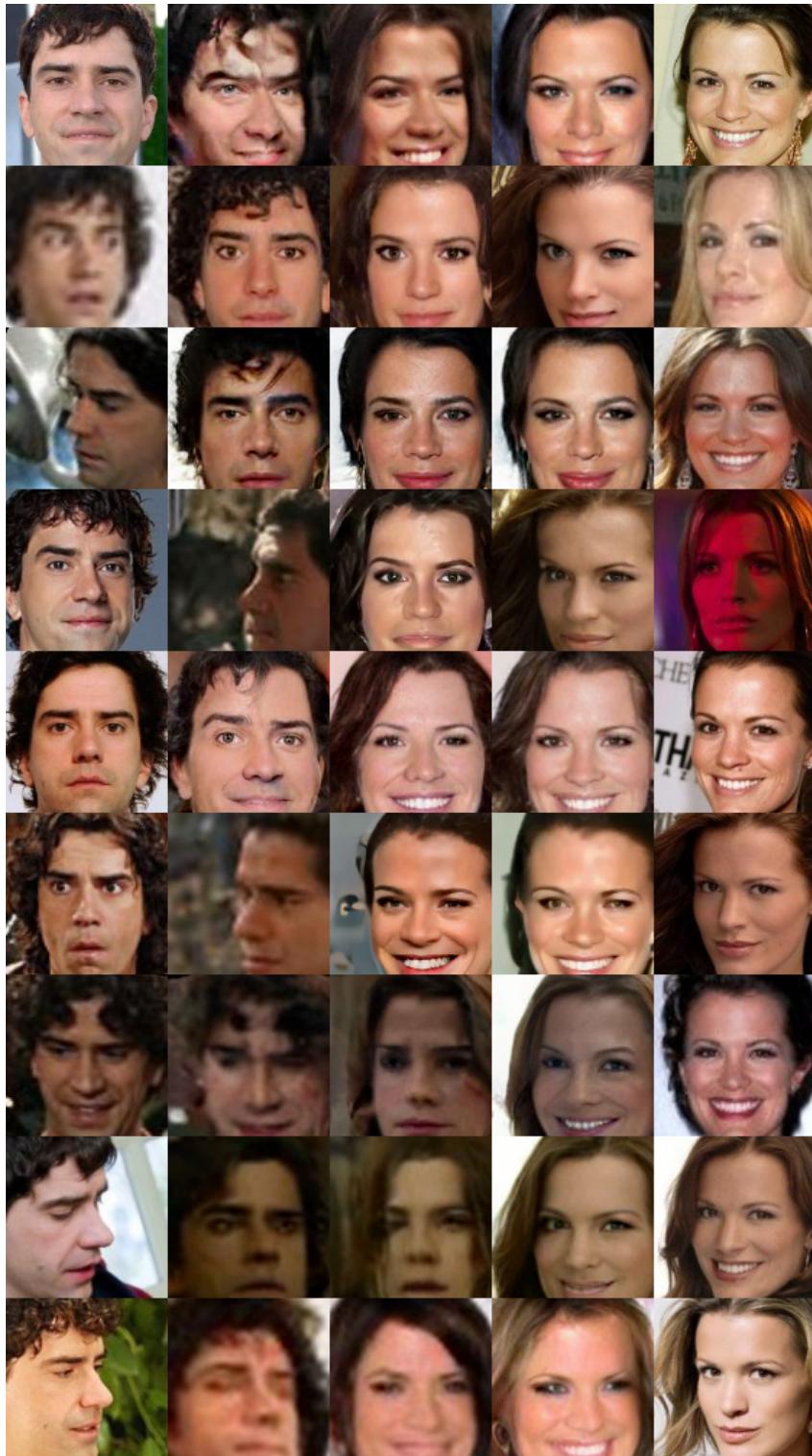


Figure 8. Small-sized generator trained till it sees 335M images in different noise levels (~ 700 Epochs). From left to right, the first column is variations of a random ID, 1, in the, D^{orig} , the second column is the recreation of the same ID in the first column using the generator when we set the corresponding conditions to 1. The last two columns are the same but for different IDs and the middle column representing the D^{aug} sample.



Figure 9. Medium-sized generator trained till it sees 335M images in different noise levels (~ 700 Epochs). From left to right, the first column is variations of a random ID, 1, in the, D^{orig} , the second column is the recreation of the same ID in the first column using the generator when we set the corresponding conditions to 1. The last two columns are the same but for different IDs and the middle column representing the D^{aug} sample.



Figure 10. Medium-sized generator trained till it sees 805M images in different noise levels (~ 1500 Epochs). From left to right, the first column is variations of a random ID, 1, in the, D^{orig} , the second column is the recreation of the same ID in the first column using the generator when we set the corresponding conditions to 1. The last two columns are the same but for different IDs and the middle column representing the D^{aug} sample.

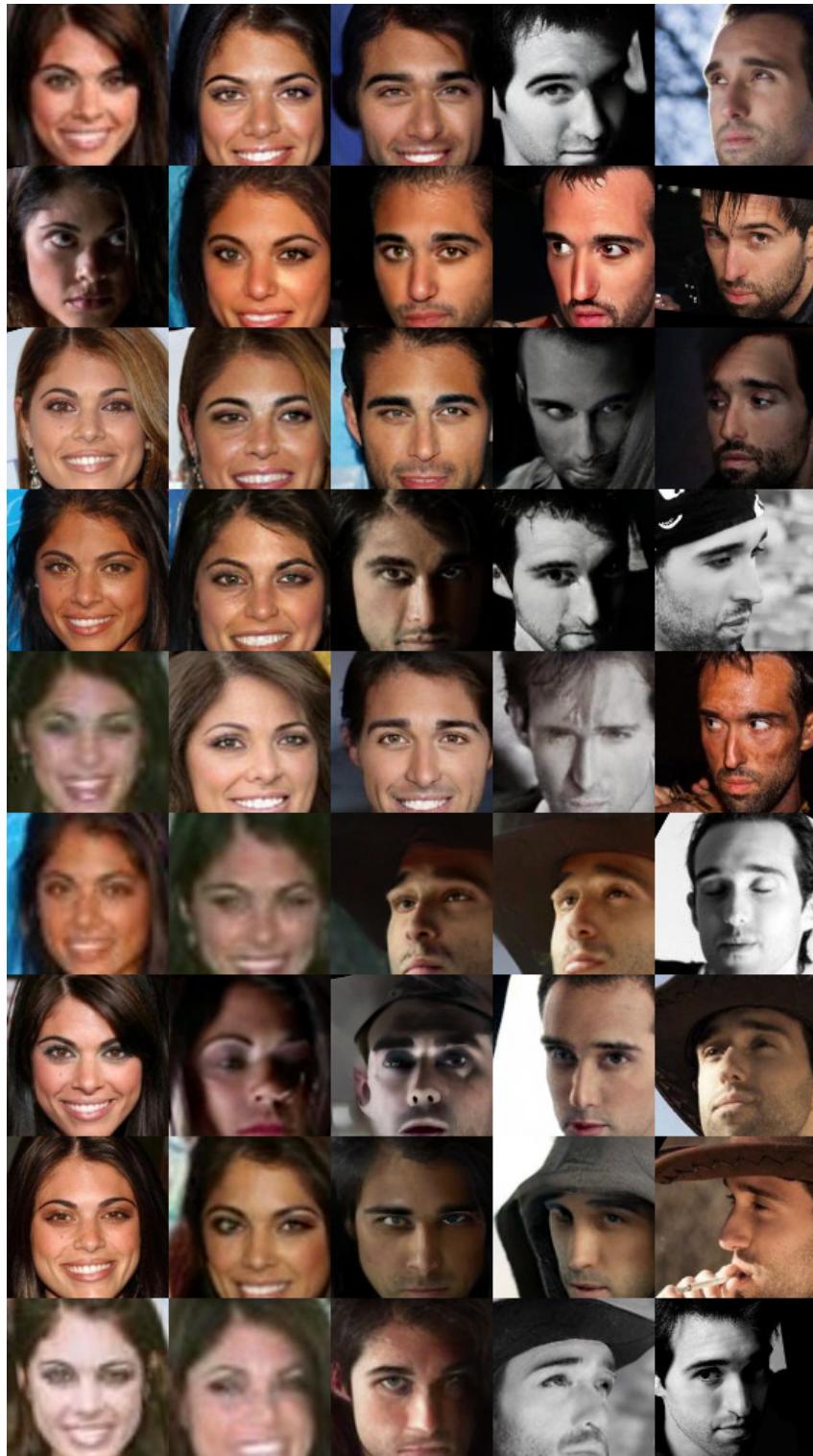
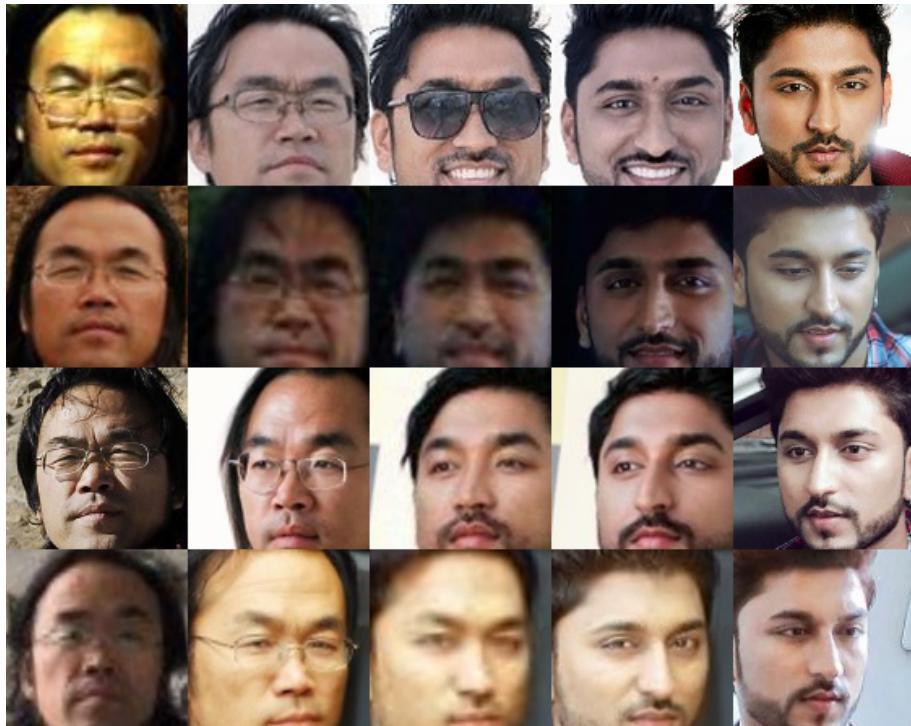


Figure 11. Medium-sized generator trained for till it sees 335M images in different noise levels (~ 700 Epochs) for different IDs. From left to right, the first column is variations of a random ID, 1, in the, D^{orig} , the second column is the recreation of the same ID in the first column using the generator when we set the corresponding conditions to 1. The last two columns are the same but for different IDs and the middle column representing the D^{aug} sample.



(a) IDs 115 and 2668



(b) IDs 760 and 1297

Figure 12. Samples from a small-sized pixel space EDM generator trained on WebFace160K for about 31M training steps (~ 200 Epochs). From left to right, the first column is variations of a random ID, 1, in the D^{orig} , the second column is the recreation of the same ID in the first column using the generator when we set the corresponding conditions to 1. The last two columns are the same but for different IDs and the middle column represents the D^{aug} sample.



(a) IDs 2299 and 8574



(b) IDs 7858 and 8434

Figure 13. Samples from a small-sized pixel space EDM generator trained on WebFace160K for about 31M training steps (~ 200 Epochs). From left to right, the first column is variations of a random ID, 1, in the D^{orig} , the second column is the recreation of the same ID in the first column using the generator when we set the corresponding conditions to 1. The last two columns are the same but for different IDs and the middle column representing the D^{aug} sample.

Reproducibility.

All code for the discriminative and generative models, along with the generated datasets and trained models, will be publicly available for reproducibility.

Impact Statement

In our approach, we introduce a novel technique that leverages generative models to further improve state-of-the-art (SOTA) facial recognition (FR) systems, as demonstrated on publicly available medium-sized datasets. However, these same FR systems can inadvertently facilitate unauthorized identity preservation in deepfakes and other forms of fraudulent media when attackers mimic individuals without their consent.

While our primary objective is to address privacy concerns and informed consent in training FR systems, the resulting performance gains could also enhance deepfake quality.