Controllable Inversion of Black-Box Face Recognition Models via Diffusion

Manuel Kansy\textsuperscript{1,2} *, Anton Raël\textsuperscript{1}, Graziana Mignone\textsuperscript{2}, Jacek Naruniec\textsuperscript{2}, Christopher Schroers\textsuperscript{2}, Markus Gross\textsuperscript{1,2}, and Romann M. Weber\textsuperscript{2}

\textsuperscript{1}ETH Zurich, Switzerland, \textsuperscript{2}DisneyResearch|Studios, Switzerland

\{mkansy, grossm\}@inf.ethz.ch, anrael@student.ethz.ch, \{<first>.<last>\}@disneyresearch.com

Abstract

Face recognition models embed a face image into a low-dimensional identity vector containing abstract encodings of identity-specific facial features that allow individuals to be distinguished from one another. We tackle the challenging task of inverting the latent space of pre-trained face recognition models without full model access (i.e., black-box setting). A variety of methods have been proposed in literature for this task, but they have serious shortcomings such as a lack of realistic outputs and strong requirements for the data set and accessibility of the face recognition model. By analyzing the black-box inversion problem, we show that the conditional diffusion model loss naturally emerges and that we can effectively sample from the inverse distribution even without an identity-specific loss. Our method, named identity denoising diffusion probabilistic model (ID3PM), leverages the stochastic nature of the denoising diffusion process to produce high-quality, identity-preserving face images with various backgrounds, lighting, poses, and expressions. We demonstrate state-of-the-art performance in terms of identity preservation and diversity both qualitatively and quantitatively, and our method is the first black-box face recognition model inversion method that offers intuitive control over the generation process.

1. Introduction

Face recognition (FR) systems are omnipresent. Their applications range from classical use cases such as access control to newer ones such as tagging a picture by identity or controlling the output of generative models \cite{31,2,47}. The goal of an FR method $f$ is to obtain embeddings $y$ of face images $x$ such that the embeddings of images of the same person are closer to each other than those of images of other people. We refer to this embedding $y$ as the identity vector or ID vector. In this paper, we propose a technique to sample from $p(x|y)$, i.e., to produce realistic face images from an ID vector.

By design, the many-to-one mapping of FR methods assigns multiple images of a given identity to the same ID vector. The inverse one-to-many problem, i.e., producing a high-dimensional image from a low-dimensional ID vector, is extremely challenging. Previous methods often rely on the gradient of FR models either directly \cite{74} or use it during training in the form of a loss function \cite{9,47}. This
gradient or information about the model's architecture and weights is often not available, e.g. if using an API of a proprietary model. We therefore focus on the more generally applicable black-box setting, where only the resulting ID vectors are available. In addition to being more general, the black-box setting simplifies the analysis of different FR models as explored in the supplementary material. Another benefit is that we can easily extend our conditioning mechanism to include information from different, even non-differentiable, sources (e.g. labels, biological signals).

We propose the identity denoising diffusion probabilistic model (ID3PM), the first method that uses a diffusion model (DM) to invert the latent space of an FR model, i.e. to generate identity-preserving face images conditioned solely on black-box ID vectors as seen in Fig. 1. We show mathematically that we can effectively invert a model \( f \) even without access to its gradients by using a conditional DM. This allows us to train our method with an easy-to-obtain data set of pairs of images and corresponding ID vectors (easily extracting from images) without an identity-specific loss term.

Our method obtains state-of-the-art performance for the inversion task and is, to the best of our knowledge, the first black-box FR model inversion method with control over the generation process as seen in Fig. 1. Specifically, we can control (1) the diversity among samples generated from the same ID vector via the classifier-free guidance scale, (2) identity-specific features (e.g. age) via smooth transitions in the ID vector latent space, and (3) identity-agnostic features (e.g. pose) via explicit attribute conditioning.

To summarize, our main contributions are:
1. Showing that the conditional diffusion model loss naturally emerges from an analysis of the black-box inversion problem.
2. Applying the resulting framework to invert face recognition models without identity-specific loss functions.
3. Demonstrating state-of-the-art performance in generating diverse, identity-preserving face images from black-box ID vectors.
4. Providing control mechanisms for the face recognition model inversion task.

2. Related work

2.1. Face recognition

While early deep learning works such as DeepFace [64] and VGG-Face [48] treated FR as a classification problem (one class per identity), FaceNet [59] introduced the triplet loss, a distance-based loss function. The trend then shifted towards margin-based softmax methods [37, 67, 66, 14] that incorporate a margin penalty and perform sample-to-class rather than sample-to-sample comparisons. More recently, some FR methods tackle specific challenges such as robustness to different quality levels [33] and occlusions [36, 49].

2.2. Inversion of face recognition models

Similar to gradient-based feature visualization techniques [61, 38, 72, 44], Zhmoginov and Sandler [74] perform gradient ascent steps using the gradient of a pre-trained FR model to generate images that approach the same ID vector as a target image. To avoid generating adversarial examples, strong image priors such as a total-variation loss and a guiding image are necessary. Cole et al. [9] transform the one-to-many task into a one-to-one task by mapping features of an FR model to frontal, neutral-expression images, which requires a difficult-to-obtain data set. Nitzan et al. [47] map the identity features and attributes of images into the style space of a pre-trained StyleGAN [28] to produce compelling results. However, their method struggles to encode real images since it is trained exclusively with images generated by StyleGAN. Furthermore, all of the above methods require white-box access to (the gradient of) an FR model, which is not always available in practice.

Many black-box methods view the problem from a security lens, focusing on generating images that deceive an FR model rather than appearing realistic. Early attempts using linear [43] or radial basis function models [42] lacked generative capacity to produce realistic images. NbNet [40] introduces a neighborly de-convolutional neural network that can generate images with a reasonable resemblance to a given image, but it has line artifacts and relies on a huge data set augmented with a GAN. On the contrary, Razhigaev et al. [52] propose a data-set-free method using Gaussian blobs (which we call “Gaussian sampling” for simplicity), but they need thousands of FR model queries (10-15 minutes) per image, and their results lack realism. Yang et al. [70] rely on background knowledge to invert a model and only produce blurry images in the black-box setting. Vec2Face [16] uses a bijection metric and knowledge distillation from a black-box FR model to produce realistic identity-preserving faces; however, it requires a large data set with multiple images per identity (Casia-WebFace [71]) during training. The method by Vendrow and Vendrow [65] (which we call “StyleGAN search”) searches the latent space of a pre-trained StyleGAN2 [29] to find images with an ID vector close to the target. While their search strategy generates highly realistic images, it needs hundreds of FR model queries (5-10 minutes) per image and often lands in local minima, resulting in completely different identities.

Table 1 compares attributes of state-of-the-art FR model inversion methods. Ours is the only one that generates diverse, realistic, identity-preserving images in the black-box setting, can be trained with easy-to-obtain data, and only requires one FR model query during inference.

2.3. Diffusion models for inverse problems

A number of approaches for solving inverse problems in a more general setting using conditional [57, 55] and uncon-
and minimizing it via gradient descent on $x$ from some starting point $x_0$ according to

$$\Delta x_t = -\nabla_x L = \left( \frac{\partial f}{\partial x} \right)^\top (y - f(x)).$$  

(2)

In common cases where the inverse problem is one-to-many, we can take a statistical approach. Here we want to sample from $p(x|y)$, which is equivalent to drawing from the pre-image set that defines the inverse $f^{-1}(y)$.

However, if we assume a Gaussian observation model

$$p(y|x) = \mathcal{N}(y; f(x), \sigma^2 I) \propto \exp \left( -\frac{L}{\sigma^2} \right),$$  

(3)

where the last term follows from (1), then we can rewrite equation (2) as $\Delta x_t \propto \sigma^2 \nabla_x \log p(y|x_t)$.

This shows that traditional model inversion via gradient descent performs a type of deterministic sampling from $p(y|x)$—and not the distribution we want, $p(x|y)$—by pushing toward modes of $p(y|x)$ close to the initialization point $x_0$, regardless of whether it possesses the desired characteristics of the data $p(x)$. This can lead to results, such as adversarial examples [19], that, while technically satisfying the mathematical criteria of inversion, do not appear to come from $p(x)$.

Various types of regularization exist to attempt to avoid this issue, which are most often ad hoc methods geared toward the specific problem at hand [74, 39, 10, 69]. A more general approach is to introduce regularization terms proportional to the (Stein) score, $\nabla_x \log p(x)$, since

$$\nabla_x \log p(x|y) = \nabla_x \log p(y|x) + \nabla_x \log p(x)$$

provides the conditional score needed to sample from $p(x|y)$, the distribution we are actually interested in.

Previous work has shown that diffusion models (DMs) effectively learn the score $\nabla_x \log p(x)$, which allows them to be used alongside model gradients to guide sampling [63, ...]

<table>
<thead>
<tr>
<th>Method</th>
<th>Black-box</th>
<th>FR model queries (inference)</th>
<th>Training data set</th>
<th>Realistic $^1$</th>
<th>Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhmoginov and Sandler [74]</td>
<td>No</td>
<td>$\sim 1000$ $^2$</td>
<td>Any images</td>
<td>No</td>
<td>One-to-one</td>
</tr>
<tr>
<td>Cole et al. [9]</td>
<td>No</td>
<td>1</td>
<td>Frontalized images</td>
<td>Yes</td>
<td>One-to-one</td>
</tr>
<tr>
<td>Nitzan et al. [47]</td>
<td>No</td>
<td>1</td>
<td>Any images</td>
<td>Yes</td>
<td>One-to-many</td>
</tr>
<tr>
<td>NbNet [40]</td>
<td>Yes</td>
<td>1</td>
<td>Huge data set</td>
<td>No</td>
<td>One-to-one</td>
</tr>
<tr>
<td>Gaussian sampling [52]</td>
<td>Yes</td>
<td>240000</td>
<td>Data-set-free</td>
<td>No</td>
<td>One-to-many</td>
</tr>
<tr>
<td>Yang et al. [70]</td>
<td>Yes</td>
<td>1</td>
<td>Any images</td>
<td>No</td>
<td>One-to-one</td>
</tr>
<tr>
<td>Vec2Face [16]</td>
<td>Yes</td>
<td>1</td>
<td>Multiple images per identity</td>
<td>Yes</td>
<td>One-to-many</td>
</tr>
<tr>
<td>StyleGAN search [65]</td>
<td>Yes</td>
<td>400</td>
<td>Data-set-free</td>
<td>Yes</td>
<td>One-to-many</td>
</tr>
<tr>
<td>ID3PM (Ours)</td>
<td>Yes</td>
<td>1</td>
<td>Any images</td>
<td>Yes</td>
<td>One-to-many</td>
</tr>
</tbody>
</table>

Table 1: Comparison of state-of-the-art face recognition (FR) model inversion methods. Our method does not have any of the common shortcomings, producing diverse, realistic images from black-box ID vectors with few requirements for the training data set or accessibility of the FR model during inference. $^1$ By visual inspection of the results of the respective papers. $^2$ The authors propose two methods: one taking hundreds or thousands of queries and the second one doing it in one shot.

3. Motivation

3.1. Inverse problems

In a system under study, we often have a forward problem or function $f$ that corresponds to a set of observations $y \sim \mathcal{Y}$. The function $f$ has input arguments $x$ and a set of parameters $\theta$, such that $f(x; \theta) = y$. An inverse problem seeks to reverse this process and make inferences about the values of $x$ or $\theta$ given the observations $y$. For the application explored in this work, $f$ is a face recognition model that takes an image $x$ as input and produces an ID vector $y$.

When the function $f$ is not bijective, no inverse exists in the traditional mathematical sense. However, it is possible to generalize our concept of what an inverse is to accommodate the problem of model inversion, namely by considering an inverse to be the set of pre-images of the function $f$ that map $\epsilon$-close to the target $y$. For bijective $f$, this corresponds to the traditional inverse for $\epsilon = 0$.

3.1.1 Model inversion with model access

One way to handle the model-inversion problem when $f$ is not bijective is to treat it pointwise, defining a loss, such as

$$L = \frac{1}{2} ||y - f(x)||^2,$$  

(1)

and minimizing it via gradient descent on $x$ from some starting point $x_0$ according to

$$\Delta x_t = -\nabla_x L = \left( \frac{\partial f}{\partial x} \right)^\top (y - f(x)).$$  

(2)

In common cases where the inverse problem is one-to-many, we can take a statistical approach. Here we want to sample from $p(x|y)$, which is equivalent to drawing from the pre-image set that defines the inverse $f^{-1}(y)$.

However, if we assume a Gaussian observation model

$$p(y|x) = \mathcal{N}(y; f(x), \sigma^2 I) \propto \exp \left( -\frac{L}{\sigma^2} \right),$$  

(3)

where the last term follows from (1), then we can rewrite equation (2) as $\Delta x_t \propto \sigma^2 \nabla_x \log p(y|x_t)$.

This shows that traditional model inversion via gradient descent performs a type of deterministic sampling from $p(y|x)$—and not the distribution we want, $p(x|y)$—by pushing toward modes of $p(y|x)$ close to the initialization point $x_0$, regardless of whether it possesses the desired characteristics of the data $p(x)$. This can lead to results, such as adversarial examples [19], that, while technically satisfying the mathematical criteria of inversion, do not appear to come from $p(x)$.

Various types of regularization exist to attempt to avoid this issue, which are most often ad hoc methods geared toward the specific problem at hand [74, 39, 10, 69]. A more general approach is to introduce regularization terms proportional to the (Stein) score, $\nabla_x \log p(x)$, since

$$\nabla_x \log p(x|y) = \nabla_x \log p(y|x) + \nabla_x \log p(x)$$

provides the conditional score needed to sample from $p(x|y)$, the distribution we are actually interested in.

Previous work has shown that diffusion models (DMs) effectively learn the score $\nabla_x \log p(x)$, which allows them to be used alongside model gradients to guide sampling [63, ...]}
When those models are classifiers, the procedure is known as classifier guidance [15]. However, this imposes an additional computational burden on sampling and also requires that the model \( f \) be differentiable.

### 3.1.2 Model inversion without full model access

In the case we focus on in this work, we assume to have access only to the values of the function \( f \) via some oracle or a lookup table of \((x, y)\) pairs but not its gradient \( \nabla f \). In this case, also referred to as black-box setting, we may wish to train a function \( g_{\psi} \) to learn the inverse by minimizing

\[
J = \| x - g_{\psi}(y) \|^2
\]

(4)

across all observed \((x, y)\). Recalling that \( y = f(x) \), we have essentially described an encoder-decoder setup with the encoder frozen and only the decoder being trained, which requires no gradients from the “encoder” \( f \).

If we consider perturbed data \( \tilde{x} = x + \epsilon \), where \( \epsilon \sim \mathcal{N}(0, \sigma^2 I) \). Then (4) is equivalent to

\[
\tilde{J} = \| (\tilde{x} - x) - (\tilde{x} - g_{\psi}(y)) \|^2 = \| \epsilon - \epsilon_{\theta}(\tilde{x}, y, t) \|^2,
\]

(5)

and we are now training a conditional model \( \epsilon_{\theta} \) to learn the noise added to \( x \) instead of a model \( g \) to reconstruct \( x \). This new task is exactly the one facing conditional diffusion models (Section 4.1).

Although we cannot force the model to leverage the conditioning on \( y \) or \( t \), if it is to successfully minimize the loss \( \tilde{J} \), it should learn a function proportional to the conditional score. That is because, by Tweedie’s formula [17, 32],

\[
\mathbb{E}[x|\tilde{x}, y] = \tilde{x} + \sigma^2 \nabla_{\tilde{x}} \log p(\tilde{x}|y) \approx \tilde{x} - \epsilon_{\theta}(\tilde{x}, y, t).
\]

(6)

As a result, we can effectively sample from the “inverse distribution” \( p(\tilde{x}|y) \) via \( \epsilon_{\theta}(\tilde{x}, y, t) \) using Langevin dynamics [1, 67] without having access to the gradients of the model \( f \) or any other model-specific loss terms.

Intuitively, during training, especially in early denoising steps, it is difficult for the DM to both denoise an image to get a realistic face and match the specific training image. The ID vector contains information (e.g., face shape) that the DM is incentivized to use (→ lower loss) to get closer to the training image. During inference, the random seed determines identity-agnostic features (→ many results), and the ID conditioning pushes the DM to generate an image that resembles the target identity.

### 4. Method

Motivated by the results from Sec. 3, we adopt a conditional diffusion model (DM) for the task of inverting a face recognition (FR) model. Since conditional DMs have inherent advantages for one-to-many and inversion tasks, this results in a minimal problem formulation compared to task-specific methods that require complicated supervision [16] or regularization [74] signals. The overall architecture of our method is visualized in Fig. 2.

#### 4.1. Diffusion model formulation

We build up on the diffusion model proposed by Dhariwal and Nichol [15]. Given a sample \( x_0 \) from the image dis-
distribution, a sequence \(x_1, x_2, \ldots, x_T\) of noisy images is produced by progressively adding Gaussian noise according to a variance schedule. At the final time step, \(x_T\) is assumed to be pure Gaussian noise: \(N(0, I)\). A neural network is then trained to reverse this diffusion process in order to predict \(x_{t-1}\) from the noisy image \(x_t\) and the time step \(t\). To sample a new image, we sample \(x_T \sim N(0, I)\) and iteratively denoise it, producing a sequence \(x_T, x_{T-1}, \ldots, x_1, x_0\). The final image, \(x_0\), should resemble the training data.

As [15], we assume that we can model \(p_\theta(x_{t-1}|x_t)\) as a Gaussian \(N(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))\) whose mean \(\mu_\theta(x_t, t)\) can be calculated as a function of \(\epsilon_\theta(x_t, t)\), the (unscaled) noise component of \(x_t\). We extend this by conditioning on the ID vector \(y\) and thus predict \(\epsilon_\theta(x_t, y, t)\). Extending [46] to the conditional case, we predict the noise \(\epsilon_\theta(x_t, y, t)\) and the variance \(\Sigma_\theta(x_t, y, t)\) from the image \(x_t\), the ID vector \(y\), and the time step \(t\), using the objective

\[
L_{\text{simple}} = \mathbb{E}_{t, x_t, y, \epsilon}[||\epsilon - \epsilon_\theta(x_t, y, t)||^2].
\]

(7)

For more details, refer to the diffusion model works [23, 46, 15]. Note that this objective is identical to the one theoretically derived in (5). While some recent work has considered the application of diffusion models to inverse problems, they typically assume \(p(y|x)\) is known [26, 30, 62, 6, 7, 20, 5, 8, 3, 41, 63], while we make no such assumption.

Following Ramesh et al. [51], we adapt classifier-free guidance [24] by setting the ID vector to the 0-vector with 10% probability during training (unconditional setting). During inference, we sample from both settings, and the model prediction \(\hat{\epsilon}_\theta\) becomes

\[
\hat{\epsilon}_\theta(x_t, y, t) = \epsilon_\theta(x_t, 0, t) + s[\epsilon_\theta(x_t, y, t) - \epsilon_\theta(x_t, 0, t)],
\]

(8)

where \(s \geq 1\) is the guidance scale. Higher guidance scales cause the generation process to consider the identity conditioning more.

4.2. Architecture

The model is a U-net [54] that takes the image \(x_t\), the ID vector \(y\), and the time step \(t\) as input. The U-net architecture is adapted from [15] and is described in detail in the supplementary material. To condition the diffusion model on the identity, we add an identity embedding to the residual connections of the ResNet blocks, as commonly done for class embeddings [15] and the CLIP [50] embedding in text-to-image generation methods [51, 56]. The identity embedding is obtained by projecting the ID vector through a learnable fully connected layer such that it has the same size as the time step embedding and can be added to it.

4.3. Controllability

Due to its robustness and ability to pick a mode by setting the random seed during image generation, our method permits smooth interpolations and analyses in the ID vector latent space unlike other works that invert FR models. For example, we can smoothly interpolate between different identities as visualized in Fig. 1. Furthermore, we can find meaningful directions in the latent spaces. Since the directions extracted automatically using principal component analysis (PCA) are generally difficult to interpret beyond the first dimension (see supplementary material), we calculate custom directions using publicly available meta-data [11] for the FFHQ data set. For binary features (e.g. glasses), we define the custom direction vector as the difference between the mean ID vectors of the two groups. For continuous features (e.g. age), we map to the binary case by considering ID vectors with feature values below the 10th percentile and values above the 90th percentile for the two groups respectively. Examples of traveling along meaningful ID vector directions can be seen in Fig. 1.

To better disentangle identity-specific and identity-agnostic information and obtain additional interpretable control, we can optionally extend our method by also conditioning the DM on an attribute vector as done for the ID vector. In practice, we recommend using only identity-agnostic attributes (referred to as set 1) along with the identity. In the supplementary material, we also show attribute sets that overlap more with identity (sets 2 & 3) for completeness.

4.4. Implementation details

As data set, we use FFHQ [28] and split it into 69000 images for training and 1000 for testing. As we can only show images of individuals with written consent (see Sec. 7), we use a proprietary data set of faces for the qualitative results in this paper. To condition our model, we use ID vectors from a PyTorch FaceNet implementation [59, 18] or the default InsightFace method [13]. To evaluate the generated images and thereby match the verification accuracy on real images shown in Vec2Face [16] as closely as possible, we use the official PyTorch ArcFace implementation [14, 12] and a TensorFlow FaceNet implementation [59, 58]. A detailed description of the remaining implementation details and ID vectors is in the supplementary material.

5. Experiments and results

5.1. Comparison to state-of-the-art methods

We mainly compare our model with the three methods that generate faces from black-box features whose code is available online: NbNet [40] (“vgg-percept-nbnetb” parameters), Gaussian sampling [52], and StyleGAN search [65].

Figure 3 compares the outputs of our method with those of current state-of-the-art methods. While capturing the identity of the input face well in some cases, NbNet [40] and Gaussian sampling [52] both fail to produce realistic faces. In contrast, StyleGAN search [65] always generates
For the quantitative evaluation of the identity preservation, we generate one image from each ID vector of all 1000 images of the FFHQ [28] test set for each method. We then calculate the distances according to the ArcFace [14, 12] and FaceNet [59, 58] face recognition methods for the 1000 respective pairs. The resulting distance distributions are plotted in Fig. 4. Note that StyleGAN search [65] optimizes the FaceNet distance during the image generation and thus performs well when evaluated with FaceNet but poorly when evaluated with ArcFace. The opposite effect can be seen for Gaussian sampling, which optimizes ArcFace during image generation. Despite not optimizing the ID vector distance directly (neither during training nor inference), our method outperforms all other methods, producing images that are closer to the original images’ identities.

To further evaluate the identity preservation and to compare to Vec2Face [16] despite their code not being available online, we follow the procedure used in Vec2Face [16].

Specifically, we use the official validation protocols of the LFW [25], AgeDB-30 [45], and CFP-FP [60] data sets and replace the first image in each positive pair with the image reconstructed from its ID vector, while keeping the second image as the real reference face. The face matching accuracies for ArcFace [14, 12] and FaceNet [59, 58] are reported in Tab. 2. Our method outperforms NbNet [40], Gaussian sampling [52], and StyleGAN search [65] in almost all tested configurations and performs on-par with or better than Vec2Face [16]. Note that our method has fewer requirements for the training data set (70000 images vs. 490000 images grouped into 10000 classes) and produces visually superior results compared to Vec2Face [16], as confirmed in the user study in the supplementary material.

To evaluate the diversity of the generated results, we generate 100 images for the first 50 identities of the FFHQ [28] test set. Motivated by the diversity evaluation common in unpaired image-to-image translation literature [4, 35], we calculate the mean pairwise LPIPS [73] distances among all images of the same identity. We further calculate the mean pairwise pose and expressions extracted using 3DDFA [21]. We additionally calculate the mean identity vector distances according to ArcFace [14, 12] and FaceNet [59, 58] to measure the identity preservation. We report these values in Tab. 3.

Since NbNet [40] is a one-to-one method and Gaussian sampling [52] produces faces that often fail to be detected by 3DDFA.V2 [21], we only compare with StyleGAN search [65]. In our default configuration (marked with * in Tab. 3), we obtain similar diversity scores as Style-
Table 2: Quantitative evaluation of the identity preservation with state-of-the-art methods. The scores depict the matching accuracy when replacing one image of each positive pair with the image generated from its ID vector for the protocols of the LFW [25], AgeDB-30 [45], and CFP-FP [60] data sets. The best performing method per column is marked in bold. \(^1\) Values taken from their paper.

GAN search [65], while preserving the identity much better. Note that the diversity scores are slightly skewed in favor of methods whose generated images do not match the identity closely since higher variations in the identity also lead to more diversity in the LPIPS [73] features.

### 5.2. Controllability

#### 5.2.1 Guidance scale

The classifier-free guidance scale offers control over the trade-off between the fidelity and diversity of the generated results. As seen in Fig. 5, by increasing the guidance, the generated faces converge to the same identity, resemble the original face more closely, and contain fewer artifacts. At the same time, higher guidance values reduce the diversity of identity-agnostic features such as the background and expressions and also increase contrast and saturation.

![Figure 5: Effect of the guidance scale on the generated images. For the (InsightFace [13]) ID vector extracted from the image on the left, we generate images for four seeds at guidance scales \(s\) ranging from 1.0 to 5.0.](image)

To measure this effect quantitatively, we perform the same evaluation as in the previous section and report the results in Tab. 3. As the guidance scale increases, the identity preservation improves as indicated by the decreasing identity distances, but the diversity in terms of poses, expressions, and LPIPS [73] features decreases. In practice, we choose a guidance scale of 2.0 for all experiments unless stated otherwise because that appears to be the best compromise between image quality and diversity. In the supplementary material, we further show FID [22] as well as precision and recall [34] values that measure how well the image distribution is preserved as the guidance scale varies.

#### 5.2.2 Identity vector latent space

As described in Sec. 4, we can find custom directions in the ID vector latent space that enable us to smoothly interpolate identities as well as change features such as the age or hair color as seen in Fig. 1 and in the supplementary material. Note that we refer to these features as identity-specific because they exist in the ID vector latent space. In theory, this space should not contain any identity-agnostic information such as the pose. In practice, however, some FR methods inadvertently do extract this information. This is shown in great detail in the supplementary material, where we show an interesting application of our method to analyze pre-trained face recognition methods.

#### 5.2.3 Attribute conditioning

By additionally conditioning our method on attributes, we can disentangle identity-specific and identity-agnostic features. As seen in Fig. 6, the additional attribute conditioning allows us to recover more of the original data distribution in terms of head poses and expressions whereas a model conditioned only on the ID vector is more likely to overfit and learn biases from the training data set. This is also shown in Tab. 3, where the diversity increases with attribute conditioning at the expense of worse identity preservation com-
Table 3: Quantitative evaluation of the diversity and identity distances of 100 generated images for 50 identities with StyleGAN search [65], different guidance scales, and attribute conditioning (set 1). InsightFace [13] ID vectors are used for our methods in this experiment. The best performing method per column is marked in bold. * Indicates the default setting used in this paper and also for the run with attribute conditioning.

<table>
<thead>
<tr>
<th>Method</th>
<th>Setting</th>
<th>Diversity</th>
<th>Identity distance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Pose ↑</td>
<td>Expression ↑</td>
</tr>
<tr>
<td>StyleGAN search [65]</td>
<td></td>
<td>12.57</td>
<td>1.57</td>
</tr>
<tr>
<td></td>
<td>Guidance scale = 1.0</td>
<td>17.36</td>
<td>1.35</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>16.69</td>
<td>1.18</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>16.24</td>
<td>1.10</td>
</tr>
<tr>
<td></td>
<td>2.5</td>
<td>15.88</td>
<td>1.05</td>
</tr>
<tr>
<td></td>
<td>3.0</td>
<td>15.55</td>
<td>1.01</td>
</tr>
<tr>
<td>ID3PM (Ours)</td>
<td></td>
<td>16.93</td>
<td>1.45</td>
</tr>
<tr>
<td>ID3PM (Ours)</td>
<td>Attribute conditioning</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

pared to the base configuration. The attribute conditioning also enables intuitive control over the generated images by simply selecting the desired attribute values as shown in Fig. 1 and in the supplementary material.

![ID](image1.png)

![ID + Set 1](image2.png)

**Figure 6**: Attribute conditioning diversity. Through additional attribute conditioning, we can disentangle identity-specific and identity-agnostic features. As a result, we obtain more diverse results when using both (InsightFace [13]) ID vector and attribute vector conditioning (set 1) compared to when only using ID vector conditioning.

### 6. Limitations

Our method outputs images at a relatively low resolution of $64 \times 64$. While this can be upscaled using super-resolution models, some fine identity-specific details such as moles cannot be modeled currently (but this information might not even be stored in the ID vector). Our method also has relatively long inference times (15 seconds per image when using batches of 16 images on one NVIDIA RTX 3090 GPU) in the default setting, but this can be reduced to less than one second per image when using 10 respacing steps at a slight decrease in quality as shown in the supplementary material. Our method also occasionally has small image generation artifacts, but the above aspects are expected to improve with future advancements in diffusion models. Lastly, our model inherits the biases of both the face recognition model and the training data set. This can manifest as either accessorizing images corresponding to certain demographic factors (e.g., via make-up, clothing) or losing identity fidelity for underrepresented groups as shown in the supplementary material. This suggests an additional application of our work to the study of systematic biases in otherwise black-box systems.

### 7. Ethical concerns

All individuals portrayed in this paper provided informed consent to use their images as test images. This was not possible for the images from the FFHQ [28], LFW [25], AgeDB-30 [45], and CFP-FP [60] data sets. Therefore, we do not show them in the paper and cannot provide qualitative comparisons to Vec2Face [16] (code not available).

We recognize the potential for misuse of any method that creates realistic imagery of human beings, especially when the images are made to correspond to specific individuals. We condemn such misuse and support ongoing research into the identification of artificially manipulated data.

### 8. Conclusion

We propose a method to generate high-quality identity-preserving face images by injecting black-box, low-dimensional embeddings of a face into the residual blocks of a diffusion model. We mathematically reason and empirically show that our method produces images close to the target identity despite the absence of any identity-specific loss terms. Our method obtains state-of-the-art performance on identity preservation and output diversity, as demonstrated qualitatively and quantitatively. We further showcase advantages of our approach in providing control over the generation process. We thus provide a useful tool to create data sets with user-defined variations in identities and attributes as well as to analyze the latent spaces of face recognition methods, motivating more research in this direction.
References

[18] Tim Esler. Face recognition using pytorch, 2021. 5, 6, 7


