

Diffusion Autoencoders: Toward a Meaningful and Decodable Representation

Konpat Preechakul Nattanat Chatthee Suttisak Wizadwongsa Supasorn Suwajanakorn
VISTEC, Thailand

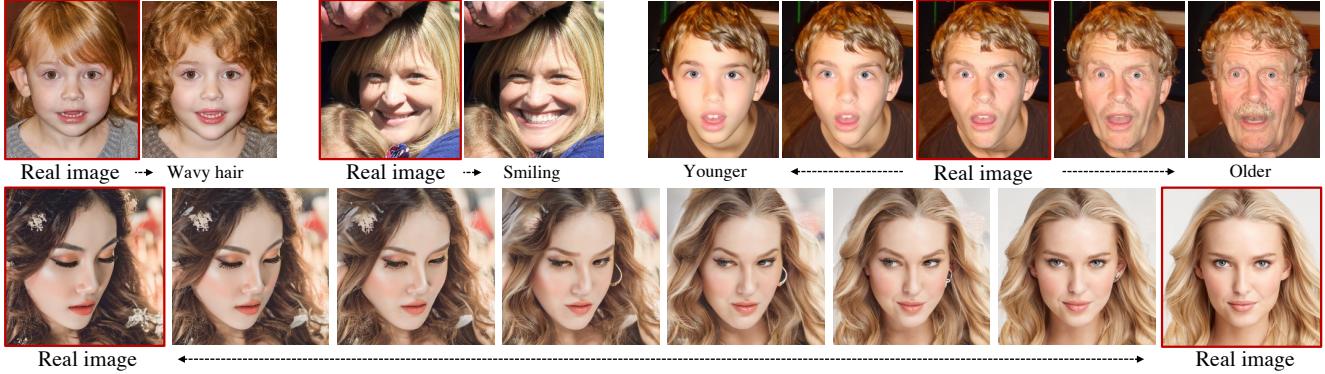


Figure 1. **Attribute manipulation and interpolation on real images.** Diffusion autoencoders can encode any image into a two-part latent code that captures both semantics and stochastic variations and allows near-exact reconstruction. This latent code can be interpolated or modified by a simple linear operation and decoded back to a highly realistic output for various downstream tasks.

Abstract

Diffusion probabilistic models (DPMs) have achieved remarkable quality in image generation that rivals GANs'. But unlike GANs, DPMs use a set of latent variables that lack semantic meaning and cannot serve as a useful representation for other tasks. This paper explores the possibility of using DPMs for representation learning and seeks to extract a meaningful and decodable representation of an input image via autoencoding. Our key idea is to use a learnable encoder for discovering the high-level semantics, and a DPM as the decoder for modeling the remaining stochastic variations. Our method can encode any image into a two-part latent code where the first part is semantically meaningful and linear, and the second part captures stochastic details, allowing near-exact reconstruction. This capability enables challenging applications that currently foil GAN-based methods, such as attribute manipulation on real images. We also show that this two-level encoding improves denoising efficiency and naturally facilitates various downstream tasks including few-shot conditional sampling. Please visit our page: <https://Diff-AE.github.io/>

1. Introduction

Diffusion-based (DPMs) [22, 46] and score-based [49] generative models have recently succeeded in synthesizing realistic and high-resolution images, rivaling those from GANs [11, 15, 23]. These two models are closely related

and, in practice, optimize similar objectives. Numerous applications have emerged notably in the image domain, such as image manipulation, translation, super-resolution [8, 32, 35, 43], in speech and text domains [5, 6], or 3D point cloud [34]. Recent studies have improved DPMs further in both theory and practice [25, 29, 31]. In this paper, however, we question whether DPMs can serve as a good representation learner. Specifically, we seek to extract a meaningful and decodable representation of an image that contains high-level semantics yet allows near-exact reconstruction of the image. Our exploration focuses on diffusion models, but the contributions are applicable also to score-based models.

One way to learn a representation is through an autoencoder. There exists a certain kind of DPM [47] that can act as an encoder-decoder that converts any input image x_0 into a spatial latent variable x_T by running the generative process backward. However, the resulting latent variable lacks high-level semantics and other desirable properties, such as disentanglement, compactness, or the ability to perform meaningful linear interpolation in the latent space. Alternatively, one can use a trained GAN for extracting a representation using the so-called GAN inversion [28, 58], which optimizes for a latent code that reproduces the given input. Even though the resulting code carries rich semantics, this technique struggles to faithfully reconstruct the input image. To overcome these challenges, we propose a diffusion-based autoencoder that leverages the powerful DPMs for decodable representation learning.

Finding a meaningful representation that is decodable requires capturing both the high-level semantics and low-level stochastic variations. Our key idea is to learn both levels of representation by utilizing a learnable encoder for discovering high-level semantics and utilizing a DPM for decoding and modeling stochastic variations. In particular, we use our conditional variant of the Denoising Diffusion Implicit Model (DDIM) [47] as the decoder and separate the latent code into two subcodes. The first “semantic” subcode is compact and inferred with a CNN encoder, whereas the second “stochastic” subcode is inferred by reversing the generative process of our DDIM variant conditioned on the semantic subcode. In contrast to other DPMs, DDIM modifies the forward process to be non-Markovian while preserving the training objectives of DPMs. This modification allows deterministically encoding an image to its corresponding initial noise, which represents our stochastic subcode.

The implication of this framework is two-fold. First, by conditioning DDIM on the semantic information of the target output, denoising becomes easier and faster. Second, this design produces a representation that is linear, semantically meaningful, and decodable—a novel property for DPMs’ latent variables. This crucial property allows harnessing DPMs for many tasks including those that are highly challenging for any GAN-based methods, such as interpolation and attribute manipulation on *real* images. Unlike GANs, which rely on error-prone inversion before operating on real images, our method requires no optimization to encode the input and produces high-quality output with original details preserved.

Despite being an autoencoder, which is generally not designed for unconditional generation, our framework can be used to generate image samples by fitting another DPM to the semantic subcode distribution. This combination achieves competitive FID scores on unconditional generation compared to a vanilla DPM. Moreover, the ability to sample from our compact and meaningful latent space also enables few-shot conditional generation (i.e., generate images with similar semantics to those of a few examples). Compared to other DPM-based techniques for the few-shot setup, our method produces convincing results with only a handful labeled examples without additional contrastive learning used in prior work [45].

2. Background

Diffusion-based (DPMs) and score-based generative models belong to a family of generative models that model the target distribution by learning a denoising process of varying noise levels. A successful process can denoise or map an arbitrary Gaussian noise map from the prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$ to a clean image sample after T successive denoising passes. Ho et al. [22] proposed to learn a function $\epsilon_\theta(\mathbf{x}_t, t)$ that takes a noisy image \mathbf{x}_t and predicts its noise

using a UNet [40]. The model is trained with a loss function $\|\epsilon_\theta(\mathbf{x}_t, t) - \epsilon\|$, where ϵ is the actual noise added to \mathbf{x}_0 to produce \mathbf{x}_t . This formulation is a simplified, reweighted version of the variational lower bound on the marginal log likelihood and has been commonly used throughout the community [11, 29, 36, 47].

More formally, we define a Gaussian diffusion process at time t (out of T) that increasingly adds noise to an input image \mathbf{x}_0 as $q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I})$, where β_t are hyperparameters representing the noise levels. With Gaussian diffusion, the noisy version of an image \mathbf{x}_0 at time t is another Gaussian $q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\sqrt{\alpha_t} \mathbf{x}_0, (1 - \alpha_t) \mathbf{I})$ where $\alpha_t = \prod_{s=1}^t (1 - \beta_s)$. We are interested in learning the reverse process of this, i.e., the distribution $p(\mathbf{x}_{t-1} | \mathbf{x}_t)$. This probability function is likely a complex one unless the gap between $t - 1$ and t is infinitesimally small ($T = \infty$) [46]. In such a case, $p(\mathbf{x}_{t-1} | \mathbf{x}_t)$ can be modeled as $\mathcal{N}(\mu_\theta(\mathbf{x}_t, t), \sigma_t)$ [22]. There are many ways to model this distribution, one of which is via $\epsilon_\theta(x_t, t)$ mentioned earlier. In practice, the assumption of $T = \infty$ is never satisfied; hence, DPMs are only approximations.

As latent-variable models, DPMs can naturally yield the latent variables $\mathbf{x}_{1:T}$ through its forward process; however, these variables are stochastic and only representing a sequence of image degradation by Gaussian noise, which does not contain much semantics. Song et al. [47] proposed another kind of DPM called Denoising Diffusion Implicit Model (DDIM) that enjoys the following deterministic generative process:

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}} \left(\frac{\mathbf{x}_t - \sqrt{1 - \alpha_t} \epsilon_\theta^t(\mathbf{x}_t)}{\sqrt{\alpha_t}} \right) + \sqrt{1 - \alpha_{t-1}} \epsilon_\theta^t(\mathbf{x}_t) \quad (1)$$

and the following novel inference distribution:

$$q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}} \mathbf{x}_0 + \sqrt{1 - \alpha_{t-1}} \frac{\mathbf{x}_t - \sqrt{\alpha_t} \mathbf{x}_0}{\sqrt{1 - \alpha_t}}, \mathbf{0}\right) \quad (2)$$

while maintaining the original DDPM marginal distribution $q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\sqrt{\alpha_t} \mathbf{x}_0, (1 - \alpha_t) \mathbf{I})$. By doing so, DDIM shares both the objective and solution with DDPM and only differs in how samples are generated.

With DDIM, it is possible to run the generative process backward deterministically to obtain the noise map \mathbf{x}_T , which represents the latent variable or encoding of a given image \mathbf{x}_0 . In this context, DDIM can be thought of as an image decoder that decodes the latent code \mathbf{x}_T back to the input image. This process can yield a very accurate reconstruction; however, \mathbf{x}_T still does not contain high-level semantics as would be expected from a meaningful representation. We show in Figure 4c that the interpolation between two latent variables \mathbf{x}_T ’s does not correspond to a semantically-smooth change in the resulting images. The images only share the overall composition and background colors but do not resemble the identity of either person. This

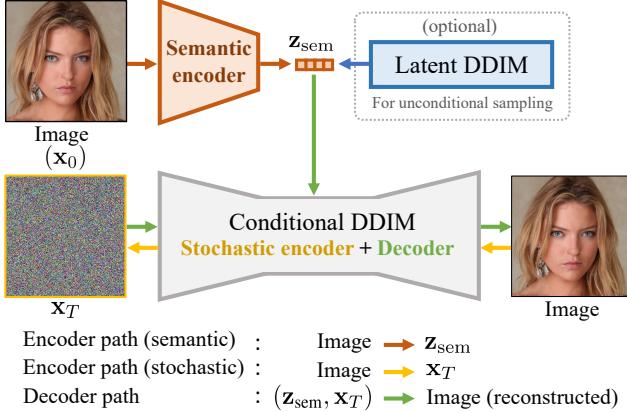


Figure 2. Overview of our diffusion autoencoder. The autoencoder consists of a “semantic” encoder that maps the input image to the semantic subcode ($x_0 \rightarrow z_{\text{sem}}$), and a conditional DDIM that acts both as a “stochastic” encoder ($x_0 \rightarrow x_T$) and a decoder ($(z_{\text{sem}}, x_T) \rightarrow x_0$). Here, z_{sem} captures the high-level semantics while x_T captures low-level stochastic variations, and together they can be decoded back to the original image with high fidelity. To sample from this autoencoder, we fit a latent DDIM to the distribution of z_{sem} and sample ($z_{\text{sem}}, x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$) for decoding.

is, perhaps, understandable as x_T is heavily influenced by the pixel values of x_0 due to an implicit linear bias from the marginals $q(x_T|x_0) = \mathcal{N}(\sqrt{\alpha_T}x_0, (1 - \alpha_T)\mathbf{I})$. This motivates approaches that augment DPMs with novel mechanisms to make their latent variables more meaningful, as will be proposed in this work.

3. Diffusion autoencoders

In the pursuit of a meaningful latent code, we design a conditional DDIM image **decoder** $p(x_{t-1}|x_t, z_{\text{sem}})$ that is conditioned on an additional latent variable z_{sem} , and a **semantic encoder** $z_{\text{sem}} = \text{Enc}_\phi(x_0)$ that learns to map an input image x_0 to a semantically meaningful z_{sem} . Here, the conditional DDIM decoder takes as input a latent variable $z = (z_{\text{sem}}, x_T)$, which consists of the high-level “semantic” subcode z_{sem} and a low-level “stochastic” subcode x_T , inferred by reversing the generative process of DDIM. In this framework, DDIM acts as both the decoder and the stochastic encoder. The overview is shown in Figure 2.

Unlike in other conditional DPMs [23, 32, 45] that use spatial conditional variables (e.g., 2D latent maps), our z_{sem} is a non-spatial vector of dimension $d = 512$, which resembles the style vector in StyleGAN [27, 28] and allows us to encode global semantics not specific to any spatial regions. One of our goals is to learn a semantically rich latent space that allows smooth interpolation, similar to those learned by GANs, while keeping the reconstruction capability that diffusion models excel.

3.1. Diffusion-based Decoder

Our conditional DDIM decoder receives as input $z = (z_{\text{sem}}, x_T)$ to produce the output image. This decoder is a conditional DDIM that models $p_\theta(x_{t-1}|x_t, z_{\text{sem}})$ to match the inference distribution $q(x_{t-1}|x_t, x_0)$ defined in Equation 2, with the following reverse (generative) process:

$$p_\theta(x_{0:T} | z_{\text{sem}}) = p(x_T) \prod_{t=1}^T p_\theta(x_{t-1} | x_t, z_{\text{sem}}) \quad (3)$$

$$p_\theta(x_{t-1}|x_t, z_{\text{sem}}) = \begin{cases} \mathcal{N}(\mathbf{f}_\theta(x_1, 1, z_{\text{sem}}), \mathbf{0}) & \text{if } t = 1 \\ q(x_{t-1}|x_t, \mathbf{f}_\theta(x_t, t, z_{\text{sem}})) & \text{otherwise} \end{cases} \quad (4)$$

Following Song et al. [47], we parameterize \mathbf{f}_θ in Equation 4 as a noise prediction network $\epsilon_\theta(x_t, t, z_{\text{sem}})$:

$$\mathbf{f}_\theta(x_t, t, z_{\text{sem}}) = \frac{1}{\sqrt{\alpha_t}} (x_t - \sqrt{1 - \alpha_t} \epsilon_\theta(x_t, t, z_{\text{sem}})) \quad (5)$$

This network is a modified version of the UNet of a recent DPM from Dhariwal et al. [11]. Training is done by optimizing L_{simple} [22] loss function with respect to θ and ϕ .

$$L_{\text{simple}} = \sum_{t=1}^T \mathbb{E}_{x_0, \epsilon_t} \left[\|\epsilon_\theta(x_t, t, z_{\text{sem}}) - \epsilon_t\|_2^2 \right] \quad (6)$$

where $\epsilon_t \in \mathbb{R}^{3 \times h \times w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $x_t = \sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon_t$, and T is set to some large number, e.g., 1,000. Note that this simplified loss function has been shown to optimize both DDPM [22] and DDIM [47], though not the actual variational lower bound. For training, the stochastic subcode x_T is not needed. We condition the UNet using adaptive group normalization layers (AdaGN), following Dhariwal et al. [11], which extend group normalization [56] by applying channel-wise scaling and shifting on the normalized feature maps $\mathbf{h} \in \mathbb{R}^{c \times h \times w}$. Our AdaGN is conditioned on t and z_{sem} :

$$\text{AdaGN}(\mathbf{h}, t, z_{\text{sem}}) = \mathbf{z}_s(t_s \text{GroupNorm}(\mathbf{h}) + \mathbf{t}_b) \quad (7)$$

where $\mathbf{z}_s \in \mathbb{R}^c = \text{Affine}(z_{\text{sem}})$ and $(\mathbf{t}_s, \mathbf{t}_b) \in \mathbb{R}^{2 \times c} = \text{MLP}(\psi(t))$ is the output of a multilayer perceptron with a sinusoidal encoding function ψ . These layers are used throughout the UNet. Please see details in Appendix A.

3.2. Semantic encoder

The goal of the semantic encoder $\text{Enc}(x_0)$ is to summarize an input image into a descriptive vector $z_{\text{sem}} = \text{Enc}(x_0)$ with necessary information to help the decoder $p_\theta(x_{t-1}|x_t, z_{\text{sem}})$ denoise and predict the output image. We do not assume any particular architecture for this encoder; however, in our experiments, this encoder shares the same architecture as the first half of our UNet decoder. One benefit of conditioning DDIM with information-rich z_{sem} is more efficient denoising process, which will be discussed further in Section 5.5.

3.3. Stochastic encoder

Besides decoding, our conditional DDIM can also be used to encode an input image \mathbf{x}_0 to the stochastic subcode \mathbf{x}_T by running its deterministic generative process backward (the reverse of Equation 1):

$$\mathbf{x}_{t+1} = \sqrt{\alpha_{t+1}} \mathbf{f}_\theta(\mathbf{x}_t, t, \mathbf{z}_{\text{sem}}) + \sqrt{1 - \alpha_{t+1}} \epsilon_\theta(\mathbf{x}_t, t, \mathbf{z}_{\text{sem}}) \quad (8)$$

We can think of this process as a **stochastic encoder** because \mathbf{x}_T is encouraged to encode only the information left out by \mathbf{z}_{sem} , which has a limited capacity for compressing stochastic details. By utilizing both semantic and stochastic encoders, our autoencoder can capture an input image to the very last detail while also providing a high-level representation \mathbf{z}_{sem} for downstream tasks. Note that the stochastic encoder is not used during training (Equation 6) and is used to compute \mathbf{x}_T for tasks that require exact reconstruction or inversion, such as real-image manipulation.

4. Sampling with diffusion autoencoders

By conditioning the decoder on \mathbf{z}_{sem} , diffusion autoencoders are no longer generative models. So, to sample from our autoencoder, we need an additional mechanism to sample $\mathbf{z}_{\text{sem}} \in \mathbb{R}^d$ from the latent distribution. While VAE is an appealing choice for this task, balancing between retaining rich information in the latent code and maintaining the sampling quality in VAE is hard [41, 42, 45, 52]. GAN is another choice, though it complicates training stability, which is one main strength of DPMs. Here, we choose to fit another DDIM, called **latent DDIM** $p_\omega(\mathbf{z}_{\text{sem},t-1} | \mathbf{z}_{\text{sem},t})$, to the latent distribution of $\mathbf{z}_{\text{sem}} = \text{Enc}_\phi(\mathbf{x}_0)$, $\mathbf{x}_0 \sim p(\mathbf{x}_0)$. Analogous to Equation 5 and 6, training the latent DDIM is done by optimizing L_{latent} with respect to ω :

$$L_{\text{latent}} = \sum_{t=1}^T \mathbb{E}_{\mathbf{z}_{\text{sem}}, \epsilon_t} \left[\|\epsilon_\omega(\mathbf{z}_{\text{sem},t}, t) - \epsilon_t\|_1 \right] \quad (9)$$

where $\epsilon_t \in \mathbb{R}^d \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{z}_{\text{sem},t} = \sqrt{\alpha_t} \mathbf{z}_{\text{sem}} + \sqrt{1 - \alpha_t} \epsilon_t$, and T is the same as in the DDIM image decoder. For L_{latent} , we empirically found that L_1 works better than L_2 loss. Unlike for 1D/2D images, there is no well-established DPM architecture for non-spatial data, but we have found that deep MLPs (10-20 layers) with skip connections perform reasonably well. The details are provided in Appendix A.1.

We first train the semantic encoder (ϕ) and the image decoder (θ) via Equation 6 until convergence. Then, we train the latent DDIM (ω) via Equation 9 with the semantic encoder fixed. In practice, the latent distribution modeled by the latent DDIM is first normalized to have zero mean and unit variance. Unconditional sampling from a diffusion autoencoder is thus done by sampling \mathbf{z}_{sem} from the latent DDIM and unnormalizing it, then sampling $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and finally decoding $\mathbf{z} = (\mathbf{z}_{\text{sem}}, \mathbf{x}_T)$ using the decoder.

Our choice of training the latent DDIM post-hoc has a few practical reasons. First, since training the latent DDIM takes only a fraction of the full training time, post-hoc training enables quick experiments on different latent DDIMs with the same diffusion autoencoder. Another reason is to keep \mathbf{z}_{sem} as expressive as possible by not imposing any constraints, such as the prior loss in VAE [30], that can compromise the quality of the latent variables.

5. Experiments

We now turn to assessing the properties of our learned latent space and demonstrating new capabilities, such as attribute manipulation and conditional generation. For fair comparison, the DDIM baseline in our experiments refers to our reimplementation of DDIM [47] based on an improved architecture of Dhariwal et al. [11] with the same UNet hyperparameters as our decoder. In short, the DDIM baseline is similar to our decoder except that it does not take \mathbf{z}_{sem} .

5.1. Latent code captures both high-level semantics and low-level stochastic variations

To demonstrate that high-level semantics are mostly captured in \mathbf{z}_{sem} and very little in \mathbf{x}_T , we first compute the semantic subcode $\mathbf{z}_{\text{sem}} = \text{Enc}(\mathbf{x}_0)$ from an input image \mathbf{x}_0 . For the stochastic subcode \mathbf{x}_T , instead of inferring it from the input, we will sample this subcode multiple times $\mathbf{x}_T^i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and decode multiple $\mathbf{z}^i = (\mathbf{z}_{\text{sem}}, \mathbf{x}_T^i)$. Figure 3 shows the variations induced by varying \mathbf{x}_T given the same \mathbf{z}_{sem} , as well as the variations from different \mathbf{z}_{sem} .

The result show that with a fixed \mathbf{z}_{sem} , the stochastic subcode \mathbf{x}_T only affects minor details, such as the hair and skin details, the eyes, or the mouth, but does not change the overall global appearance. And by varying \mathbf{z}_{sem} , we obtain completely different people with different facial shapes, illuminations, and overall structures. Quantitative results are discussed in Section 5.4 and Table 2.

5.2. Semantically meaningful latent interpolation

One desirable property of a useful latent space is the ability to represent semantic changes in the image by a simple linear change in the latent space. For example, by moving along a straight line connecting any two latent codes, we expect a smooth morphing between the corresponding two images. In Figure 4d and Figure 1, we show our interpolation results by encoding two input images into $(\mathbf{z}_{\text{sem}}^1, \mathbf{x}_T^1)$ and $(\mathbf{z}_{\text{sem}}^2, \mathbf{x}_T^2)$, then decode $\mathbf{z}(t) = (\text{Lerp}(\mathbf{z}_{\text{sem}}^1, \mathbf{z}_{\text{sem}}^2; t), \text{Slerp}(\mathbf{x}_T^1, \mathbf{x}_T^2; t))$ for various values of $t \in [0, 1]$, where linear interpolation is used for \mathbf{z}_{sem} and spherical linear interpolation is used for \mathbf{x}_T , following [47].

Compared to DDIM, which produces non-smooth transitions, our method gradually changes the head pose, background, and facial attributes between the two endpoints. The interpolation results from StyleGAN in both \mathcal{W} and

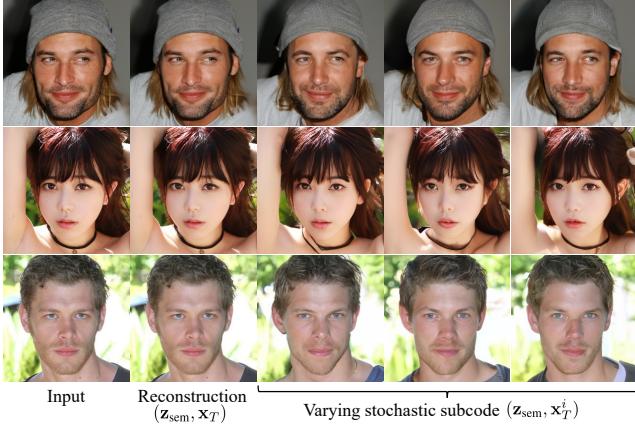


Figure 3. Reconstruction results and the variations induced by changing the stochastic subcode \mathbf{x}_T . Each row corresponds to a different \mathbf{z}_{sem} , which completely changes the person, whereas changing the stochastic subcode \mathbf{x}_T only affects minor details.

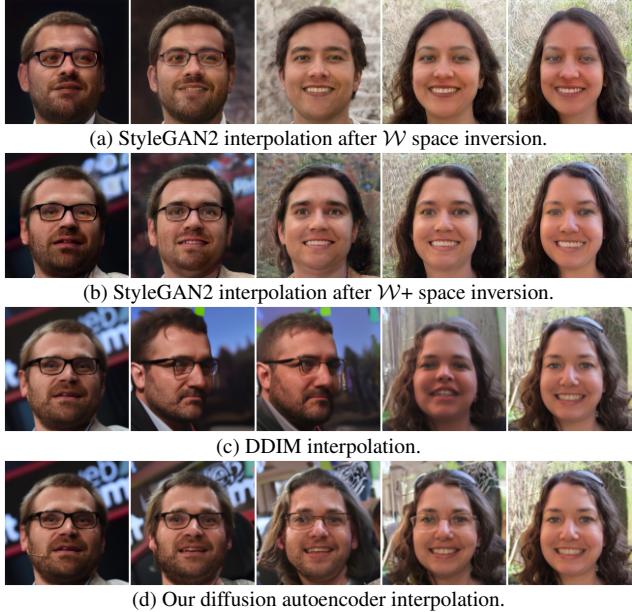


Figure 4. Interpolation between two real images. In contrast to StyleGAN2 and DDIM, our method produces smooth and consistent results with well-preserved original details from both images.

\mathcal{W}^+ spaces are smooth, but the two endpoints do not resemble the input images, whereas ours and DDIM's match the real input images almost exactly. We quantitatively evaluate how smooth our interpolation is in Appendix F.

5.3. Attribute manipulation on real images

Another way to assess the relationship between image semantics and linear motion or separability in the latent space is by moving the latent \mathbf{z}_{sem} of an image in a particular direction and observing changes in the image [44]. By finding such a direction from the weight vector of a linear classifier trained on latent codes of negative and positive im-

ages of a target attribute, e.g., smiling, this operation consequently changes the semantic attribute in the image. There exists specialized techniques for this task [3, 37, 44, 57], but here we aim to showcase the quality and applicability of our latent space by using the simplest linear operation.

We trained linear classifiers using images and attribute labels from CelebA-HQ [26] and tested on CelebA-HQ and FFHQ [27] in Figure 5. Implementation details and more results can be found in Appendix G. Note that our autoencoder was trained on FFHQ but can generalize to CelebA-HQ without fine-tuning the autoencoder. Our method is able to change local features, such as the mouth for smiling, while keeping the rest of the image and details mostly stationary. For global attributes that involve changing multiple features at the same time, such as aging, our results look highly plausible and realistic. Additionally, we compare the accuracy of these linear classifiers (Appendix E) that take \mathbf{z}_{sem} versus those taking StyleGAN's inverted \mathcal{W} as input. The AUROC \uparrow over 40 attributes of our method is 0.925 and of StyleGAN- \mathcal{W} is 0.891. And we test how much the input's identity is preserved via ArcFace [10] and quantify the manipulation quality in Appendix G.

One notable advantage of diffusion autoencoders over GAN-based manipulation techniques is the ability to manipulate *real* images while preserving details irrelevant to the manipulation (e.g., keeping the original hair and background when manipulating facial expression). When GANs are used for such tasks, the details are often altered because real images cannot be faithfully inverted back to the GAN's latent space. Compared to a recent score-based manipulation technique SDEdit [35], which focuses on local edits or translating images from another domain using a forward-backward sampling trick, our method solves changing semantic attributes by simply modifying the latent code. We also compare qualitatively to D2C [45], which uses NVAE [50] decoder to perform a similar task in our Appendix H.

5.4. Autoencoding reconstruction quality

Although good reconstruction quality of an autoencoder may not necessarily be an indicator of good representation learning, this property plays an important role in many applications, such as compression or image manipulation that requires accurate encoding-decoding abilities. For these tasks, traditional autoencoders that rely on MSE or L_1 loss functions perform poorly and produce blurry results. More advanced autoencoders combine perceptual loss and adversarial loss, e.g., VQGAN [13], or rely on a hierarchy of latent variables, e.g., NVAE [50], VQ-VAE2 [38]. Our diffusion autoencoder is an alternative design that produces a reasonable-size latent code with meaningful and compact semantic subcode and performs competitively with state-of-the-art autoencoders. The key is our two-level encoding that delegates the reconstruction of less compressible stochastic



Figure 5. Real-image attribute manipulation results on two global attributes (gender, age) and two local attributes (smile, wavy hair) by moving \mathbf{z}_{sem} along the positive or negative direction found by linear classifiers. The top two are from FFHQ [27] and the bottom two are from CelebA-HQ [26]. Our method synthesizes highly-plausible and realistic results that preserve an unprecedented level of detail.

details to our conditional DDIM.

In Table 1, we evaluate the reconstruction quality of 1) our diffusion autoencoder, 2) DDIM [47], 3) a pretrained StyleGAN2 [28] (via two types of inversion), 4) VQ-GAN [13], 5) VQ-VAE2 [38], 6) NVAE [50]. Both DDIM and ours were trained on 130M images and used $T=100$ for decoding. All these models were trained on FFHQ [27] and tested on 30k images from CelebA-HQ [26]. For our method and DDIM, we encoded downsampled test images of size 128×128 and decoded them back. For the others, we used publicly available pretrained networks for 256×256 and downsampled the results to the same 128×128 before comparison. For StyleGAN2, we performed inversion in \mathcal{W} [28] and $\mathcal{W}+$ [1, 2] spaces on the test images and used the optimized codes for reconstruction. The evaluation metrics are SSIM [53] (\uparrow), LPIPS [61] (\downarrow), and MSE. NVAE [50] achieves the lowest LPIPS and MSE scores, though it requires orders of magnitude larger latent dimension compared to others. Besides NVAE, our diffusion autoencoders outperform other models on all metrics, and only require $T=20$ steps to surpass DDIM with $T=100$ steps (Table 2).

Furthermore, we performed ablation studies to investigate 1) the reconstruction quality when only \mathbf{z}_{sem} is encoded from the input but \mathbf{x}_T is sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$ for decoding (Table 2.a), and 2) the effects of varying the dimension of \mathbf{z}_{sem} from 64 to 512 (Table 2.b-e) on our autoencoder trained with 48M images for expedience. All configs a)-e) produce realistic results but differ in the degree of fidelity, where higher latent dimensions are better. For config a) with 512D \mathbf{z}_{sem} , even though \mathbf{x}_T is random, the reconstructions still look perceptually close to the input images as measured

Table 1. Autoencoding reconstruction quality of models trained on FFHQ [27] and tested on unseen CelebA-HQ [26]. Our model is competitive with state-of-the-art NVAE while producing readily useful high-level semantics in a compact 512D \mathbf{z}_{sem} .

Model	Latent dim	SSIM \uparrow	LPIPS \downarrow	MSE \downarrow
StyleGAN2 (\mathcal{W}) [28]	512	0.677	0.168	0.016
StyleGAN2 ($\mathcal{W}+$) [28]	7,168	0.827	0.114	0.006
VQ-GAN [13]	65,536	0.782	0.109	3.61e-3
VQ-VAE2 [38]	327,680	0.947	0.012	4.87e-4
NVAE [50]	6,005,760	0.984	0.001	4.85e-5
DDIM ($T=100, 128^2$) [47]	49,152	0.917	0.063	0.002
Ours ($T=100, 128^2$, no \mathbf{x}_T)	512	0.677	0.073	0.007
Ours ($T=100, 128^2$)	49,664	0.991	0.011	6.07e-5

by LPIPS (also Figure 3). Our reconstruction with a small 64D \mathbf{z}_{sem} is already on par with StyleGAN2 inversion in 512D \mathcal{W} latent space, suggesting that our diffusion autoencoders are proficient in compression.

5.5. Faster denoising process

One useful benefit of conditioning the denoising process with semantic information from \mathbf{z}_{sem} is faster generation. One main reason DPMs require many generative steps is because DPMs can only use a Gaussian distribution to approximate $p(x_{t-1}|x_t)$ when T is sufficiently large (~ 1000). Recent attempts to improve sampling speed focus on finding a better sampling interval or noise schedule [25, 29, 31, 36], or using more efficient solvers to solve the score-based ODE counterpart [25]. Our diffusion autoencoders do not aim to solve this problem directly, nor can they be compared in the same context as generative models that lack access to the target samples. It is, however, worth mentioning the effects

Table 2. Ablation study results for a) autoencoding reconstruction quality when \mathbf{x}_T is not encoded from the input but sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$, and b-e) the effects of varying the dimension of \mathbf{z}_{sem} from 64 to 512 on our autoencoder trained with 48M images for expedience. In a), our reconstruction is perceptually close to the input images (LPIPS=0.073) even when \mathbf{x}_T is random. b-e) suggest that higher \mathbf{z}_{sem} dimensions lead to higher fidelity reconstruction. Our diffusion autoencoders with $T=20$ steps also surpass DDIM with $T=100$ steps.

Model	SSIM \uparrow				LPIPS \downarrow				MSE \downarrow			
	T=10	T=20	T=50	T=100	T=10	T=20	T=50	T=100	T=10	T=20	T=50	T=100
DDIM (@130M) [47]	0.600	0.760	0.878	0.917	0.227	0.148	0.087	0.063	0.019	0.008	0.003	0.002
Ours (@130M, 512D \mathbf{z}_{sem})	0.827	0.927	0.978	0.991	0.078	0.050	0.023	0.011	0.001	0.001	0.000	0.000
a) No encoded \mathbf{x}_T	0.707	0.695	0.683	0.677	0.085	0.078	0.074	0.073	0.006	0.007	0.007	0.007
b) No encoded \mathbf{x}_T , @48M, 512D \mathbf{z}_{sem}	0.662	0.650	0.637	0.631	0.102	0.096	0.093	0.092	0.009	0.009	0.009	0.010
c) No encoded \mathbf{x}_T , @48M, 256D \mathbf{z}_{sem}	0.637	0.624	0.612	0.606	0.116	0.109	0.106	0.105	0.010	0.011	0.011	0.011
d) No encoded \mathbf{x}_T , @48M, 128D \mathbf{z}_{sem}	0.613	0.600	0.588	0.582	0.133	0.127	0.125	0.124	0.012	0.012	0.013	0.013
e) No encoded \mathbf{x}_T , @48M, 64D \mathbf{z}_{sem}	0.551	0.538	0.527	0.521	0.168	0.165	0.163	0.162	0.018	0.019	0.020	0.020

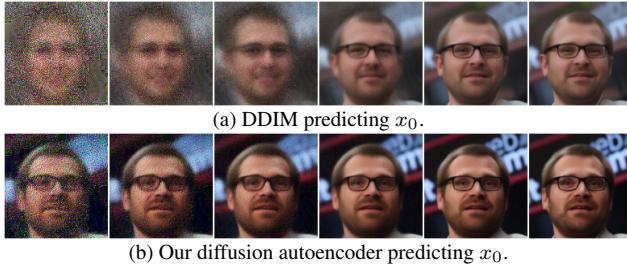


Figure 6. Predicted \mathbf{x}_0 at $t_{9,8,7,5,2,0}$ ($T=10$). By conditioning on \mathbf{z}_{sem} , our method predicts images that resemble \mathbf{x}_0 much faster.

they have within the DPM framework.

Consider a scenario where \mathbf{x}_0 is known to the denoising network. The noise prediction task will become trivial, and $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ is a Gaussian distribution regardless of the number of timesteps [22]. Since our diffusion autoencoders model the distribution $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{z}_{\text{sem}})$, it follows that $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{z}_{\text{sem}})$ is a better approximation to $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ than $p(\mathbf{x}_{t-1}|\mathbf{x}_t)$ when \mathbf{z}_{sem} has captured much information about \mathbf{x}_0 . Figure 6 shows that a diffusion autoencoder is able to predict \mathbf{x}_0 more accurately in fewer steps than DDIM and yield better image quality on four different datasets with the same timesteps T in Table 4.

5.6. Class-conditional sampling

This experiment demonstrates how our framework can be used for few-shot conditional generation and compares to D2C [45], a state-of-the-art DPM-based method for this setup. We follow the problem setup in D2C where the goal is to generate a diverse set of images of a target class, such as female, by utilizing a small number of labeled examples (≤ 100). The labels can specify both the positives and negatives with respect to the target class (binary scenario) or only the positives (positive and unlabeled, or PU scenario). Given a latent classifier $p_\gamma(c|\mathbf{z}_{\text{sem}})$ for a target class c , one simple way to do class-conditional sampling is with rejection sampling, as used by D2C. That is, we sample \mathbf{z}_{sem} from our latent DDIM and accept this sample with probability $p_\gamma(c|\mathbf{z}_{\text{sem}})$. We followed D2C’s methodology and con-

Table 3. FID scores (\downarrow) for class-conditional generation on CelebA 64 dataset computed between 5k sampled images and the target subset. \pm represents one standard deviation ($n=3$). D2C [45] results come from their paper ($n=1$ run of FID computation on 5k samples). Binary classifier was trained with 50 positives and 50 negatives. Positive-unlabeled (PU) classifier was trained with 100 positives and 10,000 unlabeled examples (as negatives). Naive FIDs were computed between all images and the target subset.

Scenario	Classes	Ours	D2C [45]	Naive
Binary	Male	11.52 \pm 1.19	13.44	23.83
	Female	7.29 \pm 0.44	9.51	13.64
	Blond	16.10 \pm 2.00	17.61	25.62
	Non-Blond	8.48 \pm 0.52	8.94	0.96
PU	Male	9.54 \pm 0.54	16.39	23.83
	Female	9.21 \pm 0.19	12.21	13.64
	Blond	7.01 \pm 0.25	10.09	25.62
	Non-Blond	7.91 \pm 0.15	9.09	0.96

ditionally sampled 5k images, then computed FID scores between these images and all the images of the same target class in CelebA dataset (with the same crop used by D2C). We used $T = 100$ for both latent and image generations. Table 3 shows that our method achieves comparable FID scores to D2C, despite not using any self-supervised contrastive learning used in D2C.

5.7. Unconditional sampling

To evaluate the quality of our unconditional samples from diffusion autoencoders, we first sample \mathbf{z}_{sem} from the latent DDIM, then decode $\mathbf{z} = (\mathbf{z}_{\text{sem}}, \mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I}))$ using our decoder. We trained our autoencoders on FFHQ [27], LSUN Horse & Bedroom [60], and CelebA [33]. For each dataset, we computed FID scores between 50k randomly sampled images from the dataset and our 50k generated images. We also varied the timestep $T = (10, 20, 50, 100)$ used in both latent DDIM and our main decoder.

As shown in Table 4, our diffusion autoencoders are competitive with DDIM baselines and produce higher FID scores in most cases across numbers of timesteps. We also provide as reference our diffusion autoencoders trained with

Table 4. FID scores (\downarrow) for unconditional generation. Our method is competitive with DDIM baselines. “+ autoencoding” refers to diffusion autoencoders that infer ground-truth semantic subcode from the test set and do not sample from the latent DDIM.

Dataset	Model	FID \downarrow			
		T=10	T=20	T=50	T=100
FFHQ 128	DDIM	29.56	21.45	15.08	12.03
	Ours	20.80	16.70	12.57	10.59
	+ autoencoding	14.43	10.70	6.69	4.56
Horse 128	DDIM	22.17	12.92	7.92	5.97
	Ours	11.97	9.37	7.44	6.71
	+ autoencoding	9.27	6.23	3.87	2.92
Bedroom 128	DDIM	13.70	9.23	7.14	5.94
	Ours	10.69	8.19	6.50	5.70
	+ autoencoding	6.36	4.88	3.61	2.88
CelebA 64	DDIM	16.38	12.70	8.52	5.83
	Ours	12.92	10.18	7.05	5.30
	+ autoencoding	12.78	9.06	5.15	3.11

ground-truth latent variables encoded from the test images, labeled “+autoencoding.” In every dataset, perhaps unsurprisingly, conditioning the DDIM decoder with \mathbf{z}_{sem} significantly improves the quality with small T s. In Appendix C, we show qualitative results and an additional experiment to verify that the latent DDIM does not memorize its input.

6. Related work

Denoising diffusion-based generative models [22, 46] are closely related to denoising score-based generative models [48]. Models under this family have been shown to produce images with high quality rivaling those of GANs [11] without using the less stable adversarial training. They are also used widely for multiple conditional generation tasks, such as image super-resolution [32, 43], image conditional generation [8, 35], class-conditional generation in ImageNet dataset [11], and mel-spectrogram conditional speech synthesis [6]. Similar to our work, these methods rely on conditional DPMs; however, most conditioning signals in prior work are known a priori and fixed, while our diffusion autoencoder augments the latent variable with an end-to-end learnable signal that the CNN encoder discovers. This puts our work closer to VAE [30], particularly Wehenkel et al. [54] and D2C [45]. While these only utilize DPMs to model the prior distribution or latent representation for another generative model [12], our focus is on how DPMs can be augmented with meaningful latent codes.

Our diffusion autoencoders share common goals with other kinds of autoencoders such as VAE [30], NVAE [50], and VQ-VAE [52] and VQ-VAE2 [38]. While VAEs provide reasonable latent quality and sample quality, they are subject to posterior collapse [52] and prior holes problems [45], whereas DPMs are not. VQ-VAE with discrete latent variables was proposed to deal with these problems by fitting an autoregressive Pixel-CNN model to the latent variable post-hoc [51]. Fitting the latent variable post-hoc

is also used in our work, but we utilize another DPM instead of an autoregressive model. Rich image representations are useful for many downstream tasks; for example, VAE are often used in model-based reinforcement learning [14, 17, 18] for predicting future outcomes of the environment. VQ-VAE’s latent variables are used as a means for video generation tasks [59]. Our diffusion autoencoders also provide useful representations with an added ability to decode the representations back near perfectly.

Besides producing impressive image samples, GANs [15] have been shown to learn meaningful latent spaces [27] with extensive studies on multiple derived spaces [24, 57] and various knobs and controls for conditional human face generation [21, 37, 55]. Encoding an image to the GAN’s latent space requires an optimization-based inversion process [28, 58] or an external image encoder [39], which has limited reconstruction fidelity (or yields high-dimensional codes outside the learned manifold). This problem may be related to the GAN’s limited latent size and mode-collapse problem, where the latent space only partially covers the support of training samples. Diffusion autoencoders do not have this problem and can readily encode any image without any additional error-prone optimization.

7. Limitations & Discussion

When encoding images that are out of the training distribution, our diffusion autoencoders can still reconstruct the images well, owing to the high-dimensional stochastic subcode from DDIM. However, both the inferred semantic and stochastic subcodes may fall outside the learned distributions, resulting in a poor representation that can no longer be interpreted or interpolated. While our choice of using non-spatial latent code is suitable for learning global semantics, certain image and spatial reasoning tasks may require more precise local latent variables. For these tasks, incorporating 2D latent maps can be beneficial.

For image generation, one unique feature of StyleGAN that is lacking from our diffusion autoencoders is the ability to control scale-specific generation. In terms of generation speed, our framework has significantly reduced the timesteps needed to achieve high-quality samples from our DDIM but still lacks behind GANs, which only require a single generator’s pass to generate an image.

In conclusion, we have presented diffusion autoencoders that can separately infer both semantics and stochastic information from an input image. In contrast to DPMs and high-fidelity autoencoders like NVAE, our latent representation allows near-exact decoding while containing compact semantics readily useful for downstream tasks. These properties enable simple solutions to various *real-image* editing tasks without requiring GANs and their error-prone inversion. Our framework also improves denoising efficiency and retains competitive unconditional sampling of DPMs.

References

- [1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space? pages 4432–4441, 2019.
- [2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN++: How to Edit the Embedded Images? pages 8296–8305, 2020.
- [3] Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. StyleFlow: Attribute-conditioned Exploration of StyleGAN-Generated Images using Conditional Continuous Normalizing Flows. *ACM Transactions on Graphics*, 40(3):1–21, May 2021. arXiv: 2008.02401.
- [4] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv:1610.01644 [cs, stat]*, Nov. 2018. arXiv: 1610.01644.
- [5] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured Denoising Diffusion Models in Discrete State-Spaces. May 2021.
- [6] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, and William Chan. WaveGrad: Estimating Gradients for Waveform Generation. *arXiv:2009.00713 [cs, eess, stat]*, Oct. 2020. arXiv: 2009.00713.
- [7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597–1607. PMLR, Nov. 2020. ISSN: 2640-3498.
- [8] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. ILVR: Conditioning Method for Denoising Diffusion Probabilistic Models. *arXiv:2108.02938 [cs]*, Sept. 2021. arXiv: 2108.02938.
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009. ISSN: 1063-6919.
- [10] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. pages 4690–4699, 2019.
- [11] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion Models Beat GANs on Image Synthesis. May 2021.
- [12] Patrick Esser, Robin Rombach, Andreas Blattmann, and Björn Ommer. ImageBART: Bidirectional Context with Multinomial Diffusion for Autoregressive Image Synthesis. May 2021.
- [13] Patrick Esser, Robin Rombach, and Björn Ommer. Taming Transformers for High-Resolution Image Synthesis. pages 12873–12883, 2021.
- [14] Daniel Freeman, David Ha, and Luke Metz. Learning to Predict Without Looking Ahead: World Models Without Forward Prediction. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- [15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.
- [16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020.
- [17] David Ha and Jürgen Schmidhuber. Recurrent World Models Facilitate Policy Evolution. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.
- [18] Danijar Hafner, Timothy P. Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning Latent Dynamics for Planning from Pixels. Jan. 2019.
- [19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. pages 9729–9738, 2020.
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. pages 770–778, 2016.
- [21] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. AttGAN: Facial Attribute Editing by Only Changing What You Want. *IEEE Transactions on Image Processing*, 28(11):5464–5478, Nov. 2019. Conference Name: IEEE Transactions on Image Processing.
- [22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.
- [23] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded Diffusion Models for High Fidelity Image Generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.
- [24] Erik Häkkinen, Aaron Hertzmann, Jaakkko Lehtinen, and Sylvain Paris. GANSpace: Discovering Interpretable GAN Controls. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9841–9850. Curran Associates, Inc., 2020.
- [25] Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. Gotta Go Fast When Generating Data with Score-Based Models. *arXiv:2105.14080 [cs, math, stat]*, May 2021. arXiv: 2105.14080.
- [26] Tero Karras, Timo Aila, Samuli Laine, and Jaakkko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *arXiv:1710.10196 [cs, stat]*, Feb. 2018. arXiv: 1710.10196.
- [27] Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. pages 4401–4410, 2019.

- [28] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. pages 8110–8119, 2020.
- [29] Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational Diffusion Models. May 2021.
- [30] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. Dec. 2013.
- [31] Max W. Y. Lam, Jun Wang, Rongjie Huang, Dan Su, and Dong Yu. Bilateral Denoising Diffusion Models. *arXiv:2108.11514 [cs, eess]*, Sept. 2021. arXiv: 2108.11514.
- [32] Haoying Li, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueling Chen. SRDiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, Mar. 2022.
- [33] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, Dec. 2015. ISSN: 2380-7504.
- [34] Shitong Luo and Wei Hu. Diffusion Probabilistic Models for 3D Point Cloud Generation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2836–2844, Nashville, TN, USA, June 2021. IEEE.
- [35] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. Sept. 2021.
- [36] Alexander Quinn Nichol and Prafulla Dhariwal. Improved Denoising Diffusion Probabilistic Models. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8162–8171. PMLR, July 2021. ISSN: 2640-3498.
- [37] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery. pages 2085–2094, 2021.
- [38] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating Diverse High-Fidelity Images with VQ-VAE-2. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- [39] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in Style: A StyleGAN Encoder for Image-to-Image Translation. pages 2287–2296, 2021.
- [40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, pages 234–241, Cham, 2015. Springer International Publishing.
- [41] Mihaela Rosca, Balaji Lakshminarayanan, and Shakir Mohamed. Distribution Matching in Variational Inference. *arXiv:1802.06847 [cs, stat]*, June 2019. arXiv: 1802.06847.
- [42] Oleh Rybkin, Kostas Daniilidis, and Sergey Levine. Simple and Effective VAE Training with Calibrated Decoders. In *Proceedings of the 38th International Conference on Machine Learning*, pages 9179–9189. PMLR, July 2021. ISSN: 2640-3498.
- [43] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image Super-Resolution via Iterative Refinement. *arXiv:2104.07636 [cs, eess]*, June 2021. arXiv: 2104.07636.
- [44] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the Latent Space of GANs for Semantic Face Editing. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9240–9249, June 2020. ISSN: 2575-7075.
- [45] Abhishek Sinha, Jiaming Song, Chenlin Meng, and Stefano Ermon. D2C: Diffusion-Decoding Models for Few-Shot Conditional Generation. May 2021.
- [46] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2256–2265. PMLR, June 2015. ISSN: 1938-7228.
- [47] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. Sept. 2020.
- [48] Yang Song and Stefano Ermon. Generative Modeling by Estimating Gradients of the Data Distribution. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- [49] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. *arXiv:2011.13456 [cs, stat]*, Feb. 2021. arXiv: 2011.13456.
- [50] Arash Vahdat and Jan Kautz. NVAE: A Deep Hierarchical Variational Autoencoder. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19667–19679. Curran Associates, Inc., 2020.
- [51] Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, koray kavukcuoglu, Oriol Vinyals, and Alex Graves. Conditional Image Generation with PixelCNN Decoders. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.
- [52] Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural Discrete Representation Learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [53] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, Apr. 2004. Conference Name: IEEE Transactions on Image Processing.
- [54] Antoine Wehenkel and Gilles Louppe. Diffusion Priors In Variational Autoencoders. June 2021.

- [55] Po-Wei Wu, Yu-Jing Lin, Che-Han Chang, Edward Y. Chang, and Shih-Wei Liao. RelGAN: Multi-Domain Image-to-Image Translation via Relative Attributes. pages 5914–5922, 2019.
- [56] Yuxin Wu and Kaiming He. Group Normalization. pages 3–19, 2018.
- [57] Zongze Wu, Dani Lischinski, and Eli Shechtman. StyleSpace Analysis: Disentangled Controls for StyleGAN Image Generation. pages 12863–12872, 2021.
- [58] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. GAN Inversion: A Survey. *arXiv:2101.05278 [cs]*, Aug. 2021. arXiv: 2101.05278.
- [59] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. VideoGPT: Video Generation using VQ-VAE and Transformers. *arXiv:2104.10157 [cs]*, Sept. 2021. arXiv: 2104.10157.
- [60] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop. *arXiv:1506.03365 [cs]*, June 2016. arXiv: 1506.03365.
- [61] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, Salt Lake City, UT, June 2018. IEEE.

Appendix

A. Diffusion autoencoder architectures

The baseline diffusion models and our diffusion autoencoders are based on the same DDIM model [11] (publicly available at <https://github.com/openai/guided-diffusion>). The architecture is specified in Table 5. We selected the hyperparameters differently due to the limited computational resources. Note that we used the linear β scheduler as in Ho et al. [22], but we do observe improvements using the cosine β scheduler [36] in our preliminary results.

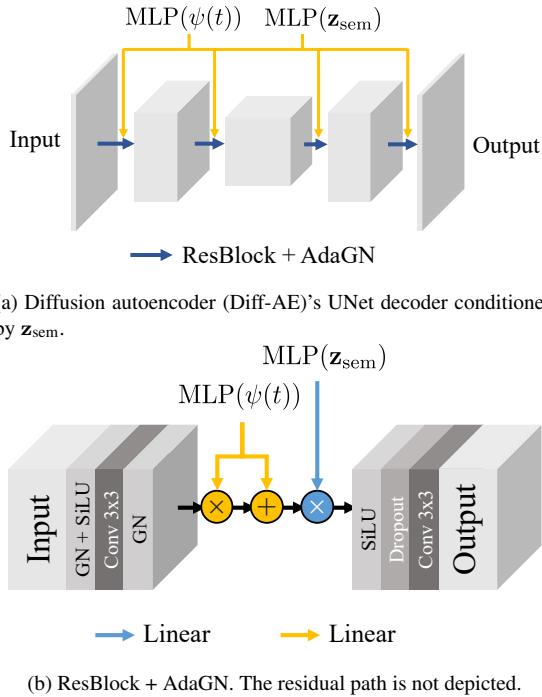


Figure 7. Architecture overview of our diffusion autoencoder.

A.1. Latent DDIM architectures

For latent DDIMs, we experimented with multiple architectures including MLP, MLP + skip connections, and projecting z_{sem} into a spatial vector before using a CNN or UNet. We have found that MLP + skip connection performed reasonably well while being very fast (See unconditional samples in Figure 20). The architecture is specified in Table 6. Each layer of the MLP has a skip connection from the input, which simply concatenates the input with the output from the previous layer. The network is conditioned on t by scaling the hidden representations to help denoising.

The architecture is shown in Figure 8 and the hyperparameters are shown in Table 6.

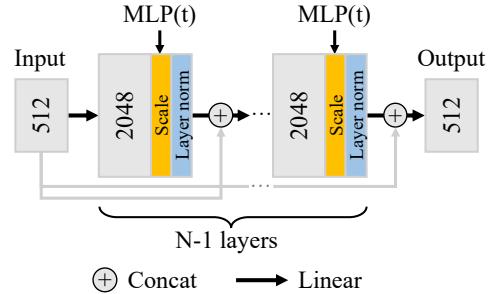


Figure 8. Architecture overview of our latent DDIM.

We have compared different β schedulers including Linear [22], and a constant of 0.008 schedulers. (We found that Cosine [36] scheduler underperformed during preliminary experiments for our latent DDIM.) We compared the two schedulers on the z_{sem} of LSUN’s Horse 128 diffusion autoencoder model. The latent DDIM is MLP + Skip with 10 layers and 2048 hidden nodes. The validation FID score for using linear beta schedule is 13.36, whereas for constant 0.008 scheduler is 10.50. We found that an L1 loss performed better for the latent DDIM with FID of 11.65 vs 13.36 of MSE (Though, the main autoencoder uses MSE loss). We provide the hyperparameter tuning results of the MLP + Skip network:

Latent model	FID
Linear β , 10 layers, size 2048	13.36
Constant 0.008 & L1	
- 10 layers	10.16
- size 3072	9.57
- size 4096	9.43
- 15 layers	9.58
- 20 layers	9.30

Even though these results come from LSUN’s Horse dataset, we found that similar settings worked well across datasets. We only tuned the network depth and the total training iterations for each dataset separately, a common practice in StyleGAN’s training on these datasets.

A.2. Classifiers

We always use linear classifiers (logistic regression) trained on z_{sem} space in all relevant experiments, which are attribute manipulation and class-conditional sampling. For training, z_{sem} is first normalized so that its entire distribution has zero mean and unit variance before putting to the classifier. For the PU classifier, we oversampled the positive data points to match the negative ones to maintain the

Table 5. Network architecture of our diffusion autoencoder based on the improved DPM architecture of Dhariwal et al. [11].

Parameter	CelebA 64	FFHQ 64	FFHQ 128	Horse 128	Bedroom 128	FFHQ256
Batch size	128	128	128	128	128	64
Base channels	64	64	128	128	128	128
Channel multipliers	[1,2,4,8]	[1,2,4,8]	[1,1,2,3,4]	[1,1,2,3,4]	[1,1,2,3,4]	[1,1,2,2,4,4,]
Attention resolution	[16]	[16]	[16]	[16]	[16]	[16]
Images trained	72M	48M	130M	130M	120M	90M
Encoder base ch	64	64	128	128	128	128
Enc. attn. resolution	[16]	[16]	[16]	[16]	[16]	[16]
Encoder ch. mult.	[1,2,4,8,8]	[1,2,4,8,8]	[1,1,2,3,4,4]	[1,1,2,3,4,4]	[1,1,2,3,4,4]	[1,1,2,2,4,4,4]
z_{sem} size	512	512	512	512	512	512
β scheduler	Linear	Linear	Linear	Linear	Linear	Linear
Learning rate			1e-4			
Optimizer			Adam (no weight decay)			
Training T			1000			
Diffusion loss			MSE with noise prediction ϵ			
Diffusion var.			Not important for DDIM			

Table 6. Network architecture of our latent DDIM.

Parameter	CelebA	FFHQ	Horse	Bedroom
Batch size	512	256	2048	2048
z_{sem} trained	300M	100M	2000M	2000M
MLP layers (N)	10	10	20	20
MLP hidden size		2048		
z_{sem} size		512		
β scheduler		Constant 0.008		
Learning rate		1e-4		
Optimizer	AdamW (weight decay = 0.01)	Adam (no weight decay)		
Train Diff T		1000		
Diffusion loss		L1 loss with noise prediction ϵ		
Diffusion var.		Not important for DDIM		

balance. For conditional generation, we follow D2C and apply rejection sampling after an additional thresholding. That is, we reject samples with the target class probabilities less than 0.5 before performing rejection sampling.

B. Computation resources

We used four Nvidia V100s for both diffusion autoencoders and DDIM and a single Nvidia RTX 2080 Ti for the latent DDIMs. Training the latent DDIMs takes only a fraction of the computational resources compared to the diffusion autoencoders. Table 7 shows the throughputs of DDIM and diffusion autoencoders. Diffusion autoencoders were around 20% slower to train than DDIM counterparts due to the additional semantic encoder. The total GPU-hours can be computed by multiplying the throughput with the number of training images for each model provided in Table 6.

Table 7. Throughputs of DDIM and diffusion autoencoders.

Model	DDIMs	Diffusion autoencoders
	Throughput (imgs/sec./V100)	Throughput (imgs/sec./V100)
FFHQ-64	160	128
FFHQ-128	51	41.65
FFHQ-256	-	10.08
Horse-128	51	41.65
Bedroom-128	51	41.65

C. Does the latent DDIM memorize its input?

To verify if our diffusion autoencoder and latent DDIM can generate novel samples and do not simply memorize the input, we generate image samples and compare them to their nearest neighbors in the training set (Figure 9). (They



Figure 9. Does latent DDIM memorize its input? For each sampled image at the top, we find its closest images from the training set in terms of LPIPS, MSE in the image space, and MSE in the semantic subcode \mathbf{z}_{sem} space. The sampled images do not closely resemble any of the training images, suggesting that our latent DDIM does not memorize the input samples.

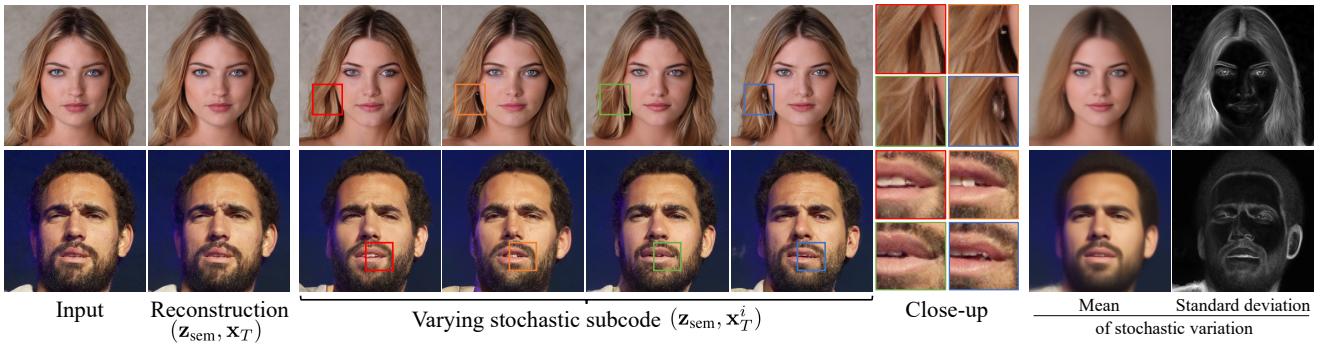


Figure 10. Reconstruction results and the variations induced by changing the stochastic subcode \mathbf{x}_T .

should look different). To find nearest neighbors, we used three different metrics: 1) lowest LPIPS [61] in the image space, 2) lowest MSE in the image space, 3) lowest MSE in the semantic subspace (\mathbf{z}_{sem}). We have found that our autoencoder can generate substantially different images from the training set, suggesting no memorization problem.

D. What is encoded in the stochastic subcode?

Figure 10 shows the stochastic variations induced by varying \mathbf{x}_T given the same \mathbf{z}_{sem} . We also compute the mean and standard deviation of these variations. All generated images look realistic and \mathbf{x}_T changes only minor details, such as the hair pattern, while keeping the overall structure the same.

E. Predictive power of the semantic subcode

We assess the quality of our proposed \mathbf{z}_{sem} via linear classification performance, which has been extensively used to evaluate the quality of learned representations [4, 7, 16, 19]. In Table 8, we measure the performance of linear classifiers trained on \mathbf{z}_{sem} and StyleGAN’s latent code in \mathcal{W} space (obtained from an inversion process [28]) using Area Under the Receiver Operating Characteristic (AUROC) on the CelebA-HQ’s 40 attributes with 30% test data out of 30,000 total data points. The classifiers were trained on z-normalized latent vectors until convergence with Adam optimizer (learning rate 1e-3). For most classes, the linear classifiers using \mathbf{z}_{sem} outperform those using StyleGAN’s \mathcal{W} with weighted averages of 0.92 vs 0.89. This suggests that \mathbf{z}_{sem} contains attribute-specific information that is more readily predictive than that of StyleGAN’s \mathcal{W} .

Table 8. Classification AUROC \uparrow on CelebA-HQ’s 40 attributes of linear classifiers trained on our \mathbf{z}_{sem} vs. StyleGAN’s latent code in \mathcal{W} space (obtained via inversion).

Class	#Positives	\mathbf{z}_{sem}	\mathcal{W}
5_o_Clock_Shadow	1318	0.96	0.94
Arched_Eyebrows	3262	0.88	0.86
Attractive	5183	0.90	0.86
Bags_Under_Eyes	2564	0.89	0.85
Bald	229	0.99	0.99
Bangs	1601	0.98	0.95
Big_Lips	3247	0.73	0.68
Big_Nose	2813	0.88	0.85
Black_Hair	1989	0.96	0.93
Blond_Hair	1546	0.99	0.97
Blurry	34	0.90	0.82
Brown_Hair	2087	0.89	0.81
Bushy_Eyebrows	1682	0.93	0.85
Chubby	622	0.95	0.93
Double_Chin	530	0.95	0.94
Eyeglasses	416	1.00	0.98
Goatee	688	0.98	0.96
Gray_Hair	395	0.98	0.97
Heavy_Makeup	4143	0.97	0.95
High_Cheekbones	4160	0.95	0.91
Male	3273	1.00	1.00
Mouth_Slightly_Open	4195	0.98	0.94
Mustache	502	0.97	0.94
Narrow_Eyes	998	0.86	0.77
No Beard	7335	0.99	0.97
Oval_Face	1872	0.77	0.71
Pale_Skin	434	0.96	0.94
Pointy_Nose	2855	0.74	0.70
Receding_Hairline	777	0.94	0.89
Rosy_Cheeks	1003	0.96	0.92
Sideburns	747	0.99	0.97
Smiling	4175	0.99	0.96
Straight_Hair	1975	0.84	0.77
Wavy_Hair	3197	0.90	0.87
Wearing_Earrings	2310	0.92	0.86
Wearing_Hat	325	0.99	0.96
Wearing_Lipstick	5064	0.98	0.97
Wearing_Necklace	1501	0.79	0.75
Wearing_Necktie	636	0.96	0.95
Young	6978	0.94	0.91
Weighted average		0.92	0.89
Macro average		0.93	0.89

F. Real-image interpolation results

We show interpolation results on real images from FFHQ [27] (Figure 14), LSUN-Bedroom [60] (Figure 15) and

LSUN-Horse [60] (Figure 16). Our method can handle challenging morphing between people with and without glasses, bedrooms from different styles and angles, or horses with different body poses.

To quantify the smoothness of the interpolation, we use Perceptual Path Length (PPL) introduced in StyleGAN [27], to measure the perceptual difference in the image as we move along the interpolation path by a small $\epsilon = 10^{-4}$ in the latent space. Specifically, we compute the following expectation over multiple sampled pairs of latent codes (z_1, z_2) and $t \in [0, 1]$:

$$\text{PPL} = \mathbb{E} \left[\frac{1}{\epsilon^2} d(G(\text{slerp}(z_1, z_2; t)), G(\text{slerp}(z_1, z_2; t + \epsilon))) \right] \quad (10)$$

where G is the decoder, and d computes the perceptual distance based on the VGG16 network. $\text{slerp}(\cdot)$ denotes spherical interpolation. We compute this expected value using 200 samples (400 images) from FFHQ. Our method significantly outperforms DDIM in terms of interpolation smoothness as shown below.

Model	DDIM	Ours
PPL	2,634.14	613.73

G. Real-image attribute manipulation results

We show real-image attribute manipulation results on FFHQ [27] and CelebA-HQ [26] in Figure 5 for smiling, wavy hair, aging, and gender change. For more results, please visit <https://Diff-AE.github.io/>. Our generated results look highly realistic and plausible.

FID between the input and its manipulated version.

To assess the quality of our manipulated results, we compare their distribution with that of real images with the target positive attribute, such as smiling. Our manipulation is done by moving \mathbf{z}_{sem} linearly along the target direction \mathbf{w} , found by training a linear classifier (logistic regression) $\mathbf{y} = \mathbf{w}^\top \mathbf{z} + b$ to predict the target attribute using a labeled dataset. The stochastic subcode \mathbf{x}_T is kept intact. Given \mathbf{z} , its manipulated version is produced by decoding $\mathbf{z}' = \mathbf{z} + s\mathbf{w}$, where $s \in \mathbb{R}$ controls the degree of manipulation. For this experiment, each input image will be manipulated by a different s_i so that the manipulated result reaches the same degree of the target attribute (e.g., similarly big smile) Specifically, we pick s_i so that the logit confidence of its \mathbf{z}'_i equals the median confidence of all real positive images:

$$s_i = \frac{\text{median} - b - \mathbf{z}'_i \top \mathbf{w}}{\mathbf{w}^\top \mathbf{w}} \quad (11)$$

In our implementation, we use normalized \mathbf{z} instead of \mathbf{z}' for this operation and unnormalize it before decoding.

In Table 9, we measure FID scores \downarrow between the manipulated (to be positive) and real positive images, as well as FID scores between real negative and real positive images as baselines for five different attributes from CelebA-HQ [26]. While we expect the manipulated images to get closer to the positive images, we also expect them to not deviate too far from the negative as some original content, such as the background, the identity, should be retained. Hence, we also provide FID scores \downarrow between the manipulated images and the real negative images. Our \mathbf{z}_{sem} manipulated images are closer to the real positive images for 4 out of 5 attributes than those of StyleGAN- \mathcal{W} while better preserving the original contents in all 5 attributes.

Identity preservation. We quantitatively evaluate how well the input’s identity is preserved under the manipulation by computing the cosine similarity \uparrow between the ArcFace embeddings [10] of the input and its manipulated version, following [39]. Table 10 shows our scores on CelebA-HQ images of 4 classes used in Figure 5: Male, Smiling, Wavy Hair, Young. For this experiment, we use the original \mathcal{W} space inversion of StyleGAN that produces the same 512D latent code as our \mathbf{z}_{sem} . Their lower scores can be attributed partly to the poor inversion in this space.

Table 10. Average cosine similarity \uparrow of the ArcFace embeddings [10] of the input and its manipulated version.

Model	Male	Smiling	Wavy Hair	Young
StyleGAN- \mathcal{W}	0.4174	0.7850	0.8544	0.6955
Ours	0.6247	0.8160	0.9821	0.8922

H. Attribute manipulation comparison to D2C

We show a qualitative comparison to D2C [45] on real-image attribute manipulation in Figure 17. These official D2C’s results are from <https://d2c-model.github.io/>. The results of the other baselines are also borrowed from the same website.

I. Class-conditional samples

We show our conditional samples of *Blond*, *Non-blond*, *Female*, and *Male* classes in Figures 18, 19. This is done by training a linear classifier for each attribute using only 100 labeled examples and 10k unlabeled examples, similar to the few-shot experiment done in D2C [45]. The details are in Section 5.6 in the main paper.

J. Unconditional samples

We show uncurated unconditional samples from our diffusion autoencoder on FFHQ [27], LSUN-Bed [60], and LSUN-horse [60] in Figure 20, 21, 22.

K. Encoding out-of-distribution images

As discussed in the main paper, when encoding images that are out of the training distribution, our diffusion autoencoders can still reconstruct the images well but the inferred semantic and stochastic subcodes may fall outside the learned distributions. We simulate simple out-of-distribution samples by translating an FFHQ face image in Figure 11 and by encoding a horse image using our diffusion autoencoder trained on *face* images in Figure 12. The reconstruction results still look very close to the input images, but the noise maps \mathbf{x}_T show some residual details and do not look normally distributed.

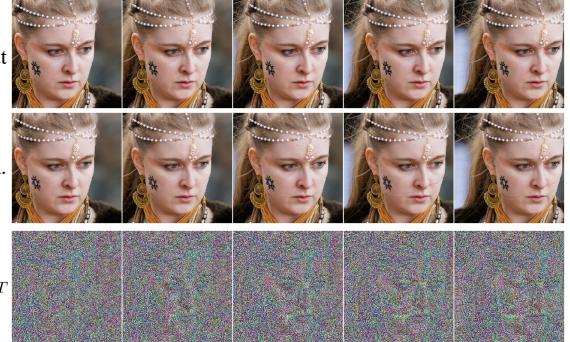


Figure 11. Noise maps \mathbf{x}_T when the input face image is shifted to the right to simulate out-of-distribution input image.

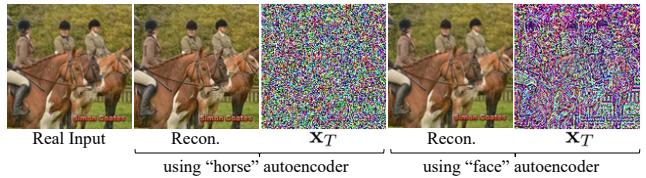


Figure 12. We test how the noise map \mathbf{x}_T of a horse image would look if it is encoded by a diffusion autoencoder trained on face images. Both reconstructions look reasonably close to the input image, but \mathbf{x}_T from the face autoencoder does not look normally distributed and contains details from the input image.

L. Potential negative impact

The ability to generate image samples and manipulate attributes of a real image can be used to generate synthetic media, such as deepfakes. We realize the potential negative impact and further conducted a study to determine the difficulty in distinguishing real and synthesized images from our method, as well as discussing some possible directions.

To detect fake images, we train a CNN architecture based on ResNet-50 [20], which is pretrained on ImageNet [9], followed by a linear layer used for classification. Our training dataset consists of “real” images from FFHQ256 [27]

Table 9. Image manipulation FID scores ↓.

Mode	Model	Male	Smiling	Wavy Hair	Young	Blond Hair
Positive vs negative		95.82	11.15	25.04	36.75	39.65
Manipulated vs. positive	Ours	52.85	9.19	20.80	20.68	33.51
	StyleGAN- \mathcal{W}	42.90	18.52	27.10	31.15	33.89
Manipulated vs. negative	Ours	23.15	7.25	4.89	11.81	6.79
	StyleGAN- \mathcal{W}	66.92	22.15	20.70	31.15	27.54

and “fake” images from either the unconditional sampling experiment (Section 5.7) or the attribute manipulation experiment (Section 5.3). This dataset contains 20k images: 10K images for each real and fake. The dataset is randomly split into train, test, and validation class-balanced subsets with the ratios of 0.7, 0.2, and 0.1, respectively. The classifier is trained using a binary cross-entropy loss function with an SGD optimizer (learning rate 0.001, momentum 0.9, batchsize 64). Fake detection accuracy is reported here:

Method	T=100	T=200	T=500
Unconditional sampling	0.9551	0.9483	0.9313
Attribute manipulation	0.9950	0.9643	0.9213

The results suggest that even though the generated samples look highly realistic, there could be some certain artifacts that can be easily detected by another neural network. Diffusion-based models also do not have a mechanism to purposely fool a classifier or discriminator like GANs do, and a neural network-based technique is currently found to be > 90% effective at detecting fake images from diffusion models. Note that sampling with higher T leads to samples that are harder to detect. A further study on how easy it is to circumvent detection through adversarial training and an analysis on those giveaway artifacts will be useful for future technical safeguards.

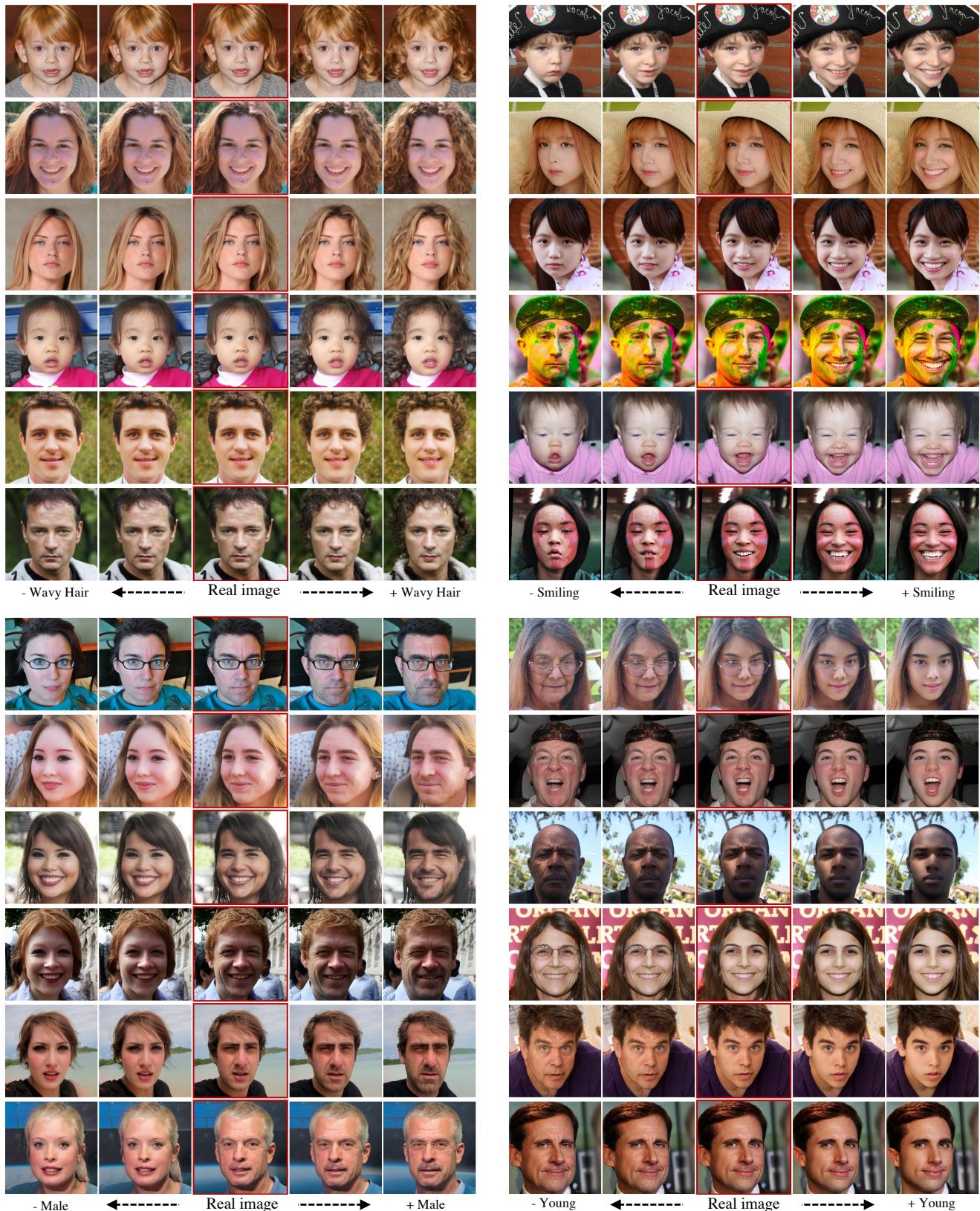


Figure 13. Real-image attribute manipulation for attributes: *Wavy Hair, Smiling, Male, Young*.



Figure 14. Real-image interpolation on FFHQ dataset [27]

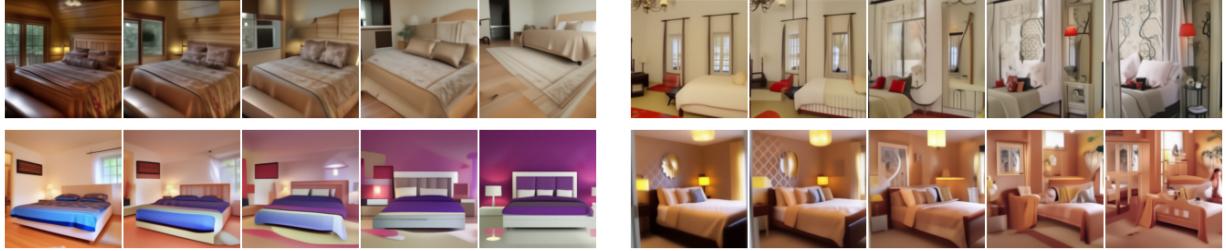


Figure 15. Real-image interpolation on LSUN bedroom-128 [60]



Figure 16. Real-image interpolation on LSUN horse-128 [60]

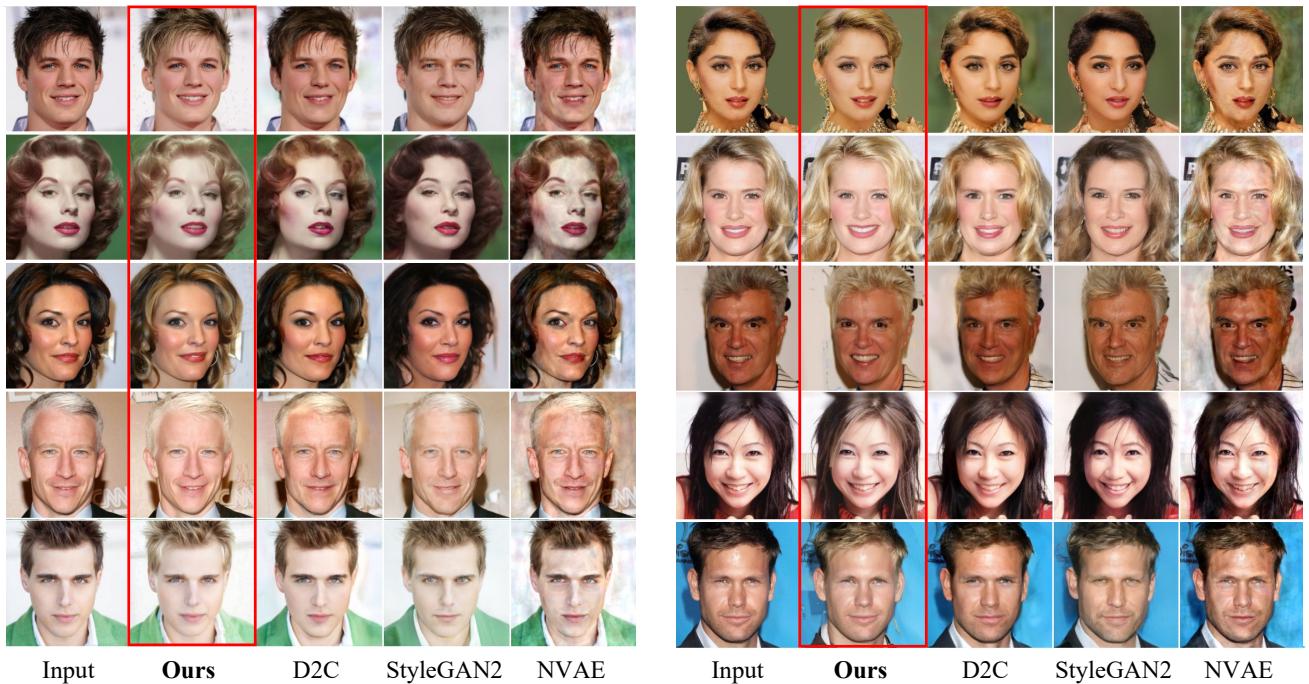


Figure 17. Comparison on attribute manipulation (blond hair) between our method, D2C [45], StyleGAN2 [28], and NVAE [50].

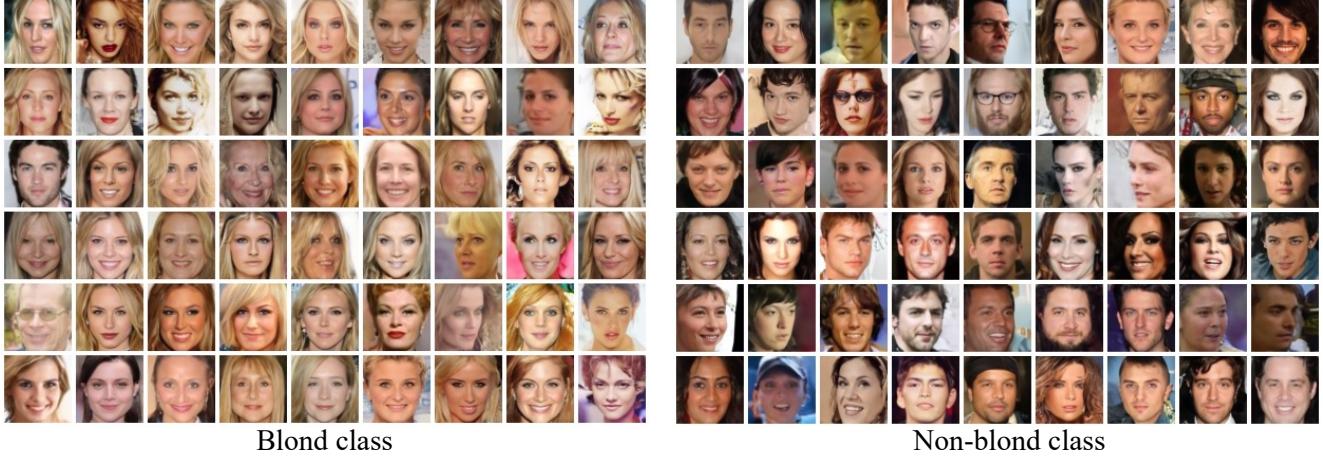


Figure 18. Class-conditional generation using 100 positive labeled examples and 10k unlabeled examples on *Blond* and *Non-blond* from CelebA [26]. These results are uncurated. Please see Section 5.6 in the main paper for details.

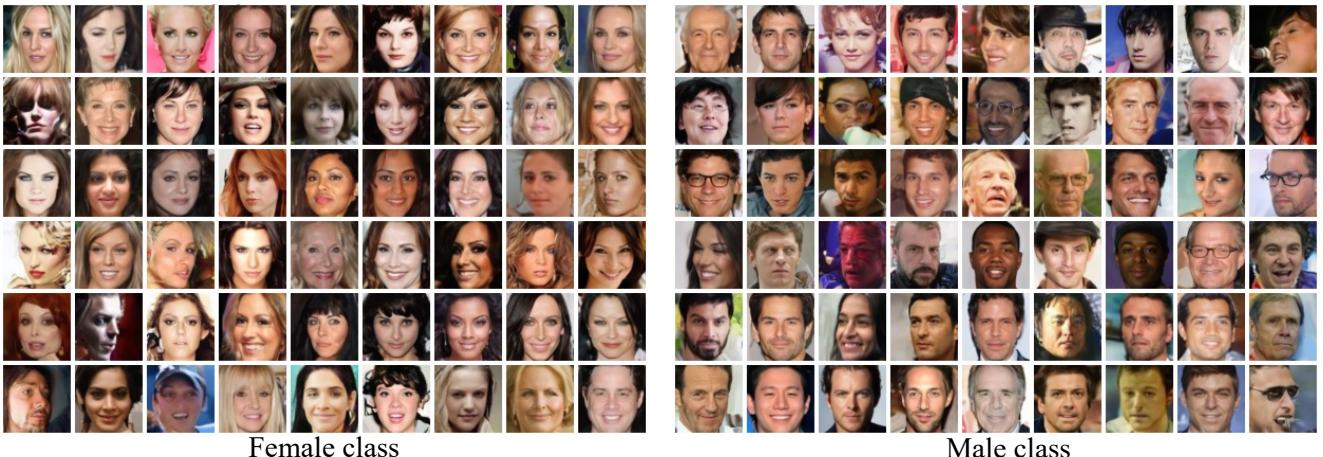


Figure 19. Class-conditional generation using 100 positive labeled examples and 10k unlabeled examples on *Female* and *Male* from CelebA [26]. These results are uncurated. Please see Section 5.6 in the main paper for details.

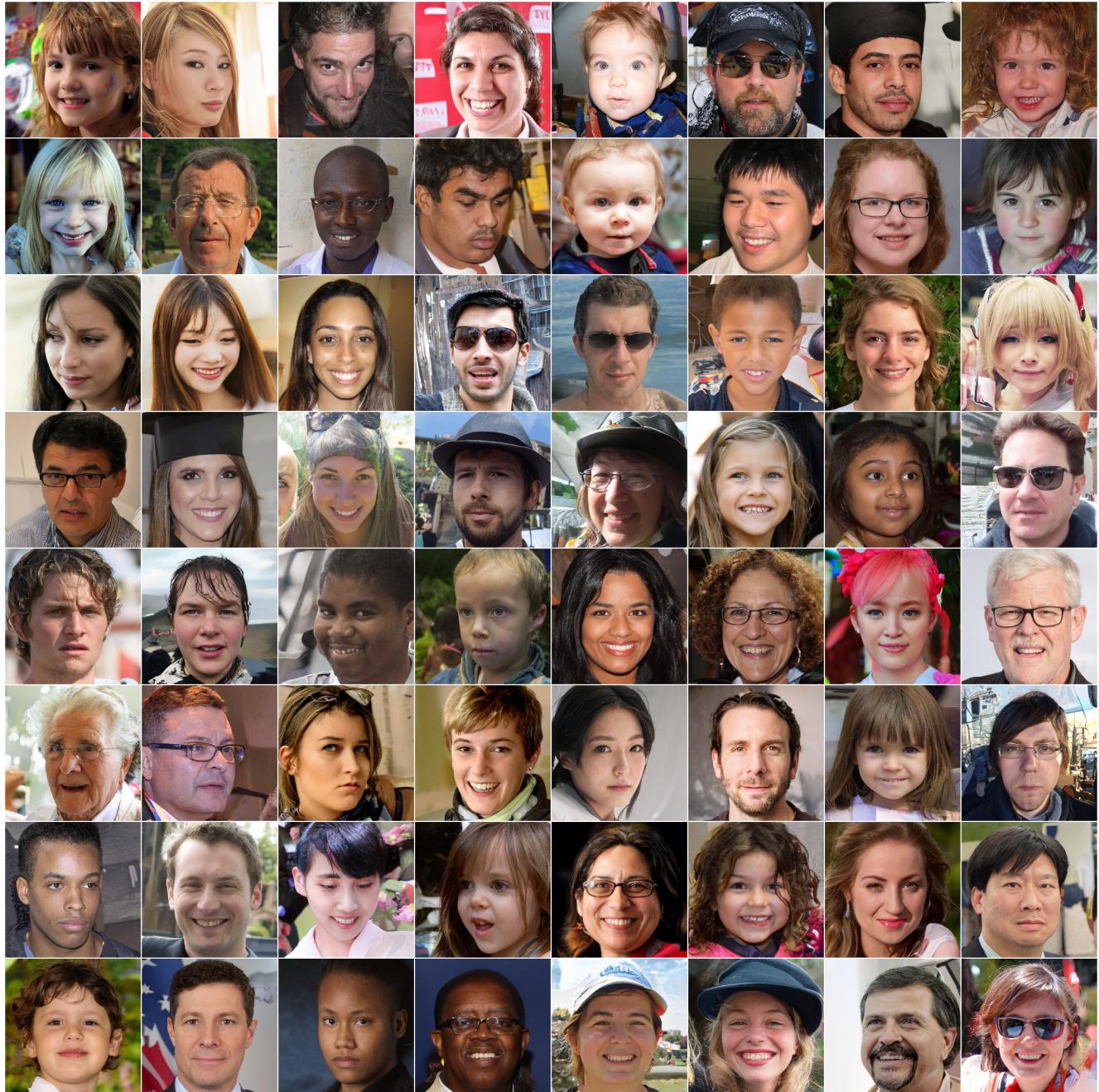


Figure 20. Unconditional samples (uncurated) from our diffusion autoencoder and latent DDIM trained on FFHQ-256 [27].

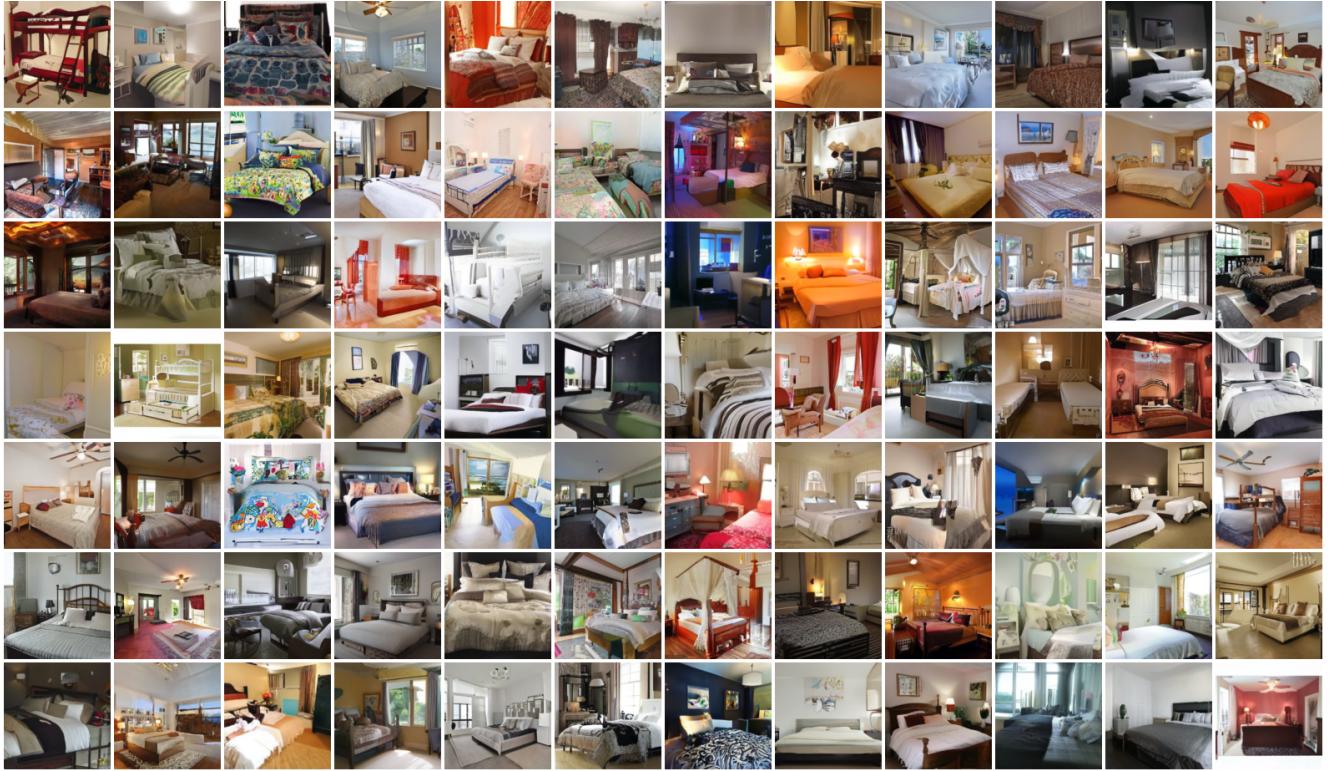


Figure 21. Unconditional samples (uncurated) from our diffusion autoencoder and latent DDIM trained on LSUN bedroom-128 [60].

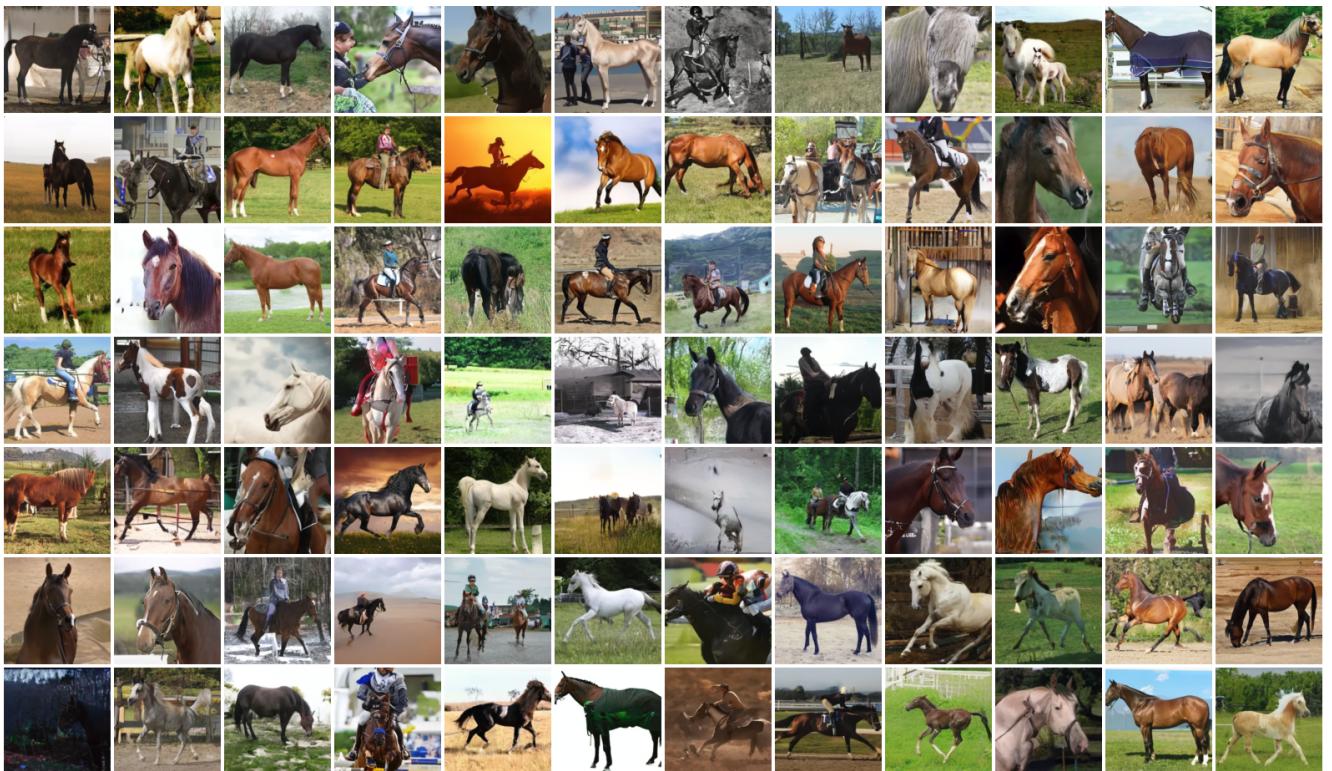


Figure 22. Unconditional samples (uncurated) from our diffusion autoencoder and latent DDIM trained on LSUN horse-128 [60].