Controllable and Compositional Generation with Latent-Space Energy-Based Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Controllable generation is one of the key requirements for successful adoption of deep generative models in real-world applications, but it still remains challenging. In particular, the compositional ability to generate novel concept combinations is out of reach for most current models. In this work, we use energy-based models (EBMs) to handle compositional generation over a set of attributes. To make them scalable to high-resolution image generation, we introduce an EBM in the latent space of a pre-trained generative model such as StyleGANs. We propose a novel EBM formulation representing the joint distribution of data and attributes together, and we show how sampling from it is formulated as solving an ordinary differential equation (ODE). Given a pre-trained generator, all we need for controllable generation is to train an attribute classifier. Sampling with ODEs is done efficiently in the latent space and is robust to hyperparameters. Thus, our method is simple, fast to train, and efficient to sample. Experimental results show that our method outperforms the state-of-the-art in both conditional sampling and sequential editing. In compositional generation, our method excels at zero-shot generation of unseen attribute combinations. Also, by composing energy functions with logical operators, this work is the first to achieve strong compositionality in generating photo-realistic images of resolution 1024×1024.

1 Introduction

Deep generative learning has made tremendous progress in image synthesis. High-quality and photorealistic images can now be generated with generative adversarial networks (GANs) [11, 21, 22], score-based models [16, 38], and variational autoencoders (VAEs) [25, 40]. However, a key requirement for the success of these generative models in many real-world applications is the controllability of the generation process. Controllable generation, in particular the compositional ability to generate novel concept combinations, still remains a great challenge to current methods.

A common approach for achieving controllable generation is to train a *conditional* model [29, 24, 7, 32, 30], where the conditional information, such as semantic attributes, is specified during training to guide the generation. There are two major drawbacks with this approach: (i) Since the set of attributes is fixed and pre-defined, it is difficult to introduce new attributes to an existing model. This, in turn, limits the model's compositional ability for unseen attributes and their novel combinations [8]. (ii) Training conditional generative models from scratch for new attributes is computationally expensive, and it can be challenging to reach to the same image quality of the uncontrollable counterparts [7, 32].

Recent works have attempted to overcome these issues by first training an unconditional generator, and then converting it to a conditional model with a small cost [19, 13, 1, 33]. This is often achieved by discovering semantically meaningful directions in the latent space of the unconditional model. This way, one would pay the most computational cost only once for training the unconditional model. However, these approaches often struggle with compositional generation, in particular with rare

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

combination of the attributes and the introduction of new attributes. An appealing solution to the compositionality problem is to use energy-based models (EBMs) for controllable generation [8, 12, 9]. This is due to the fact that energy functions representing different semantics can be combined together to form compositional image generators. However, existing approaches train EBMs directly in the pixel space, making them slow to sample and difficult to generate high-resolution images [9].

In this paper, we leverage the compositionality of EBMs and the generative power of the state-of-the-art pre-trained models such as StyleGANs [21, 22] to achieve high-quality controllable and compositional generation. **Our main contributions are summarized as follows**:

- We propose a novel formulation of a joint EBM in the latent space of pre-trained generative models for controllable generation. In particular, the EBM training reduces to training a classifier only, making it conceptually simple, and it can be applied to any latent-variable generative model.
- Based on our EBM formulation, we propose a new sampling method that only relies on an ordinary different equation (ODE) solver in the latent space. The sample quality is robust to hyperparameters.
- We show that our method is fast to train and efficient to sample, and it outperforms previous methods in both conditional sampling and sequential editing.
- Moreover, we show strong compositionality with zero-shot generation on unseen attribute combinaions, and with logical compositions of energy functions in generating high-quality images.

Specifically, we build an EBM in the joint space of data and attributes where the marginal data distribution is denoted by an implicit distribution (e.g., a pre-trained GAN generator), and the conditional distribution of attributes, given data, is represented by an attribute classifier. Using the reparameterization trick, our EBM formulation induces a joint energy function in the latent space where the latent distribution is known (a standard Gaussian distribution). This way, we only need to train the classifier in the data space and do the sampling in the latent space. Because our method only requires training a classifier to add controllability, it is conceptually simple and fast-to-train.

For sampling, most existing EBMs rely on the Langevin dynamics (LD) sampling which is often computationally expensive and sensitive to the choice of sampling parameters. Instead, we build on the recent score-based model [38], and define the sampling process in the latent space using the corresponding probability flow ODEs, induced by the reverse diffusion process. Thus, we rely on an ODE solver for sampling from our EBM formulation in the latent space. With its adaptive step size, the ODE sampling in the latent space is efficient and robust to the sampling parameters.

In experiments, we show our method achieves higher image quality and better controllability in both the conditional sampling and sequential editing, compared to various baselines, including StyleFlow [1] and JEM [12]. For the training time, we show our method is $25\times$ faster than StyleFlow on the FFHQ data. For the inference time, our sampling is at least $49\times$ and $876\times$ faster than EBMs [12, 8] and score-based models [38] in the pixel space, respectively, on CIFAR-10. More importantly, our method excels on zero-shot generation with unseen attribute combinations where StyleFlow almost completely fails. By composing energy functions with logical operators [8], our method is the first to show such strong compositionality in generating photo-realistic images of resolution 1024×1024.

2 Method

In this section, we will first give an overview of EBMs and then propose our new EBM formulation in the latent space, based on which, we introduce a new sampling method through the ODE solver.

2.1 Energy-based models

EBMs [27] represent data $x \in \mathbb{R}^d$ by learning an unnormalized probability distribution $p_\theta(x) \propto e^{-E_\theta(x)}$, where $E_\theta(x) : \mathbb{R}^d \to \mathbb{R}$ is the energy function, parameterized by a neural network. To train an EBM on a data distribution p_{data} with maximum likelihood estimation (MLE), we can estimate the gradient of the data log-likelihood $L(\theta) = \mathbb{E}_{x\sim p_{\text{data}}}[\log p_\theta(x)]$ as [15]:

$$\nabla L(\theta) = \mathbb{E}_{x\sim p_{\text{data}}}[\nabla_\theta E_\theta(x)] - \mathbb{E}_{x\sim p_\theta}[\nabla_\theta E_\theta(x)] \tag{1}$$

To sample x from p_θ for training and inference, Langevin dynamics (LD) [41] is applied as follows,

$$x_0 \sim p_0(x), \quad x_{t+1} = x_t - \frac{\eta}{2}\nabla_x E_\theta(x_t) + \omega_t, \quad \omega_t \sim N(0, \eta I) \tag{2}$$

where p_0 is the initial distribution, η is the step size. When $\eta \to 0$ and $t \to \infty$, x_t is guaranteed to follow the probability $p_\theta(x)$ under some regularity conditions [37].

2.2 Modelling joint EBM in the latent space

To formulate our EBM for controllable generation, we consider the following setting: We have the data $x \in \mathcal{X} \subseteq \mathbb{R}^d$ and its attribute code $c = \{c_1, \cdots, c_n\} \in \mathcal{C} \subseteq \mathbb{R}^n$, which is a n-dimensional vector with each entry c_i representing either a discrete m_i-class attribute where $c_i \in \{0, \cdots, m_i - 1\}$, or a continuous attribute where $c_i \in \mathbb{R}$. The goal is to learn a generative model and semantically control its generated samples by manipulating its conditioning attribute code c. To begin with, we define the generative model as follows,

$$p_\theta(x, c) := p_g(x) p_\theta(c|x) \propto p_g(x) e^{-E_\theta(c|x)} \tag{3}$$

where $p_g(x)$ is an implicit distribution defined by a pre-trained generator g (such as GANs) in the form $x = g(z)$ with the latent variable $z \in \mathcal{Z}$. And, $p_\theta(c|x)$ is conditional distribution on attributes given x, modeled by a conditional energy function $E_\theta(c|x)$.

In the following, we will assume that the generator g is fixed and we only train the energy function $E_\theta(c|x)$. We also assume the attributes are conditionally independent, i.e., $E_\theta(c|x) = \sum_{i=1}^n E_\theta(c_i|x)$. Here, we aim at preserving the image generation quality, and we design the joint in Eq. (3) such that the marginal data distribution $p_\theta(x)$ from our joint distribution satisfies $p_\theta(x) = p_g(x)$. This is obtained with an energy function that is normalized up to a constant, i.e., $\sum_{c_i} e^{-E_\theta(c_i|x)} = \text{const}$ for discrete attributes. Thus, we define:

$$E_\theta(c_i|x) = \begin{cases} -f_i(x;\theta)[c_i] + \log \sum_{c_i} \exp\left(f_i(x;\theta)[c_i]\right) & \text{if } c_i \text{ is discrete,} \\ \frac{1}{2\sigma^2}(c_i - f_i(x;\theta))^2 & \text{if } c_i \text{ is continuous} \end{cases} \tag{4}$$

where $f_i(x;\theta)$ is the output of a multi-class classifier mapping from $\mathcal{X}$ to $\mathbb{R}^{m_i}$ if the i-th attribute is discrete or a regression network mapping from $\mathcal{X}$ to $\mathbb{R}$ if it is continuous. Without loss of generality, we will always call $f_i(x;\theta)$ a classifier. Here, σ^2 is a hyperparameter to adjust the continuous energy. In practice, we set $\sigma^2 = 0.01$ after normalizing all the continuous attributes to $[0, 1]$.

We then use the reparameterization trick [4, 42] for sampling x from $p_\theta(x, c)$. In particular, sampling x from $p_\theta(x, c)$ is reparameterized to sampling z from

$$p_\theta(z, c) \propto e^{-E_\theta(c|g(z)) + \log p(z)} \tag{5}$$

and then transferring the z samples to the data space by applying the generator $x = g(z)$. See Appendix A.1 for the derivation of Eq. (5). In most generative models, the prior distribution $p(z)$ is a standard Gaussian. Thus, Eq. (5) becomes $p_\theta(z, c) \propto e^{-E_\theta(z,c)}$ with the joint energy function:

$$E_\theta(z, c) = \sum_{i=1}^n E_\theta(c_i|g(z)) + \frac{1}{2}\|z\|_2^2 \tag{6}$$

where each conditional energy function $E_\theta(c_i|g(z))$ is given by Eq. (4). For sampling, we can run LD of Eq. (2) in the z-space using the energy function above. At the end of a chain, we pass the final z samples to the generator g to get the x samples. Since our latent-space EBM is formulated for controllable generation, we term it as **LA**tent **C**ontrollable **E**BM (LACE) throughout this work.

Inspecting the joint energy $E_\theta(z, c)$ in Eq. (6) shows that when the unconditional generator g is fixed, the only trainable component is the classifier $f_i(x;\theta)$ that represents $E_\theta(c_i|x)$. Therefore, unlike the previous joint EBMs [12] that train both $p_\theta(x)$ and $p_\theta(c|x)$, our method only needs to train a classifier for the conditional $p_\theta(c|x)$, and does not require sampling with LD in the x-space during training. That is, for controllable generation, we can train the classifier only in the x-space and do the LD sampling in the z-space. This fact makes our method conceptually simple and easy to train.

Note that the considered x-space for training the classifier is not necessarily the pixel space. Instead, we can choose any intermediate layer of the pre-trained generator as the x-space. Take StyleGAN2 [22] as an example, if we train the classifier in the w-space, then $g(z)$ in Eq. (6) actually corresponds to the mapping network of StyleGAN2, and once we obtain the w samples via the LD sampler, we can pass them to the synthesis network of StyleGAN2 to get the final images. In other words, since data x is a deterministic function of z, the distribution of attributes is uniquely defined by z. We have the freedom to define $p_\theta(c_i|z)$ by applying the classifier on top of z or any representation extracted from z including the generator $g(z)$ or its intermediate representations such w-space in StyleGAN2.

2.3 Sampling through an ODE solver

Sampling with LD requires hand-crafted tricks to speed up and stabilize its convergence [10, 12]. Even in the latent space, our experiments show that LD tends to be sensitive to hyperparameters. In

this section, we introduce another sampling method, called *ODE sampler*, that relies on an ODE solver. Our method with the LD and ODE sampler is termed as *LACE-LD* and *LACE-ODE*, respectively.

Our idea is inspired by the prior work [38], which shows that controllable generation can be achieved by solving a conditional reverse-time SDE. Specifically, if we consider a *Variance Preserving (VP) SDE* [38], the forward SDE is defined as $dx = -\frac{1}{2}\beta(t)xdt + \sqrt{\beta(t)}dw$, where $\beta(t)$ represents a scalar time-variant diffusion coefficient and w is a standard Wiener process. Then, the conditional sampling from $p_0(x|c)$ is equivalent to solving the following reverse SDE:

$$dx = -\frac{1}{2}\beta(t)[x + \nabla_x \log p_t(x, c)]dt + \sqrt{\beta(t)}d\bar{w} \tag{7}$$

where $\bar{w}$ is a standard Wiener process when time flows backwards from T to 0. Accordingly, there exists a deterministic ODE whose trajectories share the same joint probability densities $p_t(x(t), c)$:

$$dx = -\frac{1}{2}\beta(t)\left[x + \nabla_x \log p_t(x, c)\right]dt \tag{8}$$

Song et al. [38] define $\nabla_x \log p_t(x(t), c) := \nabla_x \log p_t(x(t)) + \nabla_x \log p_t(c|x(t))$ which requires (i) estimating the score function $\nabla_x \log p_t(x(t))$, and (ii) training a *time-variant* classifier $p_t(c|x(t))$ in the x-space for $\nabla_x \log p_t(c|x(t))$, both of which make the training and inference challenging. Instead, in our framework, we solve the ODE in the z-space and transfer the z samples to the data space with the generator g. Specifically with $p_t(z(t), c) \propto e^{-E_t(c|g(z(t)))+\log p_t(z(t))}$, we have

$$dz = -\frac{1}{2}\beta(t)\left[z + \nabla_z \log p_t(z) - \nabla_z E_t(c|g(z))\right]dt \tag{9}$$

Since the distribution of latent variable z in most generative models satisfies $p_0(z(0)) = \mathcal{N}(0, I)$, diffusing it with VP-SDE will not change its distribution at time t, i.e., $p_t(z(t)) = \mathcal{N}(0, I)$ [38]. Since $p_t(z(t))$ is time-invariant and the generator g is fixed, the classifier $p_t(c|g(z(t)))$ receives input $g(z(t))$ with a time-invariant distribution. Thus, the classifier can be assumed to be time-invariant, and so is the energy function $E_t(c|g(z(t))) := E(c|g(z(t)))$. Therefore, the above ODE becomes:

$$dz = \frac{1}{2}\beta(t)\nabla_z E(c|g(z))dt = \frac{1}{2}\beta(t)\sum_{i=1}^{n}\nabla_z E(c_i|g(z))dt \tag{10}$$

Similarly, each energy function $E(c_i|g(z))$ is given by Eq. (4). In this case, we do not need to learn $\nabla_x \log p_t(z(t))$ as it is given from $p_t(z(t)) = N(0, I)$. Moreover, the controllable generation through the ODE sampler is achieved by training the *time-invariant* classifier only.

3 Experiments

In this section, we show the effectiveness of our method in conditional sampling, sequential editing and compositional generation, and we also perform an ablation study on the sampling method.

Experimental setting We use StyleGAN2-ADA [20] as the pre-trained model for experiments on CIFAR-10 [26], and StyleGAN2 [22] for experiments on FFHQ [21]. We train the classifier in the w-space, where our method works best (see ablation studies to Appendix A.4). For LACE-ODE, we use the 'dopri5' solver [5] with the tolerances of (1e-3, 1e-3). For LACE-LD, the step size η and the standard deviation σ of ω_t are chosen separately for faster training [12], where the number of steps $N = 100$, step size $\eta = 0.01$ and standard deviation $\sigma = 0.01$. For metrics, we use (i) *conditional accuracy* (*ACC*) to measure the controllability, where we generate images using randomly sampled attribute codes, and pass them to a pre-trained image classifier to predict the attribute codes, and (ii) *FID* to measure the generation quality and diversity [14]. See Appendix A.2 for more details.

3.1 Conditional sampling

Baselines For comparison, we consider a set of baselines: StyleFlow [1], JEM [12], Conditional EBM (Cond-EBM) [8], and SDEs [38]. We also propose *Latent-JEM*, a variant of JEM modelled in the latent space, and *LACE-PC*, which replaces the ODE sampler with the Predictor-Corrector (PC) sampler that solves the reverse SDE (Eq. 7) [38]. See the Appendix A.3 for more details.

CIFAR-10 The results of our method against baselines on CIFAR-10 are shown in Table 1 and Figure 1. Our method requires only 4 minutes to train, and it takes less than one second to sample a batch of 64 images on a single NVIDIA V100 GPU, which significantly outperforms previous EBMs (at least 49× faster) and score-based models (at least 876× faster) in the pixel space. Latent-JEM that works in the latent space is also slower in training as it performs the LD for each parameter update.

Table 1: Comparison of our method and baselines for conditional sampling on CIFAR-10. For notations, Train – training time, Infer – inference time (m: minute, s: second), which refer to the single GPU time for generating a batch of 64 images, η is the LD step size, and N is the number of predictor steps in the PC sampler.

Methods	Train	Infer	FID↓	ACC↑
JEM [12]	2160m	135s	52.35	0.645
Cond-EBM [8]	2280m	24.5s	41.72	0.792
VP-SDE [38]	52800m	438s	19.13	0.643
VE-SDE [38]	52800m	448s	**2.97**	0.662
Latent-JEM (η=0.1)	21m	0.63s	8.75	0.950
Latent-JEM (η=0.01)	21m	0.63s	5.65	0.821
LACE-PC (N=100)	**4m**	0.84s	**2.99**	0.747
LACE-PC (N=200)	**4m**	1.86s	**2.94**	0.722
LACE-LD	**4m**	0.68s	4.30	0.939
LACE-ODE	**4m**	**0.50s**	6.63	**0.972**

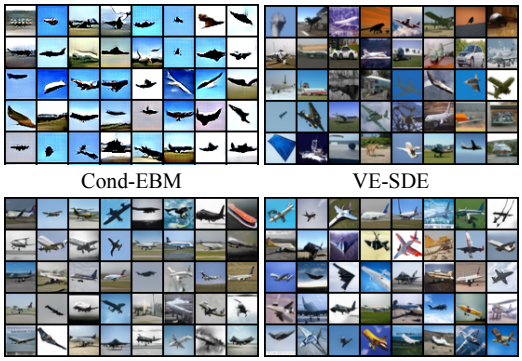


Figure 1: Conditionally generated images of our method (LACE-ODE) and baselines on the plane class of CIFAR-10. See Appendix A.5 for more results.

Table 2: Comparison between our method and baselines for conditional sampling on the `glasses` and `gender_smile_age` of FFHQ, respectively. For notations, Train – training time, Infer – inference time (m: minute, s: second), which refer to the single GPU time for generating a batch of 16 images, η is the LD step size, and N is the number of predictor steps in the PC sampler.

Methods	Train	`glasses`			`gender_smile_age`				
		Infer	FID↓	ACC_{gl}↑	Infer	FID↓	ACC_{ge}↑	ACC_{s}↑	ACC_{a}↑
StyleFlow [1]	50m	**0.61s**	42.08	0.899	**0.61s**	43.88	0.718	0.870	0.874
Latent-JEM (η=0.1)	15m	0.69s	22.83	0.765	0.93s	22.74	0.878	0.953	0.843
Latent-JEM (η=0.01)	15m	0.69s	21.58	0.750	0.93s	**21.98**	0.755	0.946	0.831
LACE-PC (N=100)	**2m**	1.29s	21.48	0.943	2.65s	24.31	0.951	0.922	0.896
LACE-PC (N=200)	**2m**	2.20s	21.38	0.925	4.62s	23.86	0.949	0.914	0.894
LACE-LD	**2m**	1.15s	**20.92**	**0.998**	2.40s	22.97	0.955	0.960	**0.913**
LACE-ODE	**2m**	0.68s	**20.93**	**0.998**	4.81s	24.52	**0.969**	**0.982**	**0.914**

For the conditional sampling performance, LACE-ODE and LACE-LD largely outperform all the baselines in precisely controlling the generation while maintaining the relatively high image quality. In particular, LACE-PC performs similarly to VE-SDE: they can achieve better image quality but have a problem with precisely controlling the generation (ACC$\leq$0.75). Besides, we observe that EBMs in the latent space achieves much better overall performance than EBMs in the pixel space.

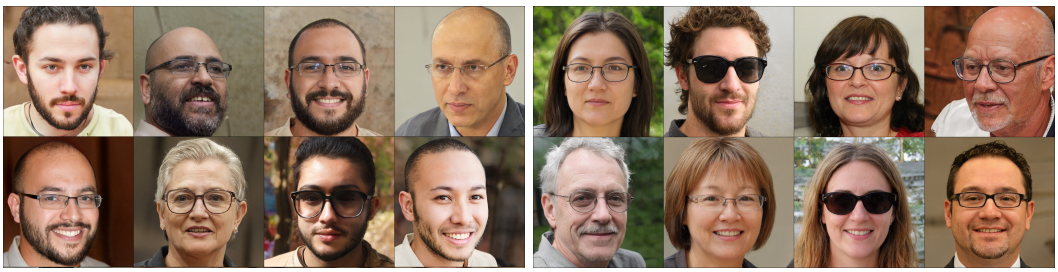
FFHQ To test on FFHQ, we use 10k (w, c) pairs created by [1] for training, where w is sampled from the w-space of StyleGAN2, and c is the attribute code. Unless otherwise specified, we use truncation $\psi = 0.7$ for our method. Following the evaluation protocol from [1], we use 1k generated samples from StyleGAN2 to compute the FID. Note that given the small sample size, FID values tend to be high. For the reference, the original unconditional StyleGAN2 with 1k samples has FID=20.87.

The results of comparing our method with baselines are shown in Table 2 and Figure 2, where we condition on `glasses` and `gender_smile_age`, respectively. For the training time, our method only takes 2 minutes, which is around $25\times$ faster than StyleFlow, and $5\times$ faster than Latent-JEM. Our inference time increases with the number of attributes to control. For instance, LACE-ODE needs similar inference time with StyleFlow (0.68s vs. 0.61s) on `glasses`, but more inference time than StyleFlow (4.81s vs. 0.61s) on `gender_smile_age`. In practice, we could adjust the tolerances of the ODE solver, allowing for trade-offs between the inference time and overall performance.

For the controllability, both LACE-ODE and LACE-LD outperform the baselines by a large margin. Also, the generation quality of our method is on par with our proposed baselines, and much better than the prior work StyleFlow. For instance, LACE-ODE has much lower FID than StyleFlow (24.52 vs. 43.55). Latent-JEM achieves better FID but always has worse controllability. These quantitative results could be verified by the visual samples in Figure 2, where our method achieves high-quality controllable generation. However, StyleFlow has difficulty with (i) the full controllability in different cases, and (ii) the lack of image diversity specifically when conditioning on more attributes.

3.2 Sequential editing

In sequential editing, we semantically edit the images by changing an attribute each time without affecting other attributes and face identity. Given a sequence of attributes $\{c_1, \cdots, c_n\}$, we adapt our



(a) `glasses=1` (*Left*: StyleFlow, *Right*: Ours)

(b) `gender=female,smile=1,age=55` (*Left*: StyleFlow, *Right*: Ours)

Figure 2: Conditionally generated images of our method (LACE-ODE) and StyleFlow [1] on the `glasses` and `gender_smile_age` of FFHQ, respectively. Visually, our method achieves higher image generation quality, more diversity and better controllability than StyleFlow.

method for sequential editing by relying on the compositionality of energy functions. We define the joint energy function of the i-th edit as

$$E_\theta^{\text{seq}}(z, \{c_j\}_{j=1}^i) := E_\theta(z, \{c_j\}_{j=1}^i) + \mu \tilde{d}(z, z_{i-1}) + \gamma \sum\nolimits_{j>i} d\left(f_j(g(z);\theta), f_j(g(z_{i-1});\theta)\right)$$

where the joint energy function $E_\theta(z, \{c_j\}_{j=1}^i)$ is from Eq. (6), and $f_j(\cdot;\theta)$ is the classifier output for the j-th attribute, $\tilde{d}(z, z_{i-1}) = \|g(z) - g(z_{i-1})\|_2^2 + \|z - z_{i-1})\|_2^2$ prevents z from moving too far from the previous z_{i-1}, and the last term penalizes z for changing other attributes, with $d(\cdot,\cdot)$ defined as the squared L2 norm. Note that in the i-th edit, we use z_{i-1} as the new initial point of the sampling for a faster convergence. By default, we set $\mu = 0.04$ and $\gamma = 0.01$.

We compare our method against StyleFlow [1], the state-of-the-art in sequential editing, with two additional metrics that quantify the disentanglement of the edits: (i) the *face identity loss* (*ID*) [1, 33], which calculates the distance between the image embeddings before and after editing to measure the identity preservation. (ii) the *disentangled edit strength* (*DES*), defined as $\text{DES} = \frac{1}{n}\sum_{i=1}^n \text{DES}_i$ and $\text{DES}_i = \mathbb{E}_{p_\theta(x)}[\Delta_i - \max_{j\neq i} |\Delta_j|]$, where $\Delta_i = \frac{\text{ACC}_i - \text{ACC}_{0i}}{1-\text{ACC}_{0i}}$ denotes the *normalized* ACC improvement, with ACC_i and ACC_{0i} being the ACC score of i-th attribute after and before the i-th edit, respectively. Intuitively, the maximum DES is achieved when each edit precisely control the considered attribute only but leave other attributes unchanged, leading to good disentanglement.

Table 3 shows the quantitative results, where we use the truncation $\psi = 0.5$. Our method outperforms StyleFlow in terms of disentanglement (DES), image quality (FID) and controllability (ACC) while achieving competitive identity preservation quality (ID). This is confirmed by the qualitative results in Figure 3, where StyleFlow usually has the following issues: (i) *changing unedited attributes*, such as accidentally adding or losing glasses, changing the hair color, and so on, and (ii) *having incomplete edits*, for example, the smiling face still appears after setting `smile=0` (last row in Figure 3). On the contrary, our method suffers less from the above problems. Moreover, unlike StyleFlow that applies hand-crafted tricks to control the edit, such as editing specific subset selection and edit reprojection [1], our method is simple as it is defined by composing different energy functions.

3.3 Compositional generation

Zero-shot generation The goal is to generate novel images conditioned on *unseen combinations* of attributes that are not present in the training data, which is used for evaluating a model's compositionality in controllable generation. We compare our method against StyleFlow on zero-shot generation in Figure 4, where we condition on (a) {`beard=1, smile=0, glasses=1, age=15`} and (b) {`gender=female, smile=0, glasses=1, age=10`}, respectively. We can see our method still

Table 3: Comparison between our method and StyleFlow [1] for sequential editing on FFHQ, where we edit each attribute in the sequence of [`yaw`, `smile`, `gender`, `age`, `glasses`]. Note that here we only show the final performance *after all edits*, and the results of every individual edit are deferred to Appendix A.7.

Methods	All (`yaw_smile_gender_age_glasses`)							
	DES↑	ID↓	FID↓	ACC_{y}↑	ACC_{s}↑	ACC_{ge}↑	ACC_{a}↑	ACC_{gl}↑
StyleFlow [1]	0.544	**0.607**	43.81	**0.947**	0.778	0.731	0.815	0.872
LACE-ODE	**0.638**	**0.608**	**28.83**	0.938	**0.967**	**0.921**	**0.895**	**0.974**

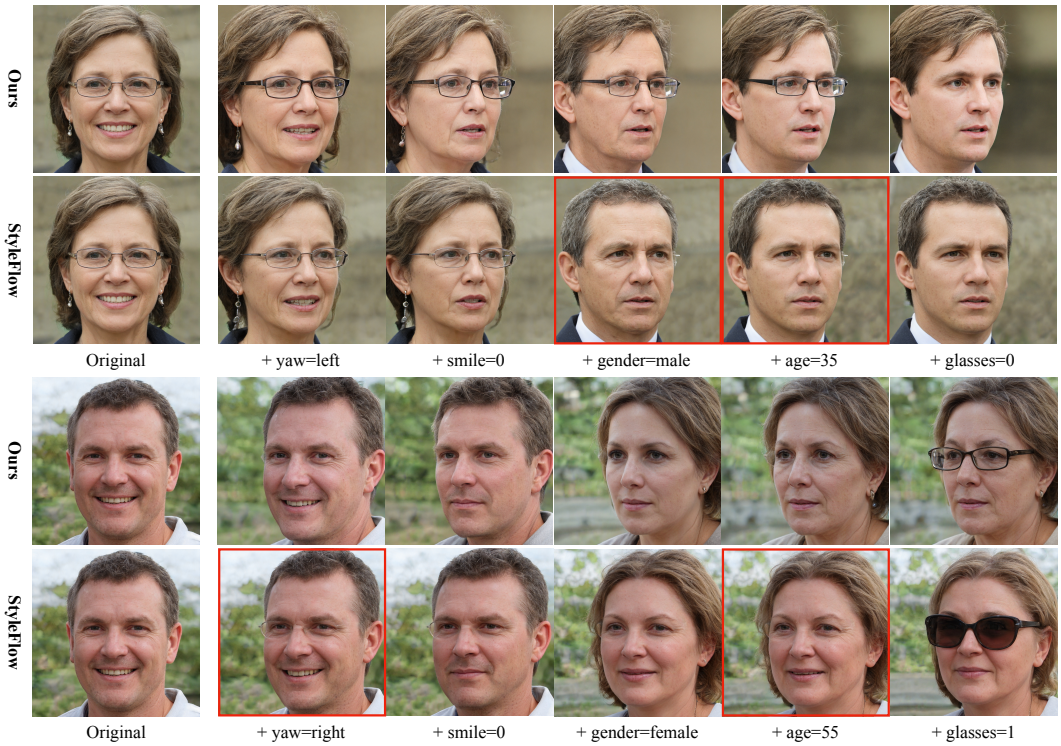


Figure 3: Sequentially editing images with our method (LACE-ODE) and StyleFlow [1] on FFHQ with a sequence of [`yaw`, `smile`, `gender`, `age`, `glasses`]. Our method can successfully perform each edit while less affecting the other attributes. On the contrary, StyleFlow may unintentionally add/lose glasses, change the hair color, or have a smiling face when it is not supposed to.

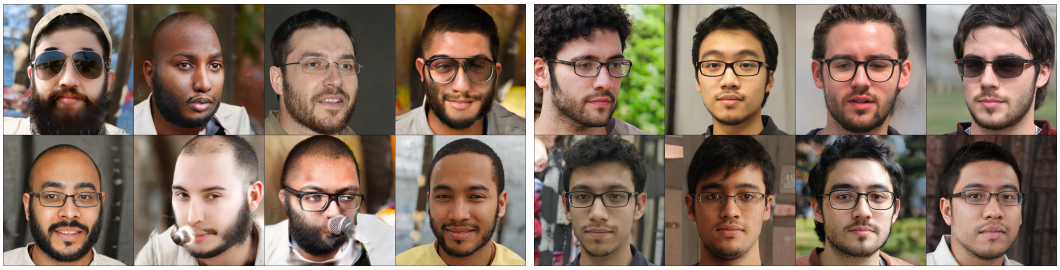
performs well in zero-shot generation while StyleFlow suffers from a severe deterioration of image quality and diversity, and almost completely fails in the controllability. Quantitatively, the ACC scores of our method are much larger than StyleFlow, as we have 0.935 vs. 0.679 (`smile`), and 0.982 vs. 0.694 (`glasses`) in the setting (a), and 0.906 vs. 0.408 (`smile`), and 0.939 vs. 0.536 (`glasses`) in the setting (b). These results clearly demonstrate the strong compositionality of our method.

Compositions of energy functions EBMs have shown great potential of concept compositionality with various ways of composing energy functions [8]. Our method can inherit their compositionality, while achieving high image quality. Inspired by [8], we also consider composing energy functions with three logical operators: conjunction (AND), disjunction (OR) and negation (NOT). In particular, given two attributes $\{c_1, c_2\}$, we can define the joint energy function of each logical operator as

$$E(z, \{c_1 \text{ AND } c_2\}) := E(c_1|g(z)) + E(c_2|g(z)) + {}^1\!/\!_2\|z\|_2^2$$
$$E(z, \{c_1 \text{ OR } c_2\}) := -\log\left(e^{\beta - E(c_1|g(z))} + e^{-E(c_2|g(z))}\right) + {}^1\!/\!_2\|z\|_2^2$$
$$E(z, \{c_1 \text{ AND } (\text{NOT } c_2)\}) := E(c_1|g(z)) - \alpha E(c_2|g(z)) + {}^1\!/\!_2\|z\|_2^2$$

where the AND operator actually boils down to the conditional sampling with multiple attributes, and $\alpha, \beta > 0$ are tunable hyperparameters to balance the importance of energy functions. For more complex compositionality, we can recursively apply these logical operators.

Figure 6 demonstrates the concept compositionality of our method for `glasses` and `yaw` where we set $\beta = \ln 10$, and $\alpha = \min(\frac{0.1}{|E(z,c_2)|}, 1)$. We can see the generated images not only precisely follow



(a) `beard=1,smile=0,glasses=1,age=15` (*Left*: StyleFlow, *Right*: Ours)

(b) `gender=female,smile=0,glasses=1,age=10` (*Left*: StyleFlow, *Right*: Ours)

Figure 4: Zero-shot conditional sampling results of our method and StyleFlow [1] on FFHQ, where none of the above two attribute combinations is seen in the training set. Our method performs well on zero-shot generation while StyleFlow almost fails by either generating low-quality images or missing the conditioning information.

the rule of the given logical operators, but are also sufficiently diverse to cover all possible logical cases. For instance, given {`glasses=1 OR yaw=front`}, some images have glasses regardless of the yaw while other images satisfy `yaw=front` regardless of the glasses. Our method also works well for the recursive combinations of logical operators, as shown in the bottom-right of Figure 6. To the best of our knowledge, our method is the first to show such strong compositionality when controllably generating photo-realistic images of resolution 1024×1024.

3.4 Ablation study on the ODE and LD sampler

Here, we compare the ODE and LD sampler more thoroughly, with grid search in a large range of hyperparameter settings (see Appendix A.9 for details). Figure 5 shows the ACC and FID (i.e., ACC-FID) distributions of the two samplers on CIFAR-10, where each dot in the figure refers to the result of a particular hyperparameter setting. After grid research, there are 81 and 104 hyperparameter settings for the ODE and LD sampler, respectively.

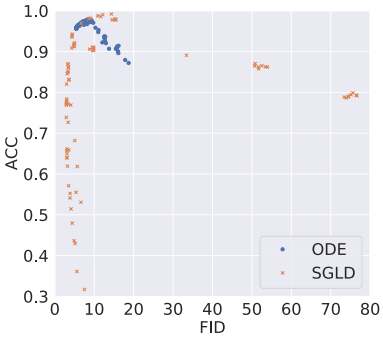


Figure 5: The ACC-FID distributions for the ODE and LD sampler, where each dot denotes each hyperparameter setting.

We can see that (i) the best ACC-FID scores (on the top-left of Figure 5) of the two samplers heavily overlap, implying that they perform equally well in their best tuned hyperparameter settings. (ii) With the concentrated ACC-FID scores for the ODE sampler on the top-left, we can see that the ODE sampler is more stable and less sensitive to the choice of hyperparameters than the LD sampler. (iii) When focusing on the top-left with ACC $\geq 95\%$ and FID ≤ 10, the ODE sampler needs much smaller (less than 21%) number of minimum average steps than the LD sampler (41.8 vs. 200). It shows that given the similar performance, the ODE sampler tends to be more efficient due to its adaptivity in step sizes.

4 Related work

Conditional generative models Significant efforts for conditional sampling and image editing have been invested in conditional generative models. In these models, various conditional information can be used to guide the generation process, such as class labels [29, 25], attribute codes [30], source images [7], text descriptions [34], and semantic maps [32]. Training these models is costly, in particular for generating high-resolution images. Additionally, adding or modifying the conditional information requires re-training the whole network. On the contrary, our method is based on pre-trained unconditional models, with little effort to introduce new conditional information.

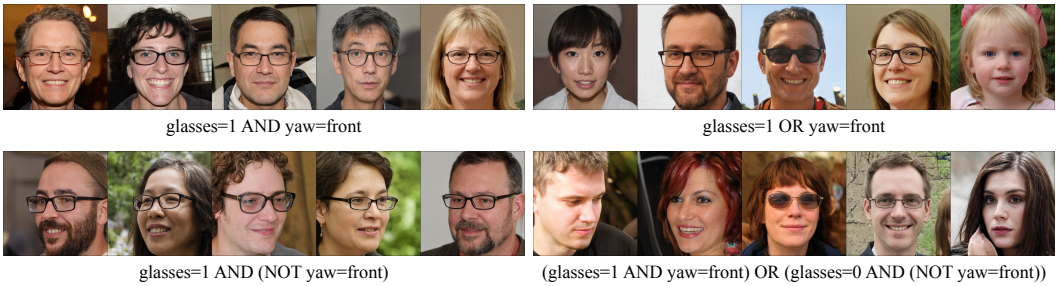


Figure 6: Compositions of energy functions in our method with different logical operators: conjunction (AND), disjunction (OR), negation (NOT), and their recursive combinations on FFHQ of resolution 1024×1024.

Latent code manipulation for image editing An alternative approach to conditional GANs is to manipulate the latent code of pre-trained unconditional models, in particular GANs. Some works explore linear manipulations of latent code [19, 13, 35], while others consider more complicated nonlinear manipulations [1, 33]. These methods can discover the directions that often correspond to meaningful semantic edits, and need not train conditional generative models from scratch. However, they tend to have issue with compositionality. For instance, StyleFlow [1] specifies a fixed length of attributes during training, limiting its ability to generalize to new attributes and novel combinations, which has been evidenced by its poor empirical performance in the zero-shot generation tasks.

EBMs and score-based models Many works have applied EBMs and score-based models for controllable and compositional generation. [12] proposes the joint EBM and shows its ability in class-conditional generation. [8] explores the compositionality of conditional EBMs in controllable generation. [38] proposes a unified framework of score-based models from the SDE perspective, and performs controllable generation by using both a conditional reverse-time SDE and its corresponding ODE. Unlike our method that works in the latent space, they all formulate energy functions or score functions in the pixel space, making them both more challenging to train and orders of magnitude slower to sample for high-resolution images. Also, several prior works have built EBMs in the latent space of generative models [42, 4, 31, 2], however, none of them focuses on controllable generation.

5 Discussions and limitations

We proposed a novel formulation of a joint EBM in the latent space of pre-trained generative models for controllable generation. Based on our formulation, all we need for controllability is to train an attribute classifier, and sampling is done in the latent space. Moreover, we proposed a more stable and adaptive sampling method by formulating it as solving an ODE. Our experimental results showed that our method is fast to train and efficient to sample, and it outperforms state-of-the-art techniques in both conditional sampling and sequential editing. With our strong performance in the zero-shot generation with unseen attribute combinations and compositions of energy functions with logical operators, our method also demonstrated compositionality in generating high-quality images.

One limitation of this work is that the generation quality is limited by the generative power of the underlying pre-trained generator. Since our EBM can be applied to any latent-variable model including GANs and VAEs, it is interesting to apply our method to different generative models based on downstream tasks. Another limitation is that training the attribute classifier requires the availability of attribute labels. We believe that advanced semi-supervised and self-supervised techniques [36, 6] can improve the classifier training while reducing the dependency on labels.

6 Broader impact

The method presented in this work enables high-quality controllable image synthesis, built upon pre-trained generative models (e.g., StyleGAN2 [22]). Technically, it inherits the compositionality of EBMs and overcomes their difficulty in generating high-resolution images. Since our method does not train any conditional generator from scratch, it significantly reduces the computational cost of training generators for new conditioning attributes.

On the application side, it shares with other image synthesis tools similar potential benefits and risks, which have been discussed extensively in [3, 39]. Our method does not produce new images but only guides the generation process of existing generators. Thus, it inherits potential biases from the pre-trained generators, e.g. StyleGAN2 trained on the FFHQ dataset [21]. On the other hand, by providing a semantic control of image generation with strong compositionality, our method can be used to discover and proactively reduce unknown biases in existing generators.

References

[1] Abdal, R., Zhu, P., Mitra, N. J., and Wonka, P. (2021). Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Trans. Graph.*

[2] Arbel, M., Zhou, L., and Gretton, A. (2020). Generalized energy based models. *arXiv e-prints*, pages arXiv–2003.

[3] Bailey, J. (2020). The tools of generative art, from flash to neural networks. *Art in America*.

[4] Che, T., Zhang, R., Sohl-Dickstein, J., Larochelle, H., Paull, L., Cao, Y., and Bengio, Y. (2020). Your gan is secretly an energy-based model and you should use discriminator driven latent sampling. *arXiv preprint arXiv:2003.06060*.

[5] Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. (2018). Neural ordinary differential equations. *Advances in Neural Information Processing Systems*.

[6] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

[7] Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S., and Choo, J. (2018). Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797.

[8] Du, Y., Li, S., and Mordatch, I. (2020a). Compositional visual generation with energy based models. In *Advances in Neural Information Processing Systems*.

[9] Du, Y., Li, S., Tenenbaum, J., and Mordatch, I. (2020b). Improved contrastive divergence training of energy based models. *arXiv preprint arXiv:2012.01316*.

[10] Du, Y. and Mordatch, I. (2019). Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*.

[11] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems (NIPS)*.

[12] Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., Norouzi, M., and Swersky, K. (2020). Your classifier is secretly an energy based model and you should treat it like one. In *Internal Conference of Learning Representations*.

[13] Härkönen, E., Hertzmann, A., Lehtinen, J., and Paris, S. (2020). Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546*.

[14] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *arXiv preprint arXiv:1706.08500*.

[15] Hinton, G. (1999). Products of experts. In *Proceedings of the Ninth International Conference on Artificial Neural Networks*.

[16] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*.

[17] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

[18] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

[19] Jahanian, A., Chai, L., and Isola, P. (2019). On the" steerability" of generative adversarial networks. *arXiv preprint arXiv:1907.07171*.

[20] Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., and Aila, T. (2020a). Training generative adversarial networks with limited data. In *Advances in Neural Information Processing Systems*.

[21] Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[22] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. (2020b). Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*.

[23] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[24] Kingma, D. P., Rezende, D. J., Mohamed, S., and Welling, M. (2014). Semi-supervised learning with deep generative models. *arXiv preprint arXiv:1406.5298*.

[25] Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[26] Krizhevsky, A., , and Hinton, G. (2009). Learning multiple layers of features from tiny images. *(Technical Report) University of Toronto.*

[27] LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. (2006). A tutorial on energy-based learning. *Predicting structured data*, 1(0).

[28] Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

[29] Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

[30] Nie, W., Karras, T., Garg, A., Debnath, S., Patney, A., Patel, A., and Anandkumar, A. (2020). Semi-supervised stylegan for disentanglement learning. In *International Conference on Machine Learning*, pages 7360–7369. PMLR.

[31] Pang, B., Han, T., Nijkamp, E., Zhu, S.-C., and Wu, Y. N. (2020). Learning latent space energy-based prior model. *arXiv preprint arXiv:2006.08205*.

[32] Park, T., Liu, M.-Y., Wang, T.-C., and Zhu, J.-Y. (2019). Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346.

[33] Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., and Lischinski, D. (2021). Styleclip: Text-driven manipulation of stylegan imagery. *arXiv preprint arXiv:2103.17249*.

[34] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. (2016). Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, pages 1060–1069. PMLR.

[35] Shen, Y. and Zhou, B. (2021). Closed-form factorization of latent semantics in gans. In *CVPR*.

[36] Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*.

[37] Song, Y. and Kingma, D. P. (2021). How to train your energy-based models. *arXiv preprint arXiv:2101.03288*.

[38] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

[39] Vaccari, C. and Chadwick, A. (2020). Deepfakes and disinformation: Exploring the impact of synthetic political video on deception, uncertainty, and trust in news. *Social Media+ Society*, 6(1):2056305120903408.

[40] Vahdat, A. and Kautz, J. (2020). NVAE: A deep hierarchical variational autoencoder. In *Neural Information Processing Systems (NeurIPS)*.

[41] Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688.

[42] Xiao, Z., Kreis, K., Kautz, J., and Vahdat, A. (2020). Vaebm: A symbiosis between variational autoencoders and energy-based models. *arXiv preprint arXiv:2010.00654*.

[43] Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section 1.
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 2.2 and Section 2.3.
   (b) Did you include complete proofs of all theoretical results? [Yes] We defer the proofs to Appendix A.1.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We include the code to reproduce our results in the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 3 and Appendix A.2 for more details.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report the averaged results in Section 3 and the corresponding results with error bars are reported in the Appendix.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix A.2.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 3 and Appendix A.3 where we cite the datasets, pre-trained generative models, and the baseline methods we have used.
   (b) Did you mention the license of the assets? [Yes] We mention the license of the assets in the submitted code.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] We only use publicly available datasets (CIFAR-10 and FFHQ), which are standard benchmarks in the generative modeling community.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]