EFFICIENT CONTROLLABLE GENERATION WITH GUARANTEE

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative models have achieved great success in image synthesis, and controllability of the generative process is a key requirement for their successful adoption in real-world applications. Most existing methods for controllable generation lack theoretical guarantees and are time-consuming, which weakens their reliability and applicability. In this paper, we propose an identifiability theorem to provide a guarantee of controllability. This theorem ensures that semantic attributes can be disentangled and hence independently controlled by orthogonalization in latent space in a supervised manner. Based on the theoretical analysis, we propose a general method for controllable generation, which can be integrated with most latent-variable generative models. We further propose to plug it into a pre-trained NVAE. Such a scheme significantly reduces the cost of time and has better consistency in image editing due to the merits of NVAE. Experiments show that our method is comparable with the state-of-the-art methods in attribute-conditional generation and image editing, and has advantages in efficiency and consistency.

1 INTRODUCTION

In recent years, generative models have achieved great success in image synthesis. Some models can even synthesize very high-quality images, including generative adversarial networks (GANs) (Goodfellow et al., 2020; Karras et al., 2019), variational autoencoders (VAEs) (Kingma & Welling, 2013; Vahdat & Kautz, 2020) and diffusion models (Ho et al., 2020). However, most real-world applications of these models require the controllability of the generative process. Such controllability should enable generative models to generate images with some given semantic attributes. Another goal of controllability is to utilize generative models to edit some designative semantic attributes of a given image without changing other semantics.

A common paradigm for controllable generation is to train conditional generative models (Kingma et al., 2014; Mirza & Osindero, 2014; Nie et al., 2020). However, it is expensive to train generative models from scratch, and involving new attributes in a pre-trained conditional model without retraining is difficult (Nie et al., 2021). Moreover, such a paradigm limits the image editing ability of generative models, hence is not a satisfactory scheme for the controllable generation.

Another paradigm is to enforce disentanglement of latent variables in generative models in an unsupervised (Locatello et al., 2019; Sorrenson et al., 2020) or weakly-supervised manner (Shu et al., 2019; Locatello et al., 2020). Some of these works have theoretical guarantees (Hyvarinen et al., 2019; Khemakhem et al., 2020; Yang et al., 2021), but often have obstacles in dealing with real-world images due to overly restrictive conditions.

Recent works for controllable generation can be categorized into two types. One type aims to discover the semantically meaningful directions in the latent space of pre-trained generative models by sophisticated analysis and designs (Karras et al., 2019; Jahanian et al., 2019; Voynov & Babenko, 2020; Härkönen et al., 2020; Shen & Zhou, 2021), or directly converting the models to conditional versions (Abdal et al., 2021). Another type is to leverage the compositionality of energy-based models (EBMs) to achieve controllable generation (Du et al., 2020; Nie et al., 2021) in a supervised manner. The essence of these works is to discover the paths in latent or pixel space for independently manipulating attributes.

Although these works have achieved great success in many aspects of controllable generation, they still have some drawbacks. One is the lack of theoretical guarantees, hence they cannot ensure that other semantics are unchanged when manipulating some attributes. More importantly, most of them generally have difficulty in image editing since they base on GANs. Specifically, it is expensive to obtain the latent code of a given image using GANs. Besides, when editing some attributes, other semantics like background are often inconsistent due to the GANs' weak reconstruction ability. Moreover, those EBM-based methods mainly utilize Markov chain Monte Carlo (MCMC) to edit attributes, which is also time-consuming.

In this work, we propose a rigorous theory for controllable generation, which guarantees that different attributes can be manipulated independently. Specifically, this theory shows that with some natural and mild conditions, if different attributes are identified by a set of orthogonal directions in latent space, then manipulations of them are independent. Based on the theoretical analysis, we further propose a general method to achieve controllable generation. In this method, we utilize different blocks of latent variables to predict different attributes via linear models, hence the weights of these linear models form a set of orthogonal directions in latent space. These directions provide an efficient way to manipulate their corresponding attributes. The proposed method can be plugged into pre-trained generative models, by involving a normalizing flow to adapt the latent space for controllable generation.

We further integrate our proposed method with Nouveau VAE (NVAE) (Vahdat & Kautz, 2020). Such a scheme is efficient for image editing, as the latent code can be directly obtained by NVAE's encoder. Besides, when editing some attributes, other semantics (like background in images) is unchanged (consistent) due to NVAE's strong reconstruction ability. Experiments on FFHQ show that our method is comparable with the state-of-the-art methods in attribute-conditional generation, and is superior in image editing due to efficiency and consistency.

Our contributions are summarized as follows:

- We propose a novel theory, which provides a rigorous guarantee to supervised controllable generation. The conditions are natural and mild, hence the theory might be widely applicable.
- Based on the theory, we propose a general method, which can be integrated with most latent-variable generative models.
- We integrate our method with NVAE and achieve great success in experiments. To our best knowledge, this is the first work to develop NVAE for controllable generation, which opens a new path towards controllable generation.

2 Theory: Connecting Disentanglement with Orthogonality

Consider an observed random vector $\mathbf{x} \in \mathbb{R}^d$ with observed attributes $\mathbf{y} \in \mathbb{R}^n$, where the attributes can be continuous or discrete. We assume these attributes are conditionally independent and determined by a set of underlying factors $\mathbf{s} = \mathbf{f}^*(\mathbf{x}) \in \mathbb{R}^n$ separately:

$$p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} p(\mathrm{y}_i|\mathbf{x}) \tag{1}$$

$$p(\mathrm{y}_i|\mathbf{x}) = p(\mathrm{y}_i|\mathrm{s}_i). \tag{2}$$

These equations mean that all information about y_i in $\mathbf{x}$ is preserved in s_i, and hence different attributes are disentangled as their information is preserved in different dimensions of $\mathbf{s}$. Therefore, it is possible to control attribute y_i by manipulating s_i.

To identify the $\mathbf{s}$, we involve an estimating model $\mathbf{z} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}) \in \mathbb{R}^m$. Rather than directly using z_i to identify s_i following most prior works (Hyvarinen & Morioka, 2017; Hyvarinen et al., 2019; Khemakhem et al., 2020), here we choose the projection of $\mathbf{z}$ on a direction $\boldsymbol{a}_i^\top \mathbf{z}$, where $\boldsymbol{a}_i \in \mathbb{R}^m$ is a unit vector (representing a direction). This choice is much more general, as z_i is simply a special projection of $\mathbf{z}$: $\mathrm{z}_i = \boldsymbol{e}_i^\top \mathbf{z}$, where $\boldsymbol{e}_i$ is i-th base of $\mathbb{R}^m$. More importantly, $\boldsymbol{a}_i$ can be trainable, hence it can be chosen by optimization, which is possibly much better than a handcrafted one.

Similar with Eq. 2, we suppose the conditional distribution of y_i predicted via the estimating model is determined by $\boldsymbol{a}_i^\top \mathbf{z}$,

$$q(\mathrm{y}_i|\mathbf{x}) = q(\mathrm{y}_i|\boldsymbol{a}_i^\top \mathbf{z}). \tag{3}$$

Our key insight here is that if $\{\boldsymbol{a}_i\}_{i=1}^n$ is a set of orthogonal bases in $\mathbb{R}^m$ (and hence $m \geq n$), then manipulating $\boldsymbol{a}_j^\top \mathbf{z}$ cannot change the estimated distribution of y_i, $\forall\, j \neq i$. Hence we suppose

$$\boldsymbol{a}_i^\top \boldsymbol{a}_j = 0,\ \forall i \neq j. \tag{4}$$

With some additional mild conditions, we can prove that in the model defined above, s_i is identified by $\boldsymbol{a}_i^\top \mathbf{z}$, and manipulating $\boldsymbol{a}_j^\top \mathbf{z}$ does not change s_i, $\forall j \neq i$. The proof is reported in Appendix A.

Theorem 1 ***(Identifiability)*** *If the model defined by Eq. (1)-(4) also fulfills the following two conditions: $\forall\, i \in [n]$,*

(i) $\phi_i : \mathrm{s}_i \to p(\mathrm{y}_i|\mathrm{s}_i)$ *and* $\psi_i : \boldsymbol{a}_i^\top \mathbf{z} \to q(\mathrm{y}_i|\boldsymbol{a}_i^\top \mathbf{z})$ *are continuous and bijective maps;*

(ii) $p(\mathrm{y}_i|\mathbf{x}) = q(\mathrm{y}_i|\mathbf{x})$.

Then $\mathrm{s}_i = g_i(\boldsymbol{a}_i^\top \mathbf{z})$, $\forall\, i \in [n]$, *where g_i is a strictly monotonic function, and manipulating $\boldsymbol{a}_j^\top \mathbf{z}$ does not change s_i, $\forall\, j \neq i$.*

Conditions (i) and (ii) are natural in the field of identifiable latent models. In condition (i), 'bijective' indicates that distributions should preserve all information of the conditions. As for continuity, intuitively, it means that when the variation of s_i (or $\boldsymbol{a}_i^\top \mathbf{z}$) approach zero, the variation of $p(\mathrm{y}_i|\mathrm{s}_i)$ (or $q(\mathrm{y}_i|\boldsymbol{a}_i^\top \mathbf{z})$) should also approach zero. This is true in most cases as long as the parameters of distributions are continuous and bijective functions of s_i or $\boldsymbol{a}_i^\top \mathbf{z}$, as will be shown in the following examples. Condition (ii) requires the ground truth to be well estimated, which is achievable.

For clarity, next we provide two specific examples of the model defined by Eq. (1)-(4) and conditions (i) and (ii), which are applicable in our experimental setting.

Example 1 ***(Bernoulli distribution)*** *Suppose y_i is a Bernoulli random variable, $p(\mathrm{y}_i|\mathrm{s}_i) = h_i^*(\mathrm{s}_i)^k(1 - h_i^*(\mathrm{s}_i))^{1-k}$ and $q(\mathrm{y}_i|\boldsymbol{a}_i^\top \mathbf{z}) = h_i(\boldsymbol{a}_i^\top \mathbf{z})^k(1 - h_i(\boldsymbol{a}_i^\top \mathbf{z}))^{1-k}$, $k \in \{0, 1\}$, where h_i^* and h_i are continuous and bijective functions, then $\mathrm{s}_i = h_i^{*-1} \circ h_i(\boldsymbol{a}_i^\top \mathbf{z})$. Note that $h_i^{*-1} \circ h_i$ is continuous and bijective, and hence is a strictly monotonic function.*

Example 2 ***(Gaussian distribution)*** *Suppose y_i is a continuous random variable, $p(\mathrm{y}_i|\mathrm{s}_i) = \mathcal{N}(\mathrm{y}_i|\mu_i^*(\mathrm{s}_i), \sigma_i)$ and $q(\mathrm{y}_i|\boldsymbol{a}_i^\top \mathbf{z}) = \mathcal{N}\left(\mathrm{y}_i|\mu_i(\boldsymbol{a}_i^\top \mathbf{z}), \sigma_i\right)$, where μ_i^* and μ_i are continuous and bijective functions, and σ_i is a positive constant, then $\mathrm{s}_i = \mu_i^{*-1} \circ \mu_i(\boldsymbol{a}_i^\top \mathbf{z})$. Here $\mu_i^{*-1} \circ \mu_i$ is also a strictly monotonic function.*

To summarize, we propose an identifiability theorem that connects disentanglement with orthogonality. Specifically, we show that in our introduced model, disentangled attributes can be identified by a set of orthogonal directions in latent space. This theorem can motivate a general and applicable method, as will be shown in the next section.

3 METHOD

In this section, we describe a general and applicable method based on our theory. This method can achieve controllable generation, by mapping attributes into orthogonal directions in latent space of latent-variable generative models, hence we term it as **LA**tent **O**rthogonalization (LAO) through this work.

3.1 MODEL SETTING

For a generative model with inference module $\mathbf{z} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$, our goal is to let $\mathbf{z}$ satisfy $q(\mathrm{y}_i|\mathbf{x}) = q(\mathrm{y}_i|\boldsymbol{a}_i^\top \mathbf{z})$ (Eq. 3) and $p(\mathrm{y}_i|\mathbf{x}) = q(\mathrm{y}_i|\mathbf{x})$ (condition (ii)), $\forall\, i \in [n]$. Some constraints are imposed in this process, includes orthogonality of $\{\boldsymbol{a}_i\}_{i=1}^n$ (Eq. 4), and both continuity and reversibility of $\psi : \boldsymbol{a}_i^\top \mathbf{z} \to q(\mathrm{y}_i|\boldsymbol{a}_i^\top \mathbf{z})$ (condition (i)).

To achieve the goal, a simple approach is to minimize the cross entropy as follows:

$$\min_{\boldsymbol{\theta},\boldsymbol{A}} -\sum_{i=1} \mathbb{E}_{p(\mathrm{y}_i|\mathbf{x})} \left[\log q\left(\mathrm{y}_i|\boldsymbol{a}_i^\top \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})\right)\right], \tag{5}$$

where $\boldsymbol{A} = (\boldsymbol{a}_1, \cdots, \boldsymbol{a}_n)^\top$. The optimal solution is $q(\mathrm{y}_i|\boldsymbol{a}_i^\top \mathbf{z}) = p(\mathrm{y}_i|\mathbf{x})$. Note that in the optimal case, other information in $\mathbf{x}$ (besides $\boldsymbol{a}_i^\top \mathbf{z}$) cannot further improve the estimation of $p(\mathrm{y}_i|\mathbf{x})$, hence we have $q(\mathrm{y}_i|\mathbf{x}) = q(\mathrm{y}_i|\boldsymbol{a}_i^\top \mathbf{z})$ and $p(\mathrm{y}_i|\mathbf{x}) = q(\mathrm{y}_i|\mathbf{x})$.

To enforce orthogonality of $\{\boldsymbol{a}_i\}_{i=1}^n$, we choose a simple but useful setting: decompose $\mathbf{z}$ into many blocks $\mathbf{z} = (\mathbf{z}_i, \cdots, \mathbf{z}_n, \mathbf{z}_{res})$ ($\mathbf{z}_{res}$ represents the residual components for noise or other attributes apart from $\mathbf{y}$), and then let $\boldsymbol{a}_i$ only act on $\mathbf{z}_i$. In other words, the entries of $\boldsymbol{a}_i$ acting on $\mathbf{z}_j$ are set as zero, $\forall\, j \neq i$, hence $\boldsymbol{a}_i^\top \boldsymbol{a}_j = 0$. For the sake of argument, we denote the entries of $\boldsymbol{a}_i$ acting on $\mathbf{z}_i$ by $\boldsymbol{\alpha}_i$, then $\boldsymbol{a}_i^\top \mathbf{z} = \boldsymbol{\alpha}_i^\top \mathbf{z}_i$. Note that there exists many other setting or algorithms to enforce orthogonality, and they are possible to involved into our method and adapt it to different scenes and tasks.

As for continuity and reversibility in condition (i), it is sufficient in most cases to use continuous and bijective functions to transform $\boldsymbol{\alpha}_i^\top \mathbf{z}_i$ into parameters of $q(\mathrm{y}_i|\boldsymbol{\alpha}_i^\top \mathbf{z}_i)$, as shown in Examples 1 and 2. Specifically, we use Sigmoid (denoted by S) and identity for Bernoulli and continuous y_i, respectively. Then the log-likelihood

$$\log q(\mathrm{y}_i|\boldsymbol{\alpha}_i^\top \mathbf{z}_i) = \begin{cases} \mathrm{y}_i \log S(\boldsymbol{\alpha}_i^\top \mathbf{z}_i) + (1-\mathrm{y}_i)\log\left(1 - S(\boldsymbol{\alpha}_i^\top \mathbf{z}_i)\right) & \text{if } \mathrm{y}_i \text{ is Bernoulli} \\ -\frac{1}{2\sigma_i^2}(\boldsymbol{\alpha}_i^\top \mathbf{z}_i - \mathrm{y}_i)^2 - c & \text{if } \mathrm{y}_i \text{ is continuous} \end{cases}, \tag{6}$$

where c is a constant. Substitute it into Eq. 5, we can see that minimizing the cross entropy is equivalent to binomial logistic regression and least square regression when y_i is Bernoulli and continuous random variable, respectively.

Therefore, our method motivated by the proposed theory ultimately becomes a simple but elegant scheme: $\mathbf{z}$ is decomposed into several blocks, and each block is utilized to predict an attribute via a linear model. Note that the inference model $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$ and linear models (parameterized by $\boldsymbol{A}$) are jointly optimized, and hence these linear models can be viewed as regularization for the inference model to enforce the linearity of $\mathbf{z}_i$. As will be shown later, such linearity has great advantages for controllable generation.

3.2 Efficient Controllable Generation

Most previous works for controllable generation do not require linearity, and as a result, they mainly utilize Markov chain Monte Carlo (MCMC) to control the estimated latent variables. This is a time-consuming scheme, and the cost of time is almost proportional to the size of attributes n. Specifically, the control of each attribute requires hundreds of searching steps in MCMC, which is the main cost of time for controlling, and the total required steps are proportional to n.

Our method needs only one step for controlling, which is a great advantage. Due to the linearity of the optimized $\mathbf{z}_i$ in our method, we can control attributes y_i by simply manipulating $\mathbf{z}_i$ in direction $\boldsymbol{\alpha}_i$. To see this, consider the conclusion of Theorem 1: $\mathrm{s}_i = g_i(\boldsymbol{\alpha}_i^\top \mathbf{z}_i)$, we can update s_i by updating $\boldsymbol{z}_i$,

$$\boldsymbol{z}_i \leftarrow \boldsymbol{z}_i + \lambda_i \boldsymbol{\alpha}_i \Rightarrow s_i \leftarrow g_i(\boldsymbol{\alpha}_i^\top \boldsymbol{z}_i + \lambda_i), \tag{7}$$

where $\boldsymbol{\alpha}_i$ is normalized, and λ_i is the step length.

Such an update leads to manipulation of the distribution of attribute y_i, hence controls the attribute. For example, if $\mathrm{y}_i = 1$ represents smiling and $\mathrm{y}_i = 0$ represents not smiling in a face image, and s_i is the probability of smiling (i.e. h_i^* is an identity in Example 1), then we can control the level of smiling in the face image by updating $\boldsymbol{z}_i$ as stated above.

The step length λ is determined by the desirable level of the given attribute. Specifically, if we want a Bernoulli y_i to take value 1 with probability q_0, then according to the expression of $q(\mathrm{y}_i|\boldsymbol{\alpha}_i^\top \boldsymbol{z}_i)$, we can choose

$$\lambda_i^* = \log \frac{q_0}{1-q_0} - \boldsymbol{\alpha}_i^\top \boldsymbol{z}_i, \tag{8}$$

where $\log \frac{q_0}{1-q_0}$ is the inverse function of Sigmoid.

3.3 PLUG-IN VERSION

The method described above requires the joint training of inference model $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$ and linear models $\boldsymbol{A}$, which might be an obstacle in many generative models. First, some generative models have no explicit inference module like GANs, and some cost a lot of time for training from scratch. Moreover, it is more difficult to simultaneously learn generation and controllability for most generative models.

To solve these problems, we propose a plug-in version of our method above, which can be easily plugged into pre-trained generative models to improve their controllability without harming their generative ability. Specifically, for a pre-trained generative model with latent variables $\mathbf{w}$, we propose to transform it to another set of latent variables $\mathbf{z}$ with a normalizing flow (denoted by $f_{\boldsymbol{\theta}}$ without ambiguity), then $\mathbf{z} = f_{\boldsymbol{\theta}}(\mathbf{w})$ and linear models $\boldsymbol{A}$ on $\mathbf{z}$ are jointly trained to enforce controllability.

Note that if the normalizing flow is volume-preserving, and the prior of it is set as a standard Gaussian, then maximizing the log-likelihood of $\mathbf{z}$ is equivalent to penalizing $\|\mathbf{z}\|_2^2$. This is a regularization for the linear models to prevent it form over-fitting. The optimization of non-volume-preserving normalizing flows is also a similar regularization for linear models.

Such a scheme does not require an explicit inference module, as long as the pair of latent variables and attributes $(\mathbf{w}, \mathbf{y})$ is given for training. Therefore, our plug-in method can be integrated with a wide range of generative models.

3.4 INTEGRATION WITH NVAE

NVAE (Vahdat & Kautz, 2020) has many advantages for controllable generation. First, NVAE has an explicit inference module and hence is much faster to infer the latent variables from images compared with GAN-based models, which might make image editing much more efficient. More importantly, the reconstructions of images by NVAE are much better than GAN-based models, as shown in Figure 1, and this is vital for the consistency during editing. Otherwise, the edits of some semantic attributes in images might changes the background and some irrelevant details. Besides, NVAE assembles most of the global information into several top groups of latent variables, which enables us to achieve controllable generation by handling several groups of latent variables. These merits of NVAE are further introduced and discussed in Appendix B.

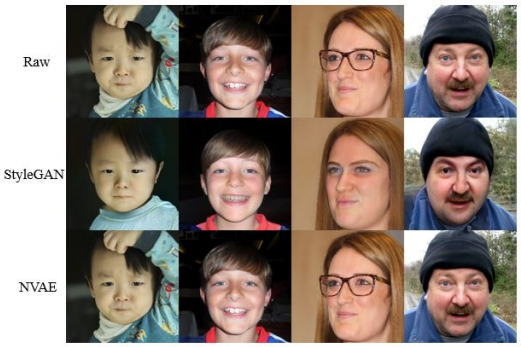


Figure 1: Comparison of reconstructions by StyleGAN and NVAE. StyleGAN often fails to reconstruct some semantics like the background and some details, while NVAE can almost perfectly reconstruct the raw images.

Therefore, in this paper, we integrate our plug-in method with NVAE. The integration is summarized in Figure 2, named by LAO-NVAE. In NVAE, the latent variables are divided into L groups, hence we transform each group with a normalized flow. Here we choose volume-preserving normalized flow for simplicity, and we empirically find it sufficient for the controllability of NVAE. Then for merging the information between different groups, we exact one channel from each group and concatenate them to form $\mathbf{z}_i$. Finally $\mathbf{z}_i$ is utilized to predict $q(\mathrm{y}_i|\mathbf{x})$ via a linear model. See Appendix C for more details and discussions.

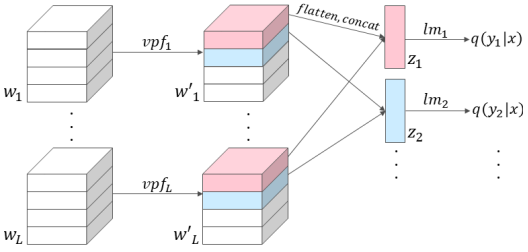


Figure 2: Pipeline of predicting $q(\mathrm{y}_i|\mathbf{x})$ by our method integrated with NVAE. $\mathbf{w}_i$ is the i-th group of latent variables encoded by NVAE, vpf_i represents the volume-preserving normalizing flow for transforming $\mathbf{w}_i$, and lm_i represents the linear model for predicting $q(\mathrm{y}_i|\mathbf{x})$ from $\mathbf{z}_i$.

(a) glasses=1 (b) gender=female, smile=1, age=35

Figure 3: Samples conditionally synthesized by LAO-NVAE ($t = 0.5$) on FFHQ.

Table 1: Comparison against the baselines on attribute-conditional sampling. 'Gen' represents the sigle GPU time for generating a batch of 16 images.

Methods	glasses			gender_smile_age				
	Gen	FID↓	ACC_{gl} ↑	Gen	FID↓	ACC_{ge} ↑	ACC_{sm} ↑	ACC_{ag} ↑
StyleFlow	0.61s	42.08	0.899	0.61s	43.88	0.718	0.870	0.874
LACE-LD	1.15s	20.92	0.998	2.40s	22.97	0.955	0.960	0.913
LACE-ODE	0.68s	20.93	0.998	4.81s	24.52	0.969	0.982	0.914
LAO-NVAE	0.64s	21.72	0.982	0.65s	25.78	0.957	0.952	0.895

4 EXPERIMENTS

In this section, we show that our proposed method LAO is powerful in conditional sampling and semantic editing when integrating with NAVE, named by LAO-NVAE, and is advantageous in efficiency and consistency for image editing.

Experimental setting We use NVAE with officially provided checkpoint as the pre-trained generative model for experiments on FFHQ. Such NVAE has 36 groups of latent variables, which have increasing size from top to bottom. We utilize the top 12 groups of latent variables in NVAE and empirically find them good enough for reconstructions, which largely reduces the cost for controllability. The temperature for NVAE is set as 0.5 on FFHQ following (Vahdat & Kautz, 2020).

The normalizing flow for transforming the latent variables is a volume-preserving version of real NVP (Dinh et al., 2016) with 24 coupling layers. We make real NVP volume-preserving by setting the sum of scale factor in coupling layers as zero (see Appendix D for details). The prior of normalizing flow is simply a standard Gaussian, and the transformed latent variables. To prevent imbalance between different groups of latent variables in NVAE due to their different size, the log-likelihood of each normalizing flow is divided by size of their inputs. The linear models for predicting are simply linear layers, with sigmoid activation for Bernoulli attributes and identity for continuous attributes. The variances of continuous attributes (σ_i in Example 2) are set as 0.1, and we empirically find that binary cross entropy loss also works well for continuous attributes. The ranges of all attributes' labels are set in $[0, 1]$. The weight of log-likelihood of normalizing flow is set as 0.05 in all experiments.

For evaluation, the considered attributes on FFHQ include smile, age, glasses, gender, beard and yaw, in which age and yaw are treated as continuous attributes, and others are Bernoulli. Following (Abdal et al., 2021), we use conditional accuracy (ACC) to measure the controllability, and FID score to measure the quality and diversity of generated images. To compute FID, we sample 1k generated samples from NVAE with 0.5 temperature to represent the target distribution. Such setting is common in works about controllable generation (Abdal et al., 2021; Nie et al., 2021). See (Nie et al., 2021) for detailed introduction.

Model architecture, hyper-parameters and other details are reported in Appendix E.

Baselines For comparisons, we consider the state-of-the-art methods for controllable generation as baselines, includeing StyleFlow (Abdal et al., 2021), and LACE with Langevin dynamics sampler (LACE-LD) and ODE sampler (LACE-ODE) (Nie et al., 2021). Both of them are based on pre-trained StyleGANs (Karras et al., 2019; 2020).

Figure 4: Samples conditioned on continuous manipulation of given attributes by our methods. The two rows are synthesized by manipulating smile and age respectively, while maintaining other attributes unchanged. Each attribute is disentangled with other semantics, including background and other attributes.

Table 2: Comparison against the baselines on semantic editing. 'Infer' represents the sigle GPU time to obtain the latent code of a image.

Methods	yaw_smile_age_glasses							
	Infer	Des↑	ID ↓	FID ↓	ACC_y↑	ACC_s↑	ACC_a↑	ACC_g↑
StyleFlow	102s	0.569	0.549	44.13	**0.947**	0.773	0.817	0.876
LACE-ODE	102s	0.735	0.501	27.94	0.938	0.956	**0.881**	**0.997**
LAO-NVAE	**0.65s**	**0.767**	**0.488**	**21.59**	0.925	**0.961**	0.823	0.967

4.1 Conditional Sampling

Given a set of fixed attributes, the conditional sampling process of our method consists of three step. First, randomly sample a latent code with top 12 groups by NVAE's decoder. Secondly, transform the code and manipulate it in directions corresponding to the given attributes, where the directions are given by the normalized weights of linear models. Here the probability of Bernoulli attributes are manipulated to be 0.95. Finally, reversely transform the manipulated code, fixed them and generate images with NVAE's decoder. In this process, the spare 24 groups are randomly sampled conditioned on the fixed 12 groups. In addition, we use combinations of attributes from training set for sampling, which is necessary for the computation of FID score (Nie et al., 2021).

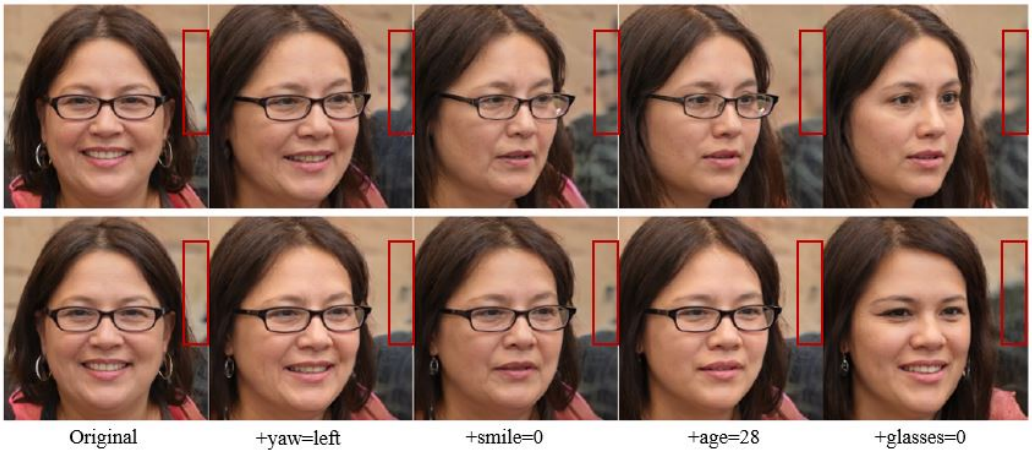
We mainly compare our method LAO-NVAE with baselines on two sets of attributes: glasses and gender_smile_age, and the results are shown in Figure 3 and Table 1. In Figure 3, we can see that the conditional samples by LAO-NVAE are high-quality and diverse, and meanwhile have the corresponding. This result quantitatively demonstrates the effectiveness of LAO-NVAE on attribute-conditional sampling, hence reflect the controllability of our method. In Table 1, our LAO-NVAE has competitive performance in terms of FID and ACC compared with baselines. The scores of our LAO-NVAE are slightly lower than LACEs, but much better than StyleFlow, and we might further improve the scores by using more powerful normalizing flows. Meanwhile, the time cost for sampling by LAO-NVAE is much lower than LACEs, and is close to StyleFlow. Especially, the time for generating conditioned on several attributes by our LAO-NVAE is almost the same as the one-attribute case, because the manipulation of each attribute by LAO-NVAE merely cost one step, as discussed in 3.2. While in LACEs, it takes hundreds of steps to search in latent space. Therefore, our LAO-NVAE combines the advantages of both StyleFlow and LACEs.

We also report the quanlitative results on conditional sampling by continuously manipulating the attributes, as shown in Figure 4. This result indicates that our LAO-NVAE have indeed learned the disentanglement of attributes, as the manipulation of each attribute does not change other visual semantics. In addition, our LAO-NVAE can fast generate a sequence of frames, hence it has potentiality to generate a video.

4.2 Semantic Editing

The process of image editing by our LAO-NVAE is natural: encode the given images, transform the top 12 groups by normalizing flows, manipulate them and then reserve, then use them together with the original images to infer the spare 24 groups, and finally generate the edited images conditioned on all groups.

(a) Edits by LAO-NVAE



(b) Edits by LACE-ODE (top) and StyleFlow (bottom)

Figure 5: Realistic images editing by our method, LACE-ODE and StyleFlow on FFHQ. Figure (b) is cropped from paper (Nie et al., 2021). Note that the edits by our method maintains the background, while the edits by GAN-based models (including LACE and StyleFlow) often change the background (see the red square frame areas in Figure (b)).

The main attributes to edit include yaw, smile, age and glasses. Following (Nie et al., 2021), we adopt two more metrics for quantitative comparisons: the face identity loss (ID) (Patashnik et al., 2021; Abdal et al., 2021) to reflect the identity preservation, the disentangled edit strength (DES) to measure the disentanglement. Besides, we also report the time for infer the latent code, which is vital for the application of editing real-world images. We refer (Nie et al., 2021) for detailed introduction.

As shown in Figure 5, our LAO-NVAE successfully edits the realistic face images. The edits of given attributes does not changes background and other attributes. As for GAN-based methods, including StyleFlow and LACE, their edits often change the background (see the areas in red square frames in Figure (b)) and some details, which is essentially due to the difficulty of reconstruction by GANs, as discussed in 3.4. Therefore, our LAO-NVAE has better consistency in image editing than GAN-based models.

The quantitative results are reported in Table 2. We can see that our LAO-NVAE has better performance in disentanglement, identity preservation and image quality. This is not surprising, as our method encourages disentanglement by orthogonality, and leverages the strong reconstruction ability of NVAE. More importantly, our method takes very short time to encode an image, while both StyleFlow and LACE-ODE take more than one hundred seconds. Note that the time for image edit-

ing consists of the time for encoding and the time for generating from the code, hence our method is far more efficient in image editing than StyleFlow and LACEs. Therefore, our method is more applicable in real-world scenes, as in most applications especially real-time scenes, the cost of time is expensive.

5 Related Works

There are many topics related to controllable generation, including conditional generative models (Kingma et al., 2014; Mirza & Osindero, 2014; Chen et al., 2016; Nie et al., 2020), disentanglement (Locatello et al., 2019; Sorrenson et al., 2020; Shu et al., 2019; Locatello et al., 2020), identifiable latent-variable model (Hyvarinen et al., 2019; Khemakhem et al., 2020; Yang et al., 2021), and so on. In the following discussion, we mainly discuss the topics most related to our theory and method.

Identifibility of linear models A theory closely related to ours is proposed by Roeder et al. (2021), in which linear models are also mainly considered, but their conditions are different from ours. In their theory, a large enough number of classes for classification is necessary for identifying the ground-truth latent variables. While in our theory, even two classes are sufficient (as our result is mainly derived from orthogonality). Therefore, our theory is more suitable for controllable generation.

Manipulation of GAN's latents There are many works that focus on manipulating latent variables of pre-trained GANs. Some works aim at finding the semantic direction in GANs' latent space via subtle methods in an unsupervised manner (Jahanian et al., 2019; Härkönen et al., 2020; Voynov & Babenko, 2020; Shen et al., 2020; Shen & Zhou, 2021). Our method is supervised, hence has guaranteed performance. Abdal et al. (2021) propose to convert a pre-trained StyleGAN (Karras et al., 2019) into a conditional model by involving a conditional normalizing flow to the latent space. While our method utilizes classifiers to inject conditions, which is more flexible to generalize to new attributes. Some other works utilize a classifier on the pixel space of GANs to indirectly control the latent variables (Nguyen et al., 2016; Goetschalckx et al., 2019), while (Nie et al., 2021) directly involve a classifier in latent space. Compared with these works, our method is more efficient for image editing due to the linearity of classifiers, as well as integration with NVAE.

Manipulation of AE's latents A method similar with ours is proposed by Esser et al. (2020). This method uses a normalizing flow to transform the latent variables of auto-encoders, then decomposes the obtained latent variables into several blocks, and finally minimizes the distance of two images with the same attribute in one block. The main difference between this method and ours is the mode of supervision. Our method push the projections of images in latent space to preserve the information of attributes, which is more natural and general. Moreover, our method can provide a semantic direction for each attribute, which is vital for controllable generation and hence is more applicable.

6 Conclusion and Discussion

In this paper, we propose a rigorous theory for controllable generation, which ensures that attributes can be disentangled if they are identified by a set of orthogonal directions in latent space. Based on the theory, we propose a simple but general method for the controllable generation of latent-variable generative models. We plug our method into NAVE and achieve success in attribute-conditional generation and semantic editing. The proposed scheme also has the advantages of efficiency and consistency in image editing.

There are many possible extensions of this work. First, our theory is based on an intuitive definition of 'disentangled attributes' (Eq. 1 and Eq. 2), a close look at this definition might spark new ideas about disentanglement. Moreover, many orthogonalization algorithms can be integrated with our methods to accommodate different tasks. Another appealing idea is to apply our method to multi-modal works like DALL-E (Ramesh et al., 2021). As in this kind of models, the supervised information is essentially entangled, hence the semantic edits of synthesized images are difficult. Our method might be a plug-in component to enhance these models' disentanglement.

REFERENCES

Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics (ToG)*, 40(3):1–21, 2021.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation with energy based models. *Advances in Neural Information Processing Systems*, 33:6637–6647, 2020.

Patrick Esser, Robin Rombach, and Bjorn Ommer. A disentangling invertible interpretation network for explaining latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9223–9232, 2020.

Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5744–5753, 2019.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *Advances in Neural Information Processing Systems*, 33:9841–9850, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Aapo Hyvarinen and Hiroshi Morioka. Nonlinear ica of temporally dependent stationary sources. In *Artificial Intelligence and Statistics*, pp. 460–469. PMLR, 2017.

Aapo Hyvarinen, Hiroaki Sasaki, and Richard Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 859–868. PMLR, 2019.

Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. *arXiv preprint arXiv:1907.07171*, 2019.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33:12104–12114, 2020.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pp. 2207–2217. PMLR, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27, 2014.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pp. 4114–4124. PMLR, 2019.

Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning*, pp. 6348–6359. PMLR, 2020.

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

Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *Advances in neural information processing systems*, 29, 2016.

Weili Nie, Tero Karras, Animesh Garg, Shoubhik Debnath, Anjul Patney, Ankit Patel, and Animashree Anandkumar. Semi-supervised stylegan for disentanglement learning. In *International Conference on Machine Learning*, pp. 7360–7369. PMLR, 2020.

Weili Nie, Arash Vahdat, and Anima Anandkumar. Controllable and compositional generation with latent-space energy-based models. *Advances in Neural Information Processing Systems*, 34:13497–13510, 2021.

Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2085–2094, 2021.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pp. 8821–8831. PMLR, 2021.

Geoffrey Roeder, Luke Metz, and Durk Kingma. On linear identifiability of learned representations. In *International Conference on Machine Learning*, pp. 9030–9039. PMLR, 2021.

Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1532–1540, 2021.

Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9243–9252, 2020.

Rui Shu, Yining Chen, Abhishek Kumar, Stefano Ermon, and Ben Poole. Weakly supervised disentanglement with guarantees. *arXiv preprint arXiv:1910.09772*, 2019.

Peter Sorrenson, Carsten Rother, and Ullrich Kothe. Disentanglement by nonlinear ica with general incompressible-flow networks (gin). In *International Conference on Learning Representations*, 2020.

Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679, 2020.

Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International conference on machine learning*, pp. 9786–9796. PMLR, 2020.

Xiaojiang Yang, Yi Wang, Jiacheng Sun, Xing Zhang, Shifeng Zhang, Zhenguo Li, and Junchi Yan. Nonlinear ica using volume-preserving transformations. In *International Conference on Learning Representations*, 2021.

A PROOF OF IDENTIFIABILITY

The proof of the proposed identifiability theorem is simple and clear. According to $p(\mathrm{y}_i|\mathbf{x}) = p(\mathrm{y}_i|\mathrm{s}_i)$, $q(\mathrm{y}_i|\mathbf{x}) = q(\mathrm{y}_i|\boldsymbol{a}_i^\top \mathbf{z})$, and $p(\mathrm{y}_i|\mathbf{x}) = q(\mathrm{y}_i|\mathbf{x})$, we have $p(\mathrm{y}_i|\mathrm{s}_i) = q(\mathrm{y}_i|\boldsymbol{a}_i^\top \mathbf{z})$. Based on this, as $\phi_i : \mathrm{s}_i \to p(\mathrm{y}_i|\mathrm{s}_i)$ and $\psi_i : \boldsymbol{a}_i^\top \mathbf{z} \to q(\mathrm{y}_i|\boldsymbol{a}_i^\top \mathbf{z})$ are continuous and bijective maps, we can construct a continuous and invertible maps between s_i and $\boldsymbol{a}_i^\top \mathbf{z}$, i.e. $\phi_i^{-1} \circ \psi_i : \boldsymbol{a}_i^\top \mathbf{z} \to \mathrm{s}_i$. Such a map is a strictly monotonic function, because s_i and $\boldsymbol{a}_i^\top \mathbf{z}$ are real-value and one-dimensional variables. Therefore, let $g_i = \phi_i^{-1} \circ \psi_i$, we have $\mathrm{s}_i = g_i(\boldsymbol{a}_i^\top \mathbf{z})$, where g_i is a strictly monotonic function. Note that $\{\boldsymbol{a}_i\}_{i=1}^n$ are orthogonal, hence when we manipulate the value of $\boldsymbol{a}_j^\top \mathbf{z}$, the value of $\boldsymbol{a}_i^\top \mathbf{z}$ is unchanged, as well as s_i, $\forall j \neq i$.

B INTRODUCTION AND DISCUSSION ABOUT NVAE

NVAE is a hierarchical VAE with autoregressive latent variables. Specifically, the latent variables in NVAE are devided into L groups: $\mathbf{z} = (\mathbf{z}_1, \cdots, \mathbf{z}_L)$, and the encoder has the form of $q(\mathbf{z}|\mathbf{x}) = q(\mathbf{z}_1|\mathbf{x})q(\mathbf{z}_2|\mathbf{x}, \mathbf{z}_1) \cdots q(\mathbf{z}_L|\mathbf{x}, \mathbf{z}_{<L})$, while the prior has the form of $p(\mathbf{z}) = p(\mathbf{z}_1)p(\mathbf{z}_2|\mathbf{z}_1) \cdots p(\mathbf{z}_L|\mathbf{z}_{<L})$. As the top groups of latent variables tend to infer the bottom groups, they are encouraged to preserve as much information of inputs $\mathbf{x}$ as possible. Therefore, in NVAE most semantic information of inputs is preserved in the top groups.

We visualize this property of NVAE by plotting the heat map in pixel space of each group, as shown in Figure 7. We can see that the top two groups control most attributes in images, while the following groups control some details, and the bottom groups only control some mild details. Another evidence is that we can well reconstruct images using several top groups, as shown in Figure 6. Therefore, NVAE's top groups latent variables preserve major semantics of inputs. Such a property is friendly to controllable generation, as we can manipulate merely several top groups of latent variables to control the synthesized image and edit realistic images.

Figure 6: Reconstruction by top 12 groups in NVAE with 36 groups. The spare groups are randomly sampled conditioned on $\mathbf{z}_{\leq 12}$.

C INTEGRATION WITH NVAE

The integration of our method and NVAE is non-trivial, because we actually utilize a subtle setting in NVAE. In the inference process of NAVE, the l-th group of latent variables is given by

$$q(\mathbf{z}_l|\mathbf{x}, \mathbf{z}_{<l}) = \mathcal{N}(\mu_l(\mathbf{z}_{<l}) + \Delta\mu_l(\mathbf{x}, \mathbf{z}_{<l}), \sigma_l(\mathbf{z}_{<l}) \cdot \Delta\sigma_l(\mathbf{x}, \mathbf{z}_{<l})), \tag{9}$$

where $\Delta\mu_l$ and $\Delta\sigma_l$ are the relative location and scale of the approximate posterior with respect to the prior $\mathcal{N}(\mu_l(\mathbf{z}_{<l}), \sigma_l(\mathbf{z}_{<l}))$. We note that since $\mathbf{z}_{<l}$ have preserved some information of $\mathbf{x}$, the additional information for $\mathbf{z}_l$ is given by $\Delta\mu_l(\mathbf{x}, \mathbf{z}_{<l})$ and $\Delta\sigma_l(\mathbf{x}, \mathbf{z}_{<l})$. This is obvious, since $\mu_l(\mathbf{z}_{<l})$ and $\sigma_l(\mathbf{z}_{<l})$ are functions of $\mathbf{z}_{<l}$ and hence cannot preserve additional information from $\mathbf{x}$.

Therefore, to disentangle the information of different groups, we actually use samples from $\mathcal{N}(\Delta\mu_l(\mathbf{x}, \mathbf{z}_{<l}), \Delta\sigma_l(\mathbf{x}, \mathbf{z}_{<l}))$ as the input of normalizing flows, i.e. $\mathbf{w}_l$ in Figure 2. We empirically find this scheme more stable and have better performance than directly using the original groups of NVAE.

D A VOLUME-PRESERVING VERSION OF REAL NVP

The real NVP is a normalizing flow architecture consists of several coupling layers. A coupling layer with inputs $\mathbf{x} \in \mathbb{R}^{2d}$ is

$$\begin{aligned} \mathbf{y}_{1:d} &= \mathbf{x}_{1:d} \\ \mathbf{y}_{d+1:2d} &= \mathbf{x}_{d+1:2d} \odot \exp(s(\mathbf{x}_{1:d})) + t(\mathbf{x}_{1:d}), \end{aligned} \tag{10}$$

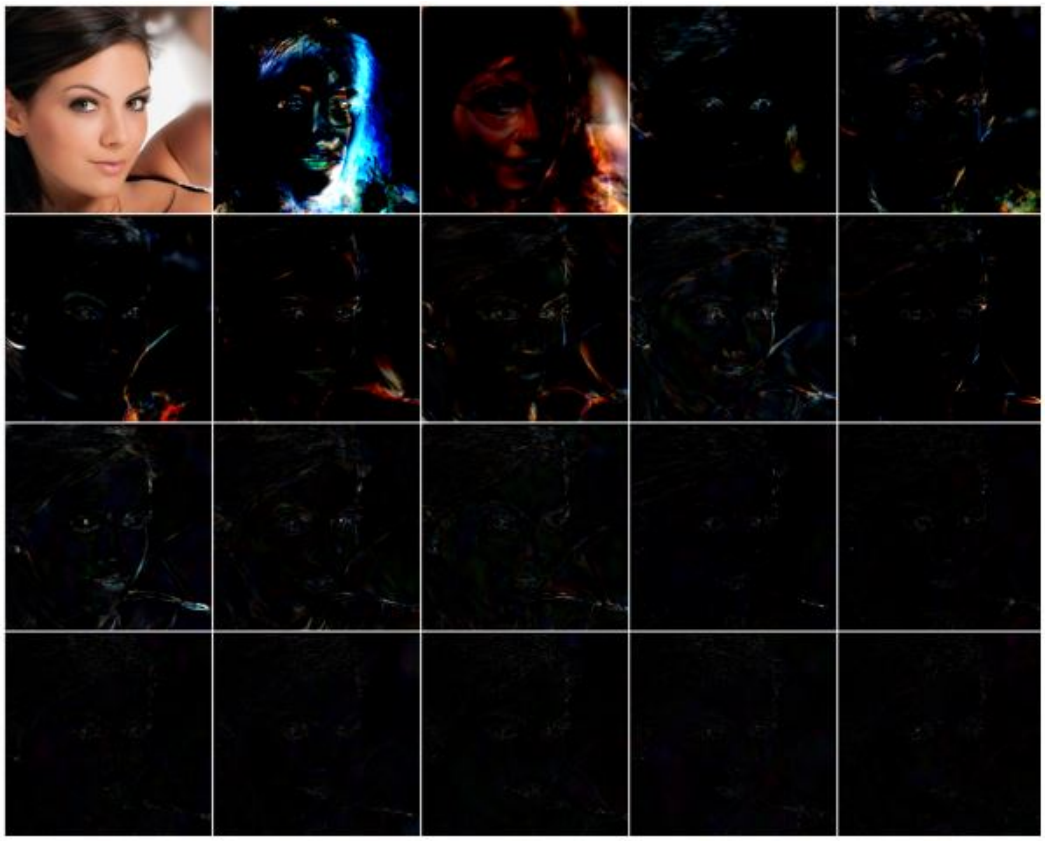


Figure 7: Heat maps of top 19 groups by a pre-trained NVAE with 36 groups. The first image is a raw face image, and the residual images from left to right and from top to bottom are the heat maps of $\mathbf{z}_1 \sim \mathbf{z}_{19}$.

where s and t are scale and translation functions from $\mathbb{R}^d$ to $\mathbb{R}^d$. The coupling layer above is non-volume-preserving, and the volume is

$$\prod_{i=1}^{d} \exp(s(\mathbf{x}_{1:d})_i) = \exp(\sum_{i=1}^{d} s(\mathbf{x}_{1:d})_i). \tag{11}$$

We note that if we set the sum $\sum_{i=1}^{d} s(\mathbf{x}_{1:d})_i$ as zero, then the coupling layer will be volume-preserving. Therefore, we let the coupling layers becomes

$$\begin{aligned} \mathbf{y}_{1:d} &= \mathbf{x}_{1:d} \\ \mathbf{y}_{d+1:2d} &= \mathbf{x}_{d+1:2d} \odot \exp(s(\mathbf{x}_{1:d}) - m) + t(\mathbf{x}_{1:d}), \end{aligned} \tag{12}$$

where $m = \frac{1}{d}\sum_{i=1}^{d} s(\mathbf{x}_{1:d})_i$. Such a modification does not harm the reversibility, and works well in experiments.

E Model Architectures and Hyper-parameters

Model architectures The sizes of top 12 groups of the pre-trained NVAE on FFHQ are 4×(20, 8, 8), 4×(20,16,16), and 4×(20,32,32). We divide each group along width and height dimensions into 4 blocks, and then stack them along channel dimension. Therefore, the inputs of normalizing flows are 12 groups with sizes of 4×(80, 4, 4), 4×(80,8,8), and 4×(80,16,16). The normalizing flows for different groups have the same setting. Each normalizing flow consists of 12 layers, and each layer consists of an ActNorm layer (Kingma & Dhariwal, 2018), a volume-preserving coupling layer and a Shuffle along the channel. In each coupling layer, the inputs are divided along the channel dimension, and the scale and translation functions are parameterized by ResNet blocks.

The architecture of each linear model is simply a linear layer with bia. The existence of bias is actually permitted in our theory, hence we do not set it as zero.

Hyper-parameters We use an AdaMax (Kingma & Ba, 2014) optimizer with learning rate 0.0005 for optimization. We train the model LAO-NVAE on 4 NVIDIA V100 GPUs with batch size 15 on each GPU, and evaluate it on one single NVIDIA V100 GPU. The number of training epochs is merely 5.