Text-Free Learning of a Natural Language Interface for Pretrained Face Generators

Xiaodan Du¹, Raymond A. Yeh²*, Nicholas Kolkin³, Eli Shechtman³, Greg Shakhnarovich¹

¹TTI-Chicago{xdu,gregory}@ttic.edu          ²Purdue University{rayyeh}@purdue.edu
³Adobe Research{kolkin,elishe}@adobe.com

Abstract

We propose *Fast text2StyleGAN*, a natural language interface that adapts pre-trained GANs for text-guided human face synthesis. Leveraging the recent advances in Contrastive Language-Image Pre-training (CLIP), no text data is required during training. *Fast text2StyleGAN* is formulated as a conditional variational autoencoder (CVAE) that provides extra control and diversity to the generated images at test time. Our model does not require re-training or fine-tuning of the GANs or CLIP when encountering new text prompts. In contrast to prior work, we do not rely on optimization at test time, making our method orders of magnitude faster than prior work. Empirically, on FFHQ dataset, our method offers faster and more accurate generation of images from natural language descriptions with varying levels of detail compared to prior work.

1. Introduction

Modern generative models like StyleGANv2 can map a random latent code to a photorealistic image, especially in the human face domain. In many application scenarios, to be useful in practice, the process of generation must allow for explicit control of the content. Previous work explored modes of control based on analysis of a pre-trained model and identification of domain-specific axes of variation in the GAN latent space (e.g., age or expression for faces). Then, by changing the values of the latent variables along these axes, one can modify the generated image. These techniques are helpful for editing an image, but for complete de-novo synthesis they are inefficient.

To address this issue, we pursue a more natural control mechanism: via explicit natural language description of the desired properties of the generated content. In other words, we aim to generate *human face domain-specific* photorealistic images purely from natural language descriptions; See Fig. 1. This task is often formulated as learning a con-
A key technical difficulty, however, is that CLIP embeddings of images do not necessarily capture all the perceptually important aspects of the image. For instance, we find that while the coarse “demographics” of the face are well captured by CLIP, the pose (orientation) and background are not. In addition, a description may omit certain visual attributes (i.e., it may refer to the desired expression and age, but not to the hair style). We would like to be able to achieve diversity in these aspects that are under-specified by the CLIP embedding of the description. To this end we propose a novel two-channel encoder for images. One channel is the (pre-trained, fixed) CLIP embedding. The other channel is a conditional variational autoencoder (CVAE), conditioned on the CLIP embedding. The CVAE channel, intuitively, learns to complement the information captured by CLIP, to allow better image reconstruction at training time, and to provide a source of diversity (and possibly additional control) in the generated images at inference time.

A remaining issue is the discrepancy of CLIP text vs. image embedding distribution, and the lack of diversity within attributes that are specified by the description. Thus, as an additional contribution, we propose a non-parametric prior for mapping text embedding to a distribution of image embeddings, without training.

To date most successful GAN models are trained for one domain at a time, and they have achieved particular success on the domain of human faces, where they can produce diverse images of synthetic faces at very high resolutions. This, and the importance of faces in applications (e.g., syn-
thetica stock photos) motivates us to focus on faces as our main experimental domain. Empirically, we demonstrate that our approach generates high quality images and more faithfulness to text prompt than prior work. Importantly, our method can generate images that follow complex language description with different level of details. Additionally, our approach is very fast and requires only a single forward pass to generate an image, hence we name it **Fast text2StyleGAN**.

**Our contributions are as follows:**

- We propose **Fast text2StyleGAN**, a natural language interface between a joint image/text embedding model and a GAN. We introduce a novel two-channel architecture, proposing a CLIP-conditional variational autoencoder for training this interface.
- **Fast text2StyleGAN** generates high resolution images from text dramatically (orders of magnitude) faster than prior work.
- Our method supports arbitrary text inputs at inference time; *no re-training or fine-tuning on paired image and text prompt is necessary.*
- We propose a non-parametric text-to-image embedding prior which significantly improves diversity and accuracy.

## 2. Related Work

**CLIP.** By leveraging a massive dataset of 400 million text-image pairs, CLIP [21] uses contrastive loss to learn a multi-modal embedding of images and text into a shared “semantic space”. Associated image-text pairs are mapped to vectors with higher cosine similarity than random pairs. These embeddings have proven to be broadly applicable, becoming a vital component in a variety of image generation and manipulation frameworks [9, 20, 22].

**Unconditional image generation.** While unconditional image generation remains an open problem, four leading paradigms have emerged in recent years: autoregressive models [3, 8, 35], VAE models [24, 34], diffusion models [12, 29], and GANs [10, 19]. Of these GANs are perhaps the most popular, owing to samples’ high visual quality, fast inference speed, and the ease of performing impressive image manipulations using their fairly compact latent space.

We adopt StyleGANv2 [14], which at inference time maps a Gaussian random input $z$ to a transformed latent vector $w$, then decodes $w$ into an image.

**Image manipulation in latent space.** A great deal of recent work has focused on “inverting” a generator, typically a GAN, *i.e.*, to find a point in the latent space that, when fed to the generator, generates a given image $[1, 2, 25, 33]$, then performing image manipulations by altering the latent code. A major line of work in this area focuses on discovering dimensions of the latent space which control certain predetermined characteristics of the image (most commonly, attributes of face images, such as age, gender, expression, hairstyle *etc.* [27, 36, 41]). An input image is “inverted”, i.e., mapped to the latent space, the latent representation is moved along relevant attribute-specific dimensions, and fed back to the generator which produces an image with the desired attributes modified. This approach has simplified complex image manipulations which only a few years ago would have required time consuming effort of experts. However, these methods are limited by requiring controls in latent space to be found before inference, and lack the flexibility and intuitive control of natural language.

**Text-driven image generation.** Image manipulation is just one aspect of control which increases the utility of generative models, and conditional de-novo generation is another way generative models can become a useful tool. This second regime, specifically text-driven conditional generation, is our focus. GANs [16, 32, 38, 39] have been widely used for text-driven image synthesis tasks. However, unlike most work in this area, **Fast text2StyleGAN** does not rely on additional labels such as captions, bounding boxes, or segmentation maps; instead using the CLIP embedding of images in the training set as a proxy for paired captions. Two recent works operate in a similar regime, TediGAN [37] and StyleCLIP [20], making them a good point of comparison with **Fast text2StyleGAN**. We conduct qualitative and quantitative comparisons with these works.

Recently several works have made great strides in generating high quality images conditioned on natural language. DALL-E [22, 23], CogView [6, 7], ImaGen [26], *etc.* [17, 30], have drawn a lot of attention because of their ability to synthesize diverse and high-quality images from long and complex prompts. However, they usually require large amount of paired training data and computing resources. In this work, we do not compare to these models because: (a) they require text data during training; and/or (b) they are trained on much larger domain than human face so it is hard to compare quantitatively; especially when most of them are not open-sourced.

## 3. Approach

The overview of our approach is shown in Fig. 2. We rely on two pre-trained (and frozen) components: the StyleGANv2 image generator $G$, and the CLIP image/text embedding networks, respectively $C_I$ and $C_T$. The end goal is to map $C_T$(prompt) to the latent space of $G$. We do so without any availability of detailed prompt/image pairs. Leveraging the multi-modal nature of CLIP, we formulate the problem as a conditional variational autoencoder CVAE [28] in Sec. 3.1. Architecture and training details are discussed in Sec. 3.2 and Sec. 3.3.

At train time, an image is encoded into two parts: (a) a CLIP image embedding for conditioning both the encoder and decoder in CVAE; (b) a latent vector that encodes the variation of the image to follow a Gaussian distribution. Given these two vectors, the decoder’s aims to
reconstruct the image. At test time, our goal is to perform text conditional generation. Given a text input, we extract a text embedding using CLIP. Due to the distribution gap between CLIP’s text/image embedding, we construct a non-parametric generative model of CLIP image embedding given CLIP text embedding; see Sec. 3.4. Using this model, we sample a corresponding CLIP image embedding to condition the decoder. Finally, we sample the latent vector from a Gaussian distribution following a standard CVAE. Importantly, our approach requires a single forward pass through the decoder at test time leading to a short generation time, hence we named our model *Fast text2StyleGAN*.

### 3.1. Interface learning with CVAE

Let $x$ denote an image. Our goal can be viewed as modeling a conditional distribution $p(x|c)$, where $c$ is a conditioning context. At training time, $c$ is provided by the (deterministic) CLIP-image embedding $c(x) = C_I(x)$. CVAE incorporates both inference (mapping from image to the latent) and generation (mapping from latent to image). Given an input $x$ and a context $c$, CVAE maps them to two components: $f_{\text{enc}}^\mu$ and $f_{\text{enc}}^{\sigma^2}$ that models the mean and (per-dimension) variance of the Gaussian distribution for a latent vector $z$. To enable sampling, the distribution of the latent is required to match the prior:

$$q_\theta(z|x, c) = \mathcal{N}(f_{\text{enc}}^\mu(x, c), f_{\text{enc}}^{\sigma^2}(x, c)) \quad (1)$$

$$p_0(x|z, c) = \mathcal{N}(f_{\text{dec}}^\mu(z, c), \sigma I) \quad (2)$$

$$p(z|c) = \mathcal{N}(0, I). \quad (3)$$

Typically, the prior is chosen to follow a standard multivariate Gaussian. A standard CVAE is trained on an image set $D = \{x\}$ by maximizing the evidence lower bound (ELBO):

$$\sum_{(x, c(x)) \in D} \log(p_0(x|z, c)) - D_{KL}(q_\theta(z|x, c)||p(z|c)) \quad (4)$$

In practice, additional loss functions were introduced to improve image generation quality; we defer discussion of these to Sec. 3.3. We will next describe the model’s architecture.

### 3.2. CVAE architecture

The overall architecture of *Fast text2StyleGAN* can be viewed as a CVAE shown in Fig. 2. The encoder parameterizes the distribution $q_\theta$, where as the decoder parameterizes the distribution $p_0$. At test-time, given the text description $t$, a random vector drawn from standard Gaussian distribution $\mathcal{N}(0, I)$ and the CLIP embedding of the text prompt $c = C_T(t)$ are used as inputs. During inference time, an image, instead of a random noise, can be used as extra guidance in addition to text prompt.

**Encoder $q_\phi(z|x, c)$ architecture.** Our CVAE’s encoder consists of two branches in parallel. Given an image as input, the first branch extracts the CLIP embedding via CLIP’s pre-trained encoder, $c = C_I(x)$. Another branch in parallel is a standard convolutional neural network containing five 3x3 Conv2d layers and a fully-connected layer. Let its output be $e = \text{ConvNet}(x)$.

The two output vectors are concatenated, and mapped by a four-layer network MLP-E to output the the mean and variance of the latent space:

$$[f_{\text{enc}}^\mu, f_{\text{enc}}^{\sigma^2}] = \text{MLP-E}([e, c]). \quad (5)$$

To sample latent vector $z$ from the posterior distribution $q_\phi(z|x, c)$ and back-propagate through the samples for training, we use the reparameterization trick [15].

**Decoder $p_\theta(x|z, c)$ architecture.** The CLIP embedding of the input $c$ is concatenated with $z$ and sent into a decoder network MLP-D, with architecture identical to that of MLP-E. MLP-D outputs vector $\Delta$, interpreted as the offset from the average latent code $\bar{w}$ of StyleGAN’s latent. We follow definition of $\bar{w}$ by Richardson et al. [25]. Finally, $\Delta$ is added to $\bar{w}$ and passed to StyleGANv2, $G$, to produce the output image

$$\hat{x} = G(\bar{w} + \Delta), \quad \text{where } \Delta = \text{MLP-D}([z, c]). \quad (6)$$

### 3.3. Loss functions

Inspired by Richardson et al. [25] and Tov et al. [33], we replaced the pixel-wise $\ell_2$-loss resulting from the Gaussian assumption (Eq. (2)) with the Learned Perceptual Image Patch Similarity (LPIPS) loss [40] to ensure high perceptual similarity between the deep representations of input and output images.

Next, to encourage our model to learn input and output images that are similar in CLIP space (to eventually drive text generation) we introduce a CLIP cycle loss:

$$\mathcal{L}_{\text{CLIP cycle}}(x) = 1 - \text{Sim}_\text{cos}(C_I(x), C_I(\hat{x})), \quad (6)$$

where $\text{Sim}_\text{cos}$ represents the cosine similarity between the two embeddings and $C_I$ is the image encoder of CLIP. Recall (Sec. 3.2) that during training $\Delta$ is the output of MLP-D, computed from both the CLIP embedding of $x$ and from $z$, sampled from the output of the variational encoder.

Finally, following Richardson et al. [25], we included a $w$ normalization loss to prevent the predicted StyleGAN latent vector $w$ from straying too much from the distribution:

$$\mathcal{L}_{\text{w-norm}}(x) = ||\Delta||^2. \quad (7)$$

To train the model, we freeze model parameters of CLIP and StyleGANv2 and only perform gradient updates on MLP-E and MLP-D to minimize a weighted combination of the aforementioned losses.
3.4. Non-parametric sampling for text-to-image embedding generation

At test-time, we are given a text $t$ of the desired image. Naively, one can directly use this text’s CLIP embedding, $C_T(t)$, in-place of an image embedding for conditioning the decoder. However, this CLIP embedding only provides a deterministic descriptor for the content, with the CVAE Gaussian latent sample $z$ being only source of randomness.

This poses two problems. First, while CLIP’s training objective is to maximize cosine similarity between associated text/image pairs, there is a significant discrepancy between image and text embeddings, yielding a discrepancy between training and inference regime, potentially reducing generation accuracy. Second, even a relatively detailed description of an image necessarily leaves a lot unspcified; while sampling $z$ conditional on $C_T(t)$ provides some of these complementary details, we find that the resulting diversity is too limited, as seen in Fig. 5.

A similar observation in concurrent text-to-image generation work, notably in DALL-E2 [22], where the solution is to train a 1B parameter model that maps CLIP text embedding $C_T(t)$ to a distribution of CLIP image embeddings $C_J(x)$ for images that match $t$. This requires a dataset of image/text pairs, which in DALL-E2 count in the hundreds of millions. Our approach is aimed at domains where we do not have any paired image/text data. Thus, we propose a very simple, and as we show very effective, non-parametric conditional generative model that does not require separate training and uses domain images only.

At test time, we retrieve $K$ training images whose CLIP image embeddings, $c_1, \ldots, c_K$ are closest in cosine distance to the CLIP text embedding of the prompt, $C_T(t)$. Then we sample a random subset of $M < K$ of these nearest neighbors – without loss of generality, let these be $c_1, \ldots, c_M$. We compute a random convex combination $c^* = \sum_{j=1}^M \alpha_j c_j$, with $\alpha_j$s sampled from a Dirichlet distribution. The resulting vector $c^*$, instead of $C_T(t)$, is passed to the decoder at test time. We observe the non-parametric model to yield better visual quality, presumably because $c$ at test time is closer to the $c$ used at training time. The stochasticity in the process (random $M$ neighbors and random convex combination of their embeddings) also tends to provide more meaningful and significant diversity. Visual comparisons can be found in Fig. 5.

4. Experiments

We conduct experiments using the Flickr-Faces-HQ (FFHQ) [13] dataset. FFHQ contains 70,000 diverse 1024 × 1024 high-resolution images of human faces, which StyleGANv2 is also trained on.

Evaluation metrics. For empirical validation, we consider three aspects, image quality, image diversity, and correctness. Following the protocol of prior work [18], we use their method to generate 400 captions. All models randomly generate 25 images for each caption. The 10,000 generated images for each model are used for comparison.

For quality, we have included numerous non-curated examples (also in the Appendix) generate from our approach and baseline. We refrain from using FID [11] for evaluation as the metric is designed for unconditional generation. We provide a detailed discussion and evidence on why FID is not suitable for conditional generation in the Appendix.

For correctness, we report retrieval accuracy which measures the consistency with input text. For an output image $\hat{x}$ with text guidance $t$, 99 images in FFHQ dataset are randomly chosen as negative samples. CLIP is used to measure the cosine similarity between $t$ and the 100 images. Retrieval accuracy is the average over all text prompts and generated images. This metric is introduced in a prior work [18].

For diversity, we report identity diversity (ID Div.) Recall for each of the 400 captions we generate 25 images resulting in $\binom{25}{2} = 300$ image pairs. For each image pairs, we use the pre-trained ArcFace [5] model to obtain feature vectors for both images. We then compute the cosine similarity between these two vectors. We denote one minus this similarity as the identity diversity. Finally, we average the identity diversity over all image pairs and all captions. As ArcFace is trained to perform face recognition, features extracted from ArcFace focus on the facial attributes. I.e., a high ID Div. indicates that the two face images are not of the same identity, hence, more diverse.

Baselines. We compare our approach with TediGAN [37] and StyleCLIP [20] using their official publicly released implementation. We select these two methods because they do not require text data during training, hence a fair comparison with our approach. We briefly review each of the methods.

StyleCLIP provides three ways for doing image generation/manipulation: Optimization, Mapper and Global directions. We compare to the Optimization method because only this method works for open-world captions without re-training or instance-level hyperparameter selection. The
Optimization method uses CLIP to guide the iterative modifications of images in StyleGAN’s latent space. StyleCLIP Optimization provides two modes in their official Colab notebook: free generation and edit. The “free generation” mode is not mentioned in the paper as StyleCLIP focuses on editing. We include this mode, as it is the closest to our task and achieves the best retrieval accuracy among all baselines.

- **StyleCLIP (g).** The free generation (g) mode can be found in the drop-down menu of “experiment_mode” in the Colab notebook. It initializes the latent vector with the mean latent vector of StyleGAN’s latent space and gradually minimizes the CLIP loss between the output and the text. We turn “stylespace” flag on as it provides best possible visual quality.

- **StyleCLIP (m).** The other mode of StyleCLIP Optimization is image manipulation (m) which first generates a face using StyleGANv2 and then manipulates this specific image according to the caption. We use the default hyperparameters in the GitHub code base.

TediGAN also does instance-level optimization according to given text prompt, providing two modes:

- **TediGAN (g).** This is the free generation (g) mode of TediGAN. TediGAN contains a text encoder that is trained to project text to the latent space of StyleGAN. It first samples a latent code in StyleGAN’s latent space and maps the text to the same latent space with the text encoder. Certain attributes are altered while minimizing the distance between the image and text latent vectors.

- **TediGAN (m).** This is the image manipulation (m) mode of TediGAN. It requires an input image to start with. We use StyleGANv2 to generate random faces as the inputs to TediGAN (m). A GAN inversion model is used to obtain the latent code corresponding to the input image. The rest is the same as for TediGAN (g).

We also consider two variants of our method, namely,

- **Ours (Pt).** This variant do not use the proposed non-parametric sampling scheme and directly uses a point (Pt) estimate, i.e., CLIP’s text embedding is directly passed to the decoder which was trained on image embeddings.

- **Ours (Img).** This variant samples z from the encoder’s output distribution conditioned on another image, instead of a standard Gaussian. The image can be manually selected by the user or randomly selected to provide ex-
Figure 4. Results of optional image guidance. The left-most column shows images used as input to the CVAE, while the CLIP embedding comes from a different prompt (see Appendix for prompts used) for each of the rest of the columns. We find the pose and the background color are usually captured by the variational encoder.

Figure 5. Comparison of Ours (Pt) and Ours. The non-parametric conditional sampling in Ours significantly improves diversity and, often, visual realism.

4.1. Quantitative Results

Quantitative results are reported in Tab. 1. As can be seen, Ours achieves the highest retrieval accuracy out of all the methods, i.e., the generated images more closely matches the given text prompt. StyleCLIP (g) also achieved high retrieval accuracy, but with much lower diversity. Next, StyleCLIP (m), TediGAN (g) and TediGAN (m) mostly generated images that do not follow the prompt. As expected they achieve the highest diversity as they essentially perform unconditional generation. We compute $ID_{Div}$ among random unconditional StyleGANv2 generations and get 0.961. The fact that some of the baselines have $ID_{Div}$ values close to that is a support for our claim. We’ll discuss this point in more details in Sec. 4.2. When comparing among our methods, we observe that using our full model achieves the best result demonstrating the effectiveness of the proposed nonparametric sampling scheme.

We also report the time for sampling a single image on a single QUADRO RTX 6000 GPU. Our approach only requires a single forward pass through the network, and thus is approximately two orders of magnitude faster than prior, iterative optimization based work.

4.2. Qualitative Results

In Fig. 3, we qualitatively compare our proposed approach with TediGAN and StyleCLIP. We encourage readers to zoom-in to around 500% scale to better inspect the details. Ours provides images that are consistent with the text, diverse and with high visual quality. While StyleCLIP (g) captures a large portion of the content in text prompts, its visual quality is poor and lacks diversity among images for given text. On the other hand, StyleCLIP (m) produces highly diverse results, but often the images are unrelated to the given caption. Its performance depends highly on the original face it starts with. For example, the first example of prompt (b) is satisfying because it is initialized with a woman with light-color hair. When the initial image is too “far away” from the caption, StyleCLIP (m) struggles and often only changes one attribute; E.g., grey hair in the third image of prompt (a) or blond hair in the third image of prompt (b). We have confirmed with StyleCLIP’s authors that the visual quality for generation is as expected.

TediGAN tends to generates outputs that are unrelated to the captions except in very rare cases. Similar to StyleCLIP, for optimization based methods, it depends heavily on the initializing image. It is more sensitive to certain attributes such as “old”. For example, TediGAN (g) adds wrinkles to the first output of prompt (a) and TediGAN (m) does similar to the first output of prompt (c). Others [17, 30] also report similar unsatisfactory performance of TediGAN.

In Fig. 4, we illustrate the use of image guidance (Ours (Img)) for controlling the pose and background of the image generation, beyond the text prompt. Each column is conditioned with a different text prompt and given the image guidance of the image on the very left. The background and pose of the person follow the guidance image. This is consistent with our hypothesis that $z$, and not $c$, captures background and pose information.

Pre-trained model available at github.com/openai/CLIP
We propose a method for learning an interface between pre-trained text embeddings (CLIP) and a pre-trained image generator (StyleGANv2). Once trained, our interface enables fast conditional image synthesis from natural language. Training our method requires only an unlabeled image set, without associated text information. Our introduced CVAE formulation allows modeling diverse visual attributes that are not specified by the text prompt. We also proposed a test time procedure scheme based on non-parametric sampling, which improves quality and robustness of the generation process. This is done by modelling a non-parametric distribution of CLIP’s image embedding given text embedding. Empirically, we demonstrate that our Fast text2StyleGAN produces high quality images that more faithfully follow the text prompt and is significantly faster in inference speed compared to recent baselines.
[34] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Proc. NeurIPS*, 2017.

[35] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *Proc. ICML*, 2016.

[36] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for StyleGAN image generation. In *Proc. CVPR*, 2021.

[37] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. TediGAN: Text-guided diverse face image generation and manipulation. In *Proc. CVPR*, 2021.

[38] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In *Proc. CVPR*, 2018.

[39] Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Cross-modal contrastive learning for text-to-image generation. In *Proc. CVPR*, 2021.

[40] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. CVPR*, 2018.

[41] Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. Barbershop: GAN-based image compositing using segmentation masks. *arXiv preprint arXiv:2106.01505*, 2021.
Appendix

This appendix is organized as follows:

- In Sec. A, we explain why FID is not suitable for evaluating conditional image generation tasks and provide evidence to support our claim.
- In Sec. B, we provide more qualitative comparison results against the baselines.
- In Sec. C, we provide examples of Ours (Pt) generates images faithful to prompts when Ours fail, due to the reason combinations of certain attributes not present in the original data.
- In Sec. D, we provide additional results generated by Ours using Fast text2StyleGAN.
- In Sec. E, we examine the nearest neighbors of our generated samples in FFHQ dataset, showing our model is able to generate novel faces rather than pure memorization or simple hallucination.
- In Sec. F, we provide text prompts used to generate output images in Fig. 4 in the paper.

A. Discussion on FID

**Definition.** Fréchet Inception Distance (FID), proposed by Heusel et al. [11], is an evaluation metric for image generation model. Given a set of generated images and another set of real images, FID measures the squared Wasserstein metric between the two sets modeled as a multivariate Gaussian distribution with features extracted from Inception-v3 [31] (trained on ImageNet [4]). Formally, Heusel et al. [11] define FID as:

\[
\|\mu_g - \mu_r\|_2^2 + \left(\Sigma_g + \Sigma_r - 2\left(\Sigma_g \Sigma_r\right)^{1/2}\right)
\]

where \(\mu_{g/r}\) and \(\Sigma_{g/r}\) denotes the mean and covariance matrix estimate from the generated/real data in the feature space, respectively. To extract features, FID uses the last pooling layer of Inception-v3 pre-trained on ImageNet. Better (lower) FID means the two distributions are closer in feature space.

**FID is not suitable for conditional generation.** The FFHQ data is unconditional, i.e., it does not come with text captions. On the other hand, the generated images from the models are conditioned on the 400 text prompts. In other words, unless the text covers the entire possible faces of FFHQ, the two distributions are simply different distributions. Hence, FID is not a suitable evaluation metric, as following the text prompts could potentially penalize the metric. We illustrate this point in Fig. A1, where we show images generated (for a single prompt) and corresponding FID, from the unconditional StyleGANv2, baselines and ours. The result reveals the shortcomings of FID: (a) approaches that simply ignores the text achieves low FID and (b) FID does not necessary correlates with the perceived image quality.

B. More qualitative comparisons with baselines

In Fig. A2, we provide additional visualization of our approach compared to baselines.

C. Ours (Pt) vs. Ours for out-of-distribution prompts

In Fig. A3, we show results on out-of-distribution prompts, i.e., the caption does not resemble any image that is in the FFHQ dataset. In such cases, we observe that Ours (Pt) more acutely follows the prompt but at the cost of diversity.

D. Additional results generated by Ours

In Fig. A5, Fig. A6 and Fig. A7, we show more results with Ours using Fast text2StyleGAN.

E. Nearest neighbors of generated images in FFHQ dataset

In Fig. A4 We show the nearest FFHQ neighbors of images generated by Ours. The left column contains the outputs from the last caption in Fig. A7. We use CLIP to find their top 5 nearest neighbors from FFHQ dataset in CLIP feature space. The neighbors are on the same row and to the right of the output image. The neighbors are ranked from left to right with top 1 nearest neighbors on the left. As can be seen, the outputs are not duplicates of images from FFHQ but novel faces with novel identities.

F. Text prompts of Fig. 4 in the main body

From left to right:

1. “A bald Asian man with beard.”
2. “A photo of a smiling Asian young man with curly hair.”
3. “A blond woman with wavy long hair wearing makeup.”
Figure A1. FID and sample images from the same caption given on top. The eight images for each method are uncurated. In the first row, we include FID and random unconditional outputs from the vanilla StyleGANv2 for reference. As can be seen, FID cannot correctly reflect performance of models for conditional generation.
(a) An Asian girl with purple hair.  
(b) A white person with blue eyes and red hair slightly opens mouth.  
(c) A middle-aged Black woman with heavy makeup.

Figure A2. More qualitative comparison with image manipulation (m) and generation (g) baselines.

Sad man with closed eyes and lipstick.

A female with no eyebrows.

Kid with large ears and no hair.

Figure A3. Ours (Pt) vs. Ours for out-of-distribution prompts. When it is hard to find exemplars in FFHQ that match the out-of-distribution text prompts, Ours (Pt) gives more accurate outputs while sacrificing some diversity.
Figure A4. Nearest neighbors of generated images in FFHQ dataset, drawn by computing CLIP similarity, on the text prompt “A person with a big forehead”. Our model creates novel faces with novel identities each time even given the same text prompt.
A Black female wearing bright lipstick.

A chubby person on dark background.

A smiling woman with curly blond hair wearing sunglasses.

Figure A5. Additional results for Ours. The text prompt on top is used to generate the eight images below it.
Figure A6. Additional results for Ours. The text prompt on top is used to generate the eight images below it.
A young woman with colorful hair.

Figure A7. Additional results for Ours. The text prompt on top is used to generate the eight images below it.