Interactive Image Generation with Natural-Language Feedback

Yufan Zhou[1], Ruiyi Zhang [2], Jiuxiang Gu[2], Chris Tensmeyer[2], Tong Yu[2], Changyou Chen[1], Jinhui Xu[1], Tong Sun[2]

[1]State University of New York at Buffalo [2] Adobe Research
{yufanzho, changyou, jinhui}@buffalo.edu {ruizhang, jigu, tensmeye, tyu,tsun}@adobe.com

Abstract

Using natural-language feedback to guide image generation and manipulation can greatly lower the required efforts and skills. This topic has received increased attention in recent years through refinement of Generative Adversarial Networks (GANs); however, most existing works are limited to single-round interaction, which is not reflective of real world interactive image editing workflows. Furthermore, previous works dealing with multi-round scenarios are limited to predefined feedback sequences, which is also impractical. In this paper, we propose a novel framework for Text-based interactive image generation and manipulation (TiGAN) that responds to users' natural-language feedback. TiGAN utilizes the powerful pre-trained CLIP model to understand users' natural-language feedback and exploits contrastive learning for a better text-to-image mapping. To maintain the image consistency during interactions, TiGAN generates intermediate feature vectors aligned with the feedback and selectively feeds these vectors to our proposed generative model. Empirical results on several datasets show that TiGAN improves both interaction efficiency and image quality while better avoids undesirable image manipulation during interactions.

Introduction

Text-to-image generation and text-guided image manipulation are important research topics, which have demonstrated great application potentials due to the flexibility and usability of natural language. Compared to traditional image editing software that require users to learn complex tools, language driven methods can be more intuitive for novice users. One main challenge of text-to-image generation/manipulation is that images are 2D arrays of pixels, while natural language expressions are sequences of words with no clear mapping between them. While existing works (Xu et al. 2018; Zhang et al. 2021; Xia et al. 2021; Patashnik et al. 2021) have proposed useful new models and loss functions, they share the limitation that they focus on single-round tasks, i.e., these methods generate or manipulate an image only in the context of a single natural language instruction. Such a restriction limits the applicability of the models for real-use cases, as a user may want to continually refine an image until satisfactory. While such models could be naively applied recursively, at each round, the model would be oblivious to previously given feedback leading to high likelihood that the model interferes with previous edits.

There also exist some works that sequentially generate images following different instructions (El-Nouby et al. 2019; Fu et al. 2020). However, these methods are not fully interactive and less practical. For example, models in (El-Nouby et al. 2019; Fu et al. 2020) are trained on predefined sequences of natural language instructions, while the instructions are independent of generated images and follows a predefined order. However, when a real user interacts with the model, the natural-language feedback is unpredictable and depends on generated image in each round. Thus, the use of predefined sequences is impractical for real-world interactive applications.

In this work, we focus on a new problem of interactive image generation, which generalizes text-to-image generation and text-guided image manipulation to the multiple round setting. It is a natural extension to existing single round methods, and our goal is to generate desired images with fewer interactions. Consequently, we address these two critical challenges: (*i*) how to learn a better text-to-image mapping; (*ii*) how to avoid undesirable image manipulations throughout the interaction session. A better text-to-image mapping would improve overall image quality and improve how well the image agrees with the text. An undesirable image manipulation would occur if the model accidentally changes an aspect of the image that the user already specified. For instance, assume the user requests the model to "generate a man's face" and then issues the command "make the hair long". We are expecting two generated images: an image of a man and an image of a man with long hair for this two-round example. Receiving an image of a man and an image of a woman with long hair is a failure case which also satisfies the user's requirement at every round. Since user has requested the image to be of a man in the first round, the model should not change that aspect of the image in later manipulations unless the user explicitly says otherwise.

To handle the aforementioned challenges, we propose Text-based Interactive image generation and manipulation (TiGAN). Different from existing works that focus on complicated architecture designs, we tackle the problem by directly adapting powerful unconditional generative models into our model for text-conditional generation. Specifically,

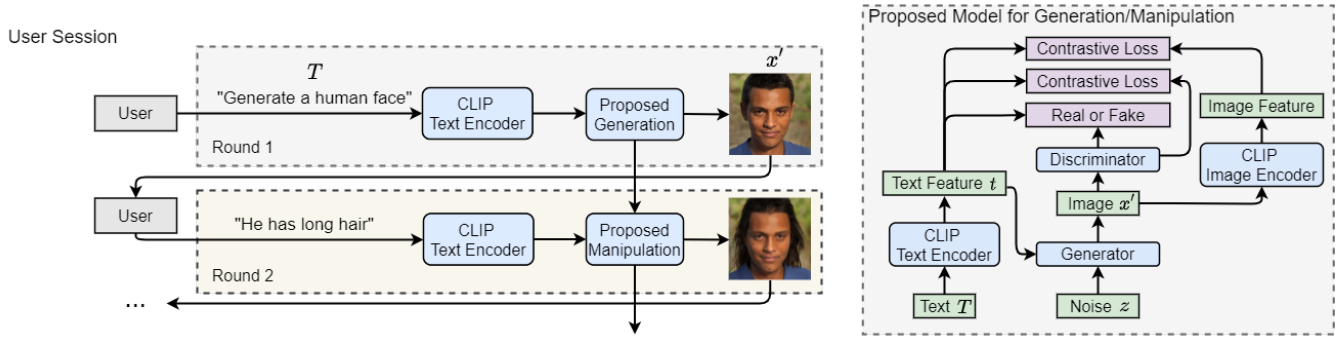

Figure 1: Overview of our interactive image generation. A user starts a session and keeps giving natural-language feedback to the generative model until they are satisfied with the generated image. We propose TiGAN with specifically designed contrastive losses that encourage a better text-to-image mapping, which is used in both generation and manipulation. The pre-trained CLIP encoders help TiGAN to better understand images and texts semantically.

TiGAN uses state-of-the-art (SOTA) StyleGAN2 (Karras et al. 2020) as its backbone and use Contrastive Language-Image Pre-training (CLIP) (Radford et al. 2021) to inject text information into StyleGAN2. CLIP is a multi-modal model pre-trained on 400 million text-image pairs and consists of one image encoder and one text encoder that respectively map images and text into a unified joint embedding space. Using CLIP, TiGAN can evaluate the semantic similarity of text with images inside the joint embedding space. We train TiGAN with various proposed contrastive losses that encourage the model to learn a better text-to-image mapping with disentangled intermediate features. On top of the trained model, we propose an image manipulation mechanism that manipulates an image according to text feedback and avoids undesirable visible changes. We achieve this by only updating the intermediate features of the generator that are relevant to the text.

To summarize, we propose a novel model for Text-based Interactive image generation (abbreviated TiGAN). Our main contributions are as follows:

- We propose a novel text-to-image generation model, which seamlessly integrates SOTA StyleGAN2 and CLIP model. To achieve a better text-to-image mapping with disentangled, semantically meaningful features, we also propose new contrastive losses to train the model;
- We further propose a new text-guided image manipulation mechanism, which can handle complex text information and maintain image consistency in the interaction;
- We conduct extensive experiments, demonstrating the advances of the proposed method over SOTA methods in both standard text-to-image generation and interactive image generation settings; Human evaluations in different scenarios further verify the effectiveness of the proposed method compared to existing works.

Proposed Framework

Based on generative adversarial networks (GANs) (Goodfellow et al. 2014) and contrastive language-image pre-training model (CLIP) (Radford et al. 2021), our framework consists of a text-to-image generation module and a text-guided image manipulation mechanism. Our proposed framework for interactive image generation is illustrated in Figure 1, with details described below. Different from the standard text-to-image generation task which is in a single-round setting, interactive image generation is naturally in a multi-round setting. At every round, the user will provide natural language to the proposed model, the model will generate or manipulate the images according to the requirements. The images will be fed to the user to obtain further feedback. The session will end when the user is satisfied with the results.

Architecture of the Proposed TiGAN

In this part, we present the detailed architecture designs for our proposed framework for text-based interactive image Generation. Throughout the paper, $\mathbf{z}$ denotes standard Gaussian noise, $\mathbf{x}$ denotes real image sample, $\mathbf{x}'$ denotes generated image, T denotes raw text description.

Generator architecture The generator is used to generate realistic and high-quality data samples. To achieve this, we build our generator based on the StyleGAN2 architecture (Karras et al. 2020). Our proposed generator architecture is illustrated in Figure 2, where $\mathbf{w}$ denotes the intermediate latent vector, and $\{\mathbf{s}_i\}_{i=1}^m$ denote vectors obtained by applying learned transformations on $\mathbf{w}$. These transformations are affine transformations in original StyleGAN2. Throughout the paper, we use $\mathbf{s}$ to denote the concatenation of vectors $\{\mathbf{s}_i\}_{i=1}^m$, which is defined as the *style vector* following previous work (Wu, Lischinski, and Shechtman 2021).

Different from the original StyleGAN2, our proposed generator requires extra text features as inputs. Thus the main challenge is how to effectively extend the unconditional SOTA model to a conditional one by utilizing the text information. Existing works (Zhu et al. 2019; Xu et al. 2018; Zhang et al. 2021) inject text information either by directly concatenating the text feature with noise vector, or updating latent noise by learn-able scale and bias factors. Different from these methods that exploit different ways to update the noise vectors (initial input of the generator), we handle the problem by updating well-disentangled, intermediate features of the generator.

Intuitively, dimensions of a well-disentangled feature vector should be highly independent. Ideally, each dimension

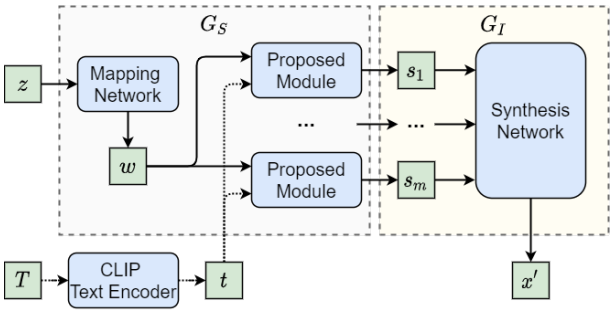

Figure 2: Illustration of the proposed generator architecture. Replacing the proposed module with affine transformation and removing text information will lead to the original generator in StyleGAN2.

should control a specific visible attribute of the generated images. Consequently, accurate text-to-image generation can be achieved if one can directly learn a mapping from text to the well-disentangled features.

To this end, let the *style space* $\mathcal{S}$ be the space spanned by style vectors. As analyzed in some previous works (Wu, Lischinski, and Shechtman 2021; Liu et al. 2020), style space is shown to be well-disentangled. Inspired by these works, we propose to directly inject text information into this disentangled style space. We propose the following two modules to replace the affine transformations on $\mathbf{w}$ in original StyleGAN2:

$$\mathbf{s}_i = \pi_i([\kappa_i(\mathbf{t}), \mathbf{w}]), \text{ and} \tag{1}$$
$$\mathbf{s}_i = \phi_i(\mathbf{t}) \odot \psi_i(\mathbf{w}) + \chi_i(\mathbf{t}) \,, \tag{2}$$

where $\pi_i, \kappa_i, \phi_i, \psi_i, \chi_i$ denote different learn-able functions constructed using 2-layer neural networks, $\odot$ denotes element-wise multiplication, $[\cdot, \cdot]$ denotes vector concatenation, $\mathbf{t}$ denotes text feature extracted with pre-trained CLIP model. With the proposed module, the generator can generate images that match text descriptions. In practice, one can choose one of the modules, or use both modules in the generator. In experiments, we start from using (1) for all $\mathbf{s}_i$, and gradually tune the model architecture by using (2) for some layers. Generally, using only (1) can lead to promising results, using (2) for the last few layers may further improve the results.

Discriminator architecture In standard unconditional settings, the discriminator $D(\cdot)$ is trained to distinguish the real samples from fake samples. In our conditional setting, the discriminator should also consider text information to distinguish samples. To incorporate the text information, we propose to use the architecture in Figure 3, where $f_R(\mathbf{x})$ is a scalar that indicates the unconditional realness of the image as the standard discriminator output; and $f_D(\mathbf{x})$ is the semantic feature extracted by the discriminator. An image $\mathbf{x}$ is classified as real when it has both high similarity with text T and large unconditional realness $f_R(\mathbf{x})$. Thus we can define $D(\mathbf{x}) = f_R(\mathbf{x}) + \langle f_D(\mathbf{x}), \mathbf{t} \rangle$ as the realness of image $\mathbf{x}$ given text feature $\mathbf{t}$.

Consequently, the standard loss functions for our generation and discriminator are:

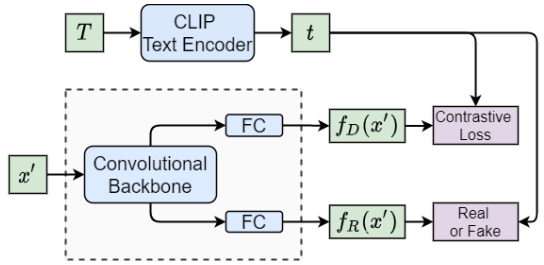

Figure 3: Illustration of the proposed discriminator. FC denotes fully-connected layers.

$$\mathcal{L}_{\mathrm{G}} = -\mathbb{E}_{p(\mathbf{x}')} \left[\log \sigma(D(\mathbf{x}'))\right]$$
$$\mathcal{L}_{\mathrm{D}} = -\mathbb{E}_{p(\mathbf{x})} \left[\log \sigma(D(\mathbf{x}))\right] - \mathbb{E}_{p(\mathbf{x}')} \left[\log(1 - \sigma(D(\mathbf{x}'))\right]$$

where $\sigma(\cdot)$ is the sigmoid function, $p(\mathbf{x}), p(\mathbf{x}')$ denote the distribution of real and generated images respectively.

Text-image matching via Contrastive Learning

In TiGAN, we propose two additional contrastive losses to enhance the text-image matching. Let $\{(\mathbf{x}_i, T_i)\}_{i=1}^n$ be a mini-batch of text-image pairs and $\{\mathbf{x}'_i\}_{i=1}^n$ be the corresponding generated fake images. f_I, f_T denote the image encoder and text encoder of CLIP respectively, and $\mathbf{t}_i = f_T(T_i)$ denotes the CLIP text feature of T_i. We propose to add the following contrastive loss

$$\begin{aligned} &\mathcal{L}_{\text{CLIP}}(\{\mathbf{x}'_i\}_{i=1}^n, \{T_i\}_{i=1}^n) \\ =& -\lambda \sum_{i=1}^n \log \frac{\exp(\tau \cos(f_I(\mathbf{x}'_i), \mathbf{t}_i))}{\sum_{j=1}^n \exp(\tau \cos(f_I(\mathbf{x}'_i), \mathbf{t}_j))} \\ &- (1-\lambda) \sum_{j=1}^n \log \frac{\exp(\tau \cos(f_I(\mathbf{x}'_j), \mathbf{t}_j))}{\sum_{i=1}^n \exp(\tau \cos(f_I(\mathbf{x}'_i), \mathbf{t}_j))} \end{aligned} \tag{3}$$

where λ and τ are hyper-parameters, and $\cos(\cdot, \cdot)$ denotes cosine similarity. Intuitively, minimizing $\mathcal{L}_{\text{CLIP}}$ encourages the generator to generate image $\mathbf{x}'_i$ that has high semantic similarity with the corresponding text description T_i. This also encourage $\mathbf{x}'_i$ to have low semantic similarity with $\{T_j\}_{j \neq i}$, which are the text descriptions of other images.

In addition, we propose the following contrastive loss to regularize the discriminator.

$$\begin{aligned} &\mathcal{L}_{\text{CD}}(\{\mathbf{x}_i\}_{i=1}^n, \{T_i\}_{i=1}^n) \\ =& -\lambda \sum_{i=1}^n \log \frac{\exp(\tau \cos(f_D(\mathbf{x}_i), \mathbf{t}_i))}{\sum_{j=1}^n \exp(\tau \cos(f_D(\mathbf{x}_i), \mathbf{t}_j))} \\ &- (1-\lambda) \sum_{j=1}^n \log \frac{\exp(\tau \cos(f_D(\mathbf{x}_j), \mathbf{t}_j))}{\sum_{i=1}^n \exp(\tau \cos(f_D(\mathbf{x}_i), \mathbf{t}_j))} \end{aligned} \tag{4}$$

where $f_D(\mathbf{x}_i)$ denotes the feature from the discriminator as illustrated in Figure 3. $\mathcal{L}_{\text{CD}}$ encourages the discriminator to extract semantically meaningful features aligned with input text.

The final loss functions for the generator and the discriminator are defined respectively as:

$$\mathcal{L}'_G = \mathcal{L}_{\text{G}} + \alpha\mathcal{L}_{\text{CLIP}}(\{\mathbf{x}'_i\}_{i=1}^n, \{T_i\}_{i=1}^n) + \beta\mathcal{L}_{\text{CD}}(\{\mathbf{x}'_i\}_{i=1}^n, \{T_i\}_{i=1}^n), \quad (5)$$

$$\mathcal{L}'_D = \mathcal{L}_{\text{D}} + \beta\mathcal{L}_{\text{CD}}(\{\mathbf{x}_i\}_{i=1}^n, \{T_i\}_{i=1}^n). \quad (6)$$

During the training process, only the parameters of the generator and discriminator are updated. The parameters of the CLIP text and image encoders are fixed and loaded from the pre-trained checkpoint. In later section, we will discuss the difference between our work and other methods that also use contrastive loss (Xu et al. 2018; Zhang et al. 2021). We also performed an ablation study to help better understanding the impact of these contrastive losses.

Interactive Image Generation

Training with (5) and (6) results in a standard text-to-image generation model for a single-round interaction. To extend our model for interactive generation, we regard the problem as a combination of text-to-image generation and a sequence of text-guided image manipulation. Thus our next step is to design a method for image manipulation that only allows the model to manipulate target attributes of the image. With this, information from previous interactions can be maximally preserved, undesirable image changes can be maximally avoided.

Let $\mathbf{z}$ be a noise sampled from standard Gaussian distribution, $\mathbf{t}$ be a text feature from the dataset extracted with CLIP. The text-to-image generation process can be formulated as $\mathbf{x} = G_I(\mathbf{s}), \mathbf{s} = G_S(\mathbf{t}, \mathbf{z})$, where $\mathbf{s} = [\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_m]$ denotes the generated style vector. As shown in Figure 2, G_S consists of a mapping network and a newly proposed module, which generates a style vector $\mathbf{s}$ given text $\mathbf{t}$. G_I denotes the synthesis network in Figure 2, which will generate an image based on the style vector $\mathbf{s}$. To manipulate an image $\mathbf{x}$ with style $\mathbf{s}$ according to a new text $\mathbf{t}'$, we first identify the most relevant dimensions of $\mathbf{s}$, denoted as $\{c_i\}_{i=1}^k$. Then we generating new style $\mathbf{s}'$ via:

$$[\mathbf{s}']_i = \begin{cases} [\mathbf{s}]_i + \gamma([G_S(\mathbf{z}, \mathbf{t}')]_i - [\mathbf{s}]_i) & \text{if } i \in \{c_i\}_{i=1}^k \\ [\mathbf{s}]_i & \text{otherwise} \end{cases} \quad (7)$$

where $[\mathbf{s}]_i$ denotes the i^{th} element of $\mathbf{s}$, $\gamma > 0$ is the step size (we set $\gamma = 1$ in practice). With the updated style vector, a new image is generated via $\mathbf{x}' = G_I(\mathbf{s}')$.

To obtain the relevant dimensions $\{c_i\}_{i=1}^k$ of $\mathbf{s}$, we follow the same strategy as (Patashnik et al. 2021). Let $\tilde{\mathbf{s}}_i \in R^{\dim(s)}$ be a vector with value η_i on its i^{th} dimension and 0 on other dimensions ($\tilde{\mathbf{s}}_i$ has the same dimensionality as $\mathbf{s}$). We use the following term to evaluate the effects of revising i^{th} dimension:

$$\Delta\mathbf{r}_i = \mathbb{E}_{\mathbf{s}}\left[f_I(G_I(\mathbf{s} + \tilde{\mathbf{s}}_i)) - f_I(G_I(\mathbf{s})\right], \quad \mathbf{s} = G_S(\mathbf{z}, \mathbf{t}) \quad (8)$$

where $\mathbf{z}$ is sampled from standard Gaussian distribution, $\mathbf{t}$ is randomly sampled text from the dataset. Intuitively, $\Delta\mathbf{r}_i$ evaluates the semantic feature change of revising i^{th} dimension of style vector. After obtaining $\Delta\mathbf{r}_i$ for all dimensions, we select all the dimension i satisfying:

$$\cos(\Delta\,\mathbf{t}, \Delta r_i) \geq a \quad (9)$$

where $a > 0$ is a threshold, $\Delta\,\mathbf{t}$ is the desired semantic change evaluated by CLIP. $\Delta\,\mathbf{t}$ can be estimated in different ways. For instance, let f_T be the text encoder of CLIP and we would like to edit the hair color of the human face in the image. $\Delta\,\mathbf{t}$ can be estimated using prompts: $\Delta\,\mathbf{t} = f_T(\text{a face with black hair}) - f_T(\text{a face with hair})$. It can also be directly estimated by $\Delta\,\mathbf{t} = f_T(\text{this person should have black hair}) - \mathbf{t}$, where $\mathbf{t}$ is the text feature of previous round's instruction or the feature of an empty string (for the first round). In practice, we found both ways work equally well.

Related Work

Compared to existing works, our proposed framework is more general and can be applied in different scenarios.

Text-to-Image Generation There are two major categories of text-to-image generation models. (Xu et al. 2018; Zhu et al. 2019; Zhang et al. 2021) propose to use GAN-based structures, (Ramesh et al. 2021; Ding et al. 2021) propose to combine discrete variational auto-encoder (VAE) (van den Oord, Vinyals, and Kavukcuoglu 2017) and transformer (Vaswani et al. 2017). Although (Ramesh et al. 2021; Ding et al. 2021) achieve better qualitative results compared to GAN-based models, they are large models trained on huge dataset, which is inaccessible to most researchers, e.g. DALL-E (Ramesh et al. 2021) has over 12 billion parameters and is trained on a dataset consists of 250 million text-image pairs.

Our proposed model follows the GAN-based structure. Although AttnGAN (Xu et al. 2018) propose DAMSM which also use contrastive loss, it only trains generator with the contrastive loss. XMC-GAN (Zhang et al. 2021) proposed to use contrastive loss for both generator and discriminator, and used a complicated architecture for SOTA results. Different from these works that directly design complex model architectures in a heuristical way, we focus on how to efficiently turn an existing SOTA unconditional GAN into a text-conditioned GAN. To this end, we propose to inject text information into the disentangled feature space of the generator, and train both generator and discriminator with the proposed contrastive loss. Pre-trained CLIP model is also incorporated to provide better semantic information for the training process. As a result, we obtains SOTA performance on text-to-image generation tasks.

Text-guided Image Manipulation General idea on manipulating images consists of three steps: map images into some latent spaces, manipulate the obtained latent vectors, generate images with the manipulated latent vectors. Existing works (Wu, Lischinski, and Shechtman 2021; Liu et al. 2020; Li et al. 2020; Xia et al. 2021; Patashnik et al. 2021) handle the third step by directly using pre-trained GANs, and focus on the first or second step. Different from these works, we solve the challenge of the third step by training a better text-to-image generation model. Now we briefly discuss the SOTA methods for the second step, which is also the topic our manipulation mechanism focus on.

TediGAN (Xia et al. 2021) propose to train different encoders that map different modalities into the same latent

(a) A green train is coming down the track (b) A yellow school bus in the forest (c) A small kitchen with a low ceiling (d) A peaceful lake in a cloudy day (e) Skyline of a modern city (f) A tower on the mountain

Figure 4: Text-to-image generation examples with model trained on MS-COCO 2014 dataset, and captions are the input text.

space of the generator. To manipulate an image according to a text description, the authors propose to first map both image and text into the joint latent space, then combine the two latent vectors by replacing some elements in image latent vector using elements from text latent vector. The resulting latent vector will be fed into the generator to generate manipulated image. StyleCLIP (Patashnik et al. 2021) also propose to utilize the pre-trained CLIP model and maximize the semantic similarity between the resulting images and text descriptions. THree different methods are proposed in (Patashnik et al. 2021), we will compare our method to the one with the most promising results, which is denoted as StyleCLIP-Global. StlyeCLIP-Global use a similar strategy as our manipulation mechanism, it first find $\Delta \mathbf{r}_i$ for all the dimensions, then select relevant dimensions and add predefined constant values to the selected elements. The potential drawback of StyleCLIP-Global is that it could lead to weird images, some examples are provided in the Appendix. This usually happens when adding inappropriate constants, rendering the style vectors are outside the support of the G_I. Compared with StyleCLIP-Global, we first train a text-to-image generation model on the given datasets, and then manipulate the style vector by (7) instead of adding constants. Since G_S is trained in conjunction with G_I, the manipulated style vector has little probability being outside the support of G_I.

Interactive Multi-modal Learning The classical multi-round manipulation problem considers the problem where a designer will sequentially generate images for the ultimate goal following a sequence of linguistic instructions (El-Nouby et al. 2019). The SOTA performance of this task is achieved by a self-supervised framework which incorporates counterfactual thinking to overcome data scarcity (Fu et al. 2020). These two methods are based on predefined sequences which are not practical for real interactive problems. Furthermore, they will suffer the exposure bias and error accumulation issues, *i.e.*, the image quality becomes worse with more interactions. A POMDP formulation for conversational image editing was also developed to enable fully interaction (Lin et al. 2018), but the manipulation is based on predefined operations without any creation. The fully interactive image generation problem was explored in (Cheng et al. 2020), while the generation quality are miserably poor and can only handle relatively simple datasets. Compared to the aforementioned methods, our proposed method is fully interactive, does not have error accumulation, and can handle both image generation and manipulation problems on complex datasets.

Experiments

We conduct extensive experiments on three different datasets: UT Zappos50k (Yu and Grauman 2014), MS-COCO 2014 (Lin et al. 2014) and Multi-modal CelebA-HQ (Xia et al. 2021). The experiments are implemented under two settings: single-round image generation and interactive (multi-round) image generation. All experiments are conducted on 4 Nvidia Tesla V100 GPU and implemented with Pytorch. Details of the datasets, the experimental setup and hyper-parameters are provided in the Appendix.

Text-to-image Generation

To test the generation quality of our method for text-to-image generation, we first evaluate it on MS-COCO 2014, a dateset containing complex scenes and many kinds of objects and is commonly used in text-to-image generation tasks. Following previous work (Zhang et al. 2021), we report Fréchet Inception Distance (FID) (Heusel et al. 2017) and Inception Score (IS) (Salimans et al. 2016), which evaluate the quality and the diversity of generated images respectively. 30,000 generated images with randomly sampled text are used to compute the metrics. The main results are provided in Table 2. Our proposed method outperforms previous SOTA model XMC-GAN (Zhang et al. 2021). Compared to XMC-GAN that contains many attention modules, our proposed model has less parameters and smaller model size, while achieving better IS and FID scores. Some generated examples are shown in Figure 4, more results are provided in the Appendix.

In addition, (Xia et al. 2021) provides results of text-to-image generation on Multi-modal CelebA-HQ. We also have compared our method with it in Table 3. Note that the results in (Xia et al. 2021) are based on the generator pre-trained on FFHQ (Karras, Laine, and Aila 2019), which is directly used to calculate the FID score on Multi-modal CelebA-HQ. Since FID measures the distance between generated images and real images from a dataset, it is fairer to fine-tune the generator on Multi-modal CelebA-HQ before evaluating FID. Thus we report both the original results from (Xia et al. 2021) and the results of fine-tuning the model before applying their methods. Following (Xia et al. 2021), all the results are evaluated by generating 6000 images using the descriptions from test set of Multi-modal CelebA-HQ. Note that we

METHOD	AR (10) ↓	SR (10) ↑	SR (20) ↑	SR (50) ↑	CGAR (10) ↑	CGAR (20) ↑	CGAR (50) ↑
				DATASET: UT ZAPPOS50K			
SEQATTNGAN	7.090	0.426	0.506	0.596	0.798	0.847	0.879
TEDIGAN	7.537	0.419	0.442	0.492	0.781	0.802	0.818
STYLECLIP-GLOBAL	6.954	0.424	0.462	0.476	0.757	0.773	0.790
TIGAN (W/O THRESHOLD)	6.056	0.628	0.724	0.818	0.896	0.922	0.951
TIGAN	**5.412**	**0.682**	**0.784**	**0.886**	**0.896**	**0.941**	**0.970**
				DATASET: MULTI-MODAL CELEBA HQ			
SEQATTNGAN	6.284	0.582	0.728	0.835	0.878	0.926	0.944
TEDIGAN	5.769	0.597	0.670	0.706	0.854	0.876	0.897
STYLECLIP-GLOBAL	5.510	0.628	0.664	0.666	0.864	0.879	0.880
TIGAN (W/O THRESHOLD)	4.942	0.737	0.816	0.852	0.923	0.950	0.957
TIGAN	**4.933**	**0.761**	**0.830**	**0.886**	**0.928**	**0.947**	**0.967**

Table 1: Interactive image generation results evaluated with user simulator. Average round (AR) is the average number of needed interactions. Success rate (SR) is defined as the ratio of number of successful cases to the number of total cases. Correctly generated attribute rate (CGAR) denotes the average percentage of correctly generated attributes in all the cases. Integer in the parenthesis denotes the maximal number of interaction rounds.

Method	IS ↑	FID ↓
AttnGAN	23.61	33.10
Obj-GAN	24.09	36.52
DM-GAN	32.32	27.23
OP-GAN	27.88	24.70
XMC-GAN	30.45	9.33
TiGAN	31.95	**8.90**

Table 2: Text-to-image generation results on MS-COCO 2014.

Method	IS ↑	FID ↓
w/o fine-tuning (Xia et al. 2021)		
AttnGAN	-	125.98
ControlGAN	-	116.32
DFGAN	-	137.60
DMGAN	-	131.05
TediGAN	-	106.57
with fine-tuning		
TediGAN + fine-tune	2.29	27.39
TiGAN	**2.85**	**11.35**

Table 3: Text-to-image generation results on Multi-modal CelebA-HQ.

do not report LPIPS (Zhang et al. 2018) as (Xia et al. 2021), because we found that LPIPS can be easily hacked in this experiment, where one can easily obtain good LPIPS that does not represent a good model. More discussions can be found in the Appendix.

Interactive Image Generation

We then test the proposed method on UT Zappos 50k and Multi-modal CelebA-HQ for the interactive image generation task. We choose these two datasets because each image has associated attributes in these datasets. Some examples are shown in Figure 11 in the Appendix.

Some visualization results are illustrated in Figure 5. It is clear that the proposed method can manipulate the image correctly and maintain the manipulated attributes during the whole interaction. We also evaluate the proposed method quantitatively. To this end, we design a user simulator to gives text feedback based on the generated images. In each test case, the user simulator has some target attributes in mind, which are randomly sampled from the dataset and unknown to the model. The model starts from generating a random image and feed the image to the user simulator. The user simulator will give feedback by randomly pointing out one of the target attributes that is not satisfied by the generation. The feedback will then be fed to the model for further image manipulation. The interaction process will stop when the user simulator find the generated image matches all the target attributes. In the experiments, we use neural network based classifier as the user simulator, which classifies the attributes of the generated images, and output text feedback based on prompt engineering. The details of constructing user simulator can be found in Appendix.

The main results of averaging over 1000 test cases are reported in Table 1. Note that we set a maximal number of interaction rounds. Once the interaction exceed this number, the user simulator would directly treat current test as a failure case and start a new test case. The attributes used in this experiment are summarized in Table 6 in Appendix. We compare our proposed method with currently SOTA image manipulation methods: StyleCLIP-Global, TediGAN and existing SOTA interactive method SeqAttnGAN (Cheng et al. 2020). Note that for fair comparison, we re-implemented SeqAttnGAN using StyleGAN2 and CLIP model, which leads to a much more powerful variant than (Cheng et al. 2020). We also provide the results of our method without threshold during image manipulation, *i.e.*, instead of using method in Eq. (7), we directly generate new style vector $\mathbf{s}'$ using feedback $\mathbf{t}'$ via $\mathbf{s}' = G_S(\mathbf{z}, \mathbf{t}')$. From the results, we can conclude that our proposed method leads to better interaction efficiency as it needs less interaction rounds in average.

Human Evaluation

We also conducted human evaluation on Amazon Mechanical Turk (MTurk) for text-to-image generation, text-guided image manipulation and interactive image generation. In the evaluation, the workers were provided 100 images from each

Method	Text-to-image Generation		Text-guided Manipulation			Interactive Generation	
	Realistic ↑	Match ↑	Realistic ↑	Match ↑	Consistency ↑	Realistic ↑	Match ↑
			Dataset: UT Zappos50k				
SeqAttnGAN	3.66	3.82	3.88	2.86	2.64	3.46	2.78
TediGAN	3.91	2.31	3.50	3.04	2.95	3.66	2.60
StyleCLIP-Global	-	-	3.28	2.30	2.93	3.84	2.28
TiGAN	**4.12**	**4.11**	**4.10**	**3.64**	**2.98**	**4.18**	**2.98**
			Dataset: Multi-modal CelebA HQ				
SeqAttnGAN	3.10	3.59	3.74	3.58	3.26	2.92	2.34
TediGAN	3.19	2.49	**4.50**	2.92	2.62	3.86	2.62
StyleCLIP-Global	-	-	4.14	3.60	3.42	2.84	2.36
TiGAN	**3.27**	**4.09**	4.36	**3.68**	**3.72**	**4.00**	**2.76**

Table 4: Results of Human Evaluation on Zappos 50k and Multi-modal CelebA-HQ. Note text-to-image generation and text-guided manipulation are under single-round setting, while interactive generation are under multi-round setting. StyleCLIP is a image manipulation method and can not be applied in single-round text-to-image generation task.

(a) A woman face (b) A face wearing earrings (c) This is a face with blonde hair (d) She has short hair (e) A face with heavy makeup

(f) A young man face (g) A face with red hair (h) He has beard (i) This is a face wearing glasses (j) He is wearing a hat

Figure 5: Interactive image generation of the proposed method. Each row is a user session, and each sub-figure is a result of one round interaction. The caption of each sub-figure is the text input from the user.

method, which are generated or manipulated according to randomly sampled texts. The workers were asked to judge whether the generated or manipulated images match the text and how realistic the images are. Furthermore, the workers are also asked to judge whether the consistency is well maintained in manipulation, in the sense that there are no undesirable changes observed. The three metrics are denoted as Match, Realistic and Consistency respectively. The workers are all from the US and were required to have performed at least 10,000 assignments approved with an approval rate $\geq$ 98%. For each metric, the workers are asked to score the images at scale of 1 to 5, where 5 denotes the most realistic/best matching/most consistent. The main results are provided in Table 4 and more details of the human evaluation can be found in the Appendix.

Ablation Study

To better understand the proposed method, we conducted an ablation study to determine how each component of the loss function influence TiGAN. The main results are provided in Table 5. We observe that excluding either $\mathcal{L}_{\text{CLIP}}$ or $\mathcal{L}_{\text{CD}}$ leads to performance degeneration. Meanwhile, $\mathcal{L}_{\text{CLIP}}$ seems to contribute more than $\mathcal{L}_{\text{CD}}$, as the model trained without $\mathcal{L}_{\text{CLIP}}$ has much poorer diversity according to IS.

Method	IS ↑	FID ↓
TiGAN w/o $\mathcal{L}_{\text{CLIP}}$	22.87	19.62
TiGAN w/o $\mathcal{L}_{\text{CD}}$	27.21	18.21
TiGAN	**31.95**	**8.90**

Table 5: Ablation study on MS-COCO 2014.

Conclusions

In this paper, we proposed TiGAN for interactive image generation and manipulation from text. Using both human and automated evaluation, we showed that TiGAN is able to generate more realistic images that better match the text in fewer rounds than prior SOTA methods. Empirical results on several datasets show that TiGAN improves both interaction efficiency and image quality while better avoids undesirable image manipulations during interaction.

References

Cheng, Y.; Gan, Z.; Li, Y.; Liu, J.; and Gao, J. 2020. Sequential attention GAN for interactive image editing. In *ACMMM*.

Ding, M.; Yang, Z.; Hong, W.; Zheng, W.; Zhou, C.; Yin, D.; Lin, J.; Zou, X.; Shao, Z.; Yang, H.; and Tang, J. 2021. CogView: Mastering Text-to-Image Generation via Transformers. arXiv:2105.13290.

El-Nouby, A.; Sharma, S.; Schulz, H.; Hjelm, D.; Asri, L. E.; Kahou, S. E.; Bengio, Y.; and Taylor, G. W. 2019. Tell, draw, and repeat: Generating and modifying images based on continual linguistic instruction. In *ICCV*.

Fu, T.-J.; Wang, X.; Grafton, S.; Eckstein, M.; and Wang, W. Y. 2020. Iterative language-based image editing via self-supervised counterfactual reasoning. In *EMNLP*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

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. *Advances in neural information processing systems*, 30.

Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4401–4410.

Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8110–8119.

Li, B.; Qi, X.; Lukasiewicz, T.; and Torr, P. H. 2020. Manigan: Text-guided image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7880–7889.

Lin, T.-H.; Bui, T.; Kim, D. S.; and Oh, J. 2018. A multimodal dialogue system for conversational image editing.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Liu, Y.; Li, Q.; Sun, Z.; and Tan, T. 2020. Style Intervention: How to Achieve Spatial Disentanglement with Style-based Generators? arXiv:2011.09699.

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*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*.

Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training gans. *Advances in neural information processing systems*, 29: 2234–2242.

van den Oord, A.; Vinyals, O.; and Kavukcuoglu, K. 2017. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6309–6318.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*.

Wu, Z.; Lischinski, D.; and Shechtman, E. 2021. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12863–12872.

Xia, W.; Yang, Y.; Xue, J.-H.; and Wu, B. 2021. TediGAN: Text-Guided Diverse Face Image Generation and Manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xu, T.; Zhang, P.; Huang, Q.; Zhang, H.; Gan, Z.; Huang, X.; and He, X. 2018. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1316–1324.

Yu, A.; and Grauman, K. 2014. Fine-Grained Visual Comparisons with Local Learning. In *Computer Vision and Pattern Recognition (CVPR)*.

Zhang, H.; Koh, J. Y.; Baldridge, J.; Lee, H.; and Yang, Y. 2021. Cross-Modal Contrastive Learning for Text-to-Image Generation. arXiv:2101.04702.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.

Zhu, M.; Pan, P.; Chen, W.; and Yang, Y. 2019. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5802–5810.