

# TIME: Text and Image Mutual-Translation Adversarial Networks

**Bingchen Liu, Kunpeng Song, Yizhe Zhu, Gerard de Melo, Ahmed Elgammal**  
**Department of Computer Science, Rutgers University**

{bingchen.liu, kumpeng.song, yizhe.zhu}@rutgers.edu  
gdm@demelo.org, elgammal@cs.rutgers.edu

## Abstract

Focusing on text-to-image (T2I) generation, we propose Text and Image Mutual-Translation Adversarial Networks (TIME), a lightweight but effective model that jointly learns a T2I generator  $G$  and an image captioning discriminator  $D$  under the Generative Adversarial Network framework. While previous methods tackle the T2I problem as a uni-directional task and use pre-trained language models to enforce the image–text consistency, TIME requires neither extra modules nor pre-training. We show that the performance of  $G$  can be boosted substantially by training it jointly with  $D$  as a language model. Specifically, we adopt Transformers to model the cross-modal connections between the image features and word embeddings, and design an annealing conditional hinge loss that dynamically balances the adversarial learning. In our experiments, TIME achieves state-of-the-art (SOTA) performance on the CUB and MS-COCO dataset (Inception Score of 4.91 and Fréchet Inception Distance of 14.3 on CUB), and shows promising performance on MS-COCO on image captioning and downstream vision-language tasks.

## 1 Introduction

There are two main aspects to consider when approaching the text-to-image (T2I) task: the image generation quality and the image–text semantic consistency. The T2I task is commonly modeled by a conditional Generative Adversarial Network (cGAN) (Mirza and Osindero 2014; Goodfellow et al. 2014), where a Generator ( $G$ ) is trained to generate images given the texts describing the contents, and a Discriminator ( $D$ ) learns to determine the authenticity of the images, conditioned on the semantics defined by the given texts.

To address the first aspect, Zhang et al. (2017) introduced StackGAN by letting  $G$  generate images at multiple resolutions, and adopted multiple  $D$ s to jointly refine  $G$  from coarse to fine levels. StackGAN invokes a pre-trained Recurrent-Neural-Network (RNN) (Hochreiter and Schmidhuber 1997; Mikolov et al. 2010) to provide text conditioning for the image generation. To approach the second aspect, Xu et al. (2018) take StackGAN as the base model and propose AttnGAN, which incorporates word embeddings into the generation and consistency-checking processes. A pre-trained Deep-Attentional-Multimodal-Similarity-Model (DAMSM)

is introduced, which better aligns the image features and word embeddings via attention mechanism.

While the T2I performance continues to advance (Qiao et al. 2019; Zhu et al. 2019; Cai et al. 2019; Li et al. 2019a; Yin et al. 2019; Hinz, Heinrich, and Wermter 2019), the follow-up methods all share two common traits. First, they all adopt the same stacked structure of  $G$  that requires multiple  $D$ s. Second, they all rely on the pre-trained DAMSM from AttnGAN to maintain the image–text consistency. However, these methods fail to take advantage of recent advances in both the GAN and NLP literature (Karras et al. 2017; Karras, Laine, and Aila 2019; Vaswani et al. 2017; Devlin et al. 2018; Radford et al. 2019). The rapidly progressing research in these two fields provides the opportunity to explore a substantial departure from previous works on text-to-image modeling. In particular, as StackGAN and follow-up works all depend on: 1. a pre-trained text encoder for word and sentence embeddings; 2. an additional image encoder to ascertain image–text consistency; two important questions arise. First, can we skip the pre-training step and elegantly train the text encoder as part of  $D$ ? Second, can we abandon the extra CNN (in the DAMSM module which extracts image features) and use  $D$  as the image encoder? If the answers are affirmative, two further questions can be explored. When  $D$  and the text encoder are jointly trained to match the visual and text features, can we obtain an image captioning model from them? Furthermore, since  $D$  is trained to extract text-relevant image features, will it benefit  $G$  in generating more semantically consistent images?

With these questions in mind, we present the Text and Image Mutual-translation adversarial nEtworK (TIME). To the best of our knowledge, this is the first work that jointly handles both text-to-image and image captioning in a single model using the GAN framework. Our contributions can be summarized as follows:

1. We propose an efficient model: Text and Image Mutual-Translation Adversarial Networks (TIME), for T2I tasks trained in an end-to-end fashion, without any need for pre-trained models or complex training strategies.
2. We introduce two techniques: 2-D positional encoding for a better attention operation and annealing hinge loss to dynamically balance the learning paces of  $G$  and  $D$ .
3. We show that the sentence-level text features are no longer

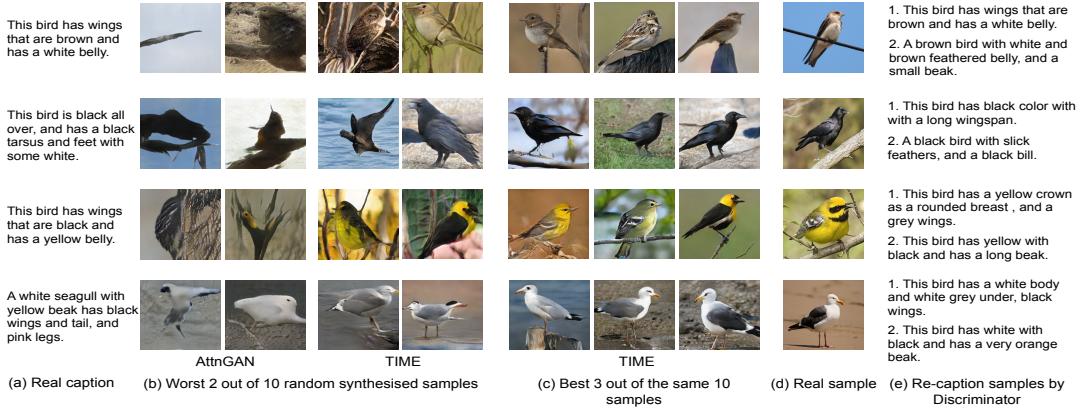


Figure 1: Text-to-image results of TIME on the CUB dataset, where  $D$  works as a stand-alone image-captioning model.

needed in T2I task, which leads to a more controllable T2I generation that is hard to achieve in previous models.

- Extensive experiments show that our proposed TIME achieves superior results on text-to-image tasks and promising results on image captioning. Fig. 1-(c) showcases the superior synthetic image quality from TIME, while Fig. 1-(e) demonstrates TIME’s image captioning capability.

## 2 Related Work and Background

Recent years have witnessed substantial progress in the text-to-image task (Mansimov et al. 2015; Nguyen et al. 2017; Reed et al. 2017, 2016; Zhang et al. 2017; Xu et al. 2018; Han, Guerrero, and Pavlovic 2019) owing largely to the success of deep generative models (Goodfellow et al. 2014; Kingma and Welling 2013; Van den Oord et al. 2016). Reed et al. first demonstrated the superior ability of conditional GANs to synthesize plausible images from text descriptions. StackGAN and AttnGAN then took the generation quality to the next level, which subsequent works built on (Qiao et al. 2019; Zhu et al. 2019; Cai et al. 2019; Li et al. 2019a; Yin et al. 2019; Hinz, Heinrich, and Wermter 2019; Li et al. 2019b). Specifically, MirrorGAN (Qiao et al. 2019) incorporates a pre-trained text re-description RNN to better align the images with the given texts, DMGAN (Zhu et al. 2019) integrates a dynamic memory module on  $G$ , ControlGAN (Li et al. 2019a) employs a channel-wise attention in  $G$ , and SDGAN (Yin et al. 2019) includes a contrastive loss to strengthen the image–text correlation. In the following, we describe the key components of StackGAN and AttnGAN.

**StackGAN as the Image Generation Backbone.** StackGAN adopts a coarse-to-fine structure that has shown substantial success on the T2I task. In practice, the generator  $G$  takes three steps to produce a  $256 \times 256$  image, where three discriminators ( $D$ ) are required to train  $G$ . However, a notable reason for seeking an alternative architecture is that the multi- $D$  design is memory-demanding and has a high computational burden during training. If the image resolution increases, the respective higher-resolution  $D$ s can raise the cost particularly dramatically.

**Dependence on Pre-trained modules.** While the overall framework for T2I models resembles a conditional GAN

(cGAN), multiple modules have to be pre-trained in previous works. In particular, AttnGAN requires a DAMSM, which includes an Inception-v3 model (Szegedy et al. 2016) that is first pre-trained on ImageNet (Deng et al. 2009), and then used to pre-train an RNN text encoder. MirrorGAN further proposes the STREAM model, which is also an additional CNN+RNN structure pre-trained for image captioning.

Such pre-training has several drawbacks, including, first, the additional pre-trained CNN for image feature extraction introduces a significant amount of weights, which can be avoided as we shall later show. Second, using pre-trained modules leads to extra hyper-parameters that require dataset-specific tuning. For instance, in AttnGAN, the weight for the DAMSM loss can range from 0.2 to 100 across different datasets. Last but not least, empirical studies (Qiao et al. 2019; Zhang et al. 2017) show that the pre-trained NLP components do not converge if jointly trained with the cGAN.

**The Image-Text Attention Mechanism.** The attention mechanism employed in AttnGAN can be interpreted as a simplified version of the Transformer (Vaswani et al. 2017), where the three-dimensional image features ( $\text{height} \times \text{width} \times \text{channel}$ ) in the CNN are flattened into a two-dimensional sequence ( $\text{seq-length} \times \text{channel}$  where  $\text{seq-length} = \text{height} \times \text{width}$ ). This process is demonstrated in Fig. 3-(a), where an image-context feature  $f_{it}$  is derived via an attention operation on the reshaped image feature and the sequence of word embeddings. The resulting image-context features are then concatenated to the image features to generate the images. We will show that a full-fledged version of the Transformer can further improve the performance without a substantial additional computational burden.

## 3 The Motivation of Mutual Translation

One may ask that since training the text-to-image model already achieves fruitful results with a pre-trained NLP model, is it necessary to explore the joint-training method? We can answer this question from several aspects.

First, a pre-trained NLP model is not always available given some image datasets. In cases where the given texts do not have a pre-trained NLP model, one can save the separated pre-training time and get a model that translates towards both

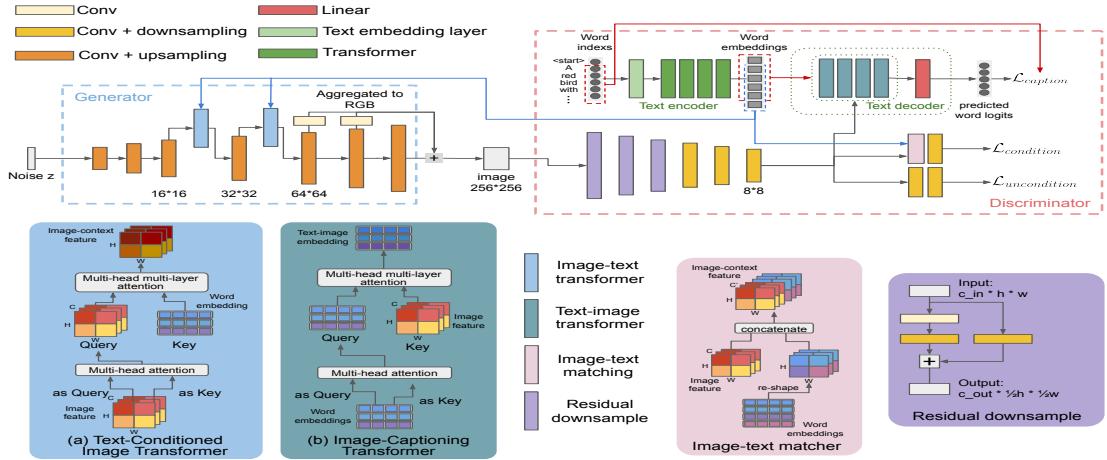


Figure 2: Model overview of TIME. The upper panel shows a high-level summary of our architecture while the lower panel demonstrates the details of the individual modules.

directions with TIME. In case a pre-trained NLP model is available, it is still not guaranteed that the fixed word embeddings are the best for training the image generator. Tuning the hyper-parameters (such as weights of loss objectives from the pre-trained NLP model) for the pre-training methods can be very costly and may not be optimal.

Second, under the GAN framework, balancing the joint training between the Discriminator  $D$  and Generator  $G$  is vital.  $G$  is unlikely to converge if trained with a fixed  $D$ . In the text-to-image task, the pre-trained NLP model serves as a part of  $D$  that provides authenticity signals to  $G$ . Using a pre-trained NLP model is equivalent to fixing a part of  $D$ , which undermines the performance of the whole training schema as a GAN. Instead, the joint training in TIME does not have such restrictions. The NLP parts in TIME learn together with  $G$  and dynamically adjust the word embeddings to serve the training best, leading to better image synthesis quality.

Finally, mutual translation itself can be a crucial pre-training method, which is also studied in (Huang et al. 2018; Li et al. 2020). As we show in the paper, the NLP models learned in TIME have a promising performance on downstream vision-language tasks. Therefore, instead of pre-training only on texts, mutual translation between image and text itself has the potential to be a powerful pre-training method.

## 4 Methodology

In this section, we present our proposed approach. The upper panel in Fig. 2 shows the overall structure of TIME, consisting of a Text-to-Image Generator  $G$  and an Image-Captioning Discriminator  $D$ . We treat a text encoder  $Enc$  and a text decoder  $Dec$  as parts of  $D$ .  $G$ 's Text-Conditioned Image Transformer accepts a series of word embeddings from  $Enc$  and produces an image-context representation for  $G$  to generate a corresponding image.  $D$  is trained on three kinds of input pairs, consisting of captions  $T^{real}$  alongside: (a) matched real images  $I_{match}$ ; (b) randomly mismatched real images  $I_{mis}$ ; and (c) generated images  $I_{fake}$  from  $G$ .

### 4.1 Model Structures

**Aggregated Generator** To trim the model size of the StackGAN structure, we present the design of an aggregated  $G$  as shown in the upper panel of Fig. 2.  $G$  still yields RGB outputs at multiple resolutions, but these RGB outputs are re-scaled and added together as a single aggregated image output. Therefore, only one  $D$  is needed to train  $G$ .

**Text-Conditioned Image Transformer** While prior works (Zhang et al. 2018; Xu et al. 2018) show the benefit of an attention mechanism for the image generative task, none of them dive deeper towards the more comprehensive “multi-head and multi-layer” Transformer design (Vaswani et al. 2017). To explore a better baseline for the T2I task, we redesign the attention in AttnGAN with the Text-Conditioned Image Transformer (TCIT) as illustrated in Fig. 2-(a). In Fig. 3, we show three main differences between TCIT and the form of attention that is widely used in previous T2I models such as AttnGAN. All the attention modules take two inputs: the image feature  $f_i$ , and the sequence of word embeddings  $f_t$ , and gives one output: the revised image feature  $f_{it}$  according to the word embeddings  $f_t$ .

First, Fig. 3-(a) shows the attention module from AttnGAN, where the projected key ( $K$ ) from  $f_t$  is used for both matching with query ( $Q$ ) from  $f_i$  and calculating  $f_{it}$ . Instead, TCIT has two separate linear layers to project  $f_t$  as illustrated in Fig. 3-(b). The intuition is, as  $K$  focuses on matching with  $f_i$ , the other projection value  $V$  can better be optimized towards refining  $f_i$  for a better  $f_{it}$ . Second, TCIT adopts a multi-head structure as shown in Fig. 3-(c). Unlike in AttnGAN where only one attention map is applied, the Transformer replicates the attention module, thus adding more flexibility for each image region to account for multiple words. Third, TCIT stacks the attention layers in a residual structure as in certain NLP models (Devlin et al. 2018; Radford et al. 2019) as illustrated in Fig. 3-(d), for a better performance by provisioning multiple attention layers and recurrently revising the learned features. In contrast, previous GAN models (AttnGAN, SAGAN) adopt attention only in a one-layer fashion.

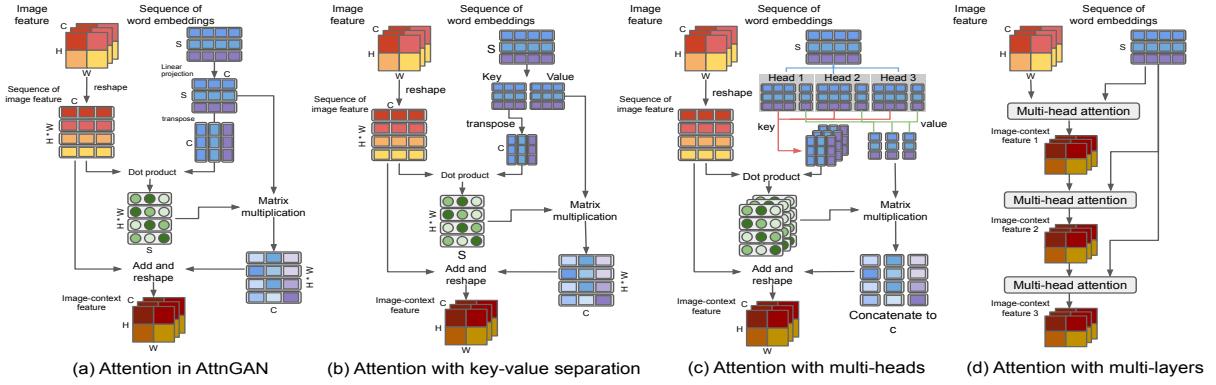


Figure 3: Differences between the attention of AttnGAN and TCIT.

**Image-Captioning Discriminator** We treat the text encoder  $Enc$  and text decoder  $Dec$  as a part of our  $D$ . Specifically,  $Enc$  is a Transformer that maps the word indices into the embeddings while adding contextual information to them. To train  $Dec$  to actively generate text descriptions of an image, an attention mask is applied on the input of  $Enc$ , such that each word can only attend to the words preceding it in a sentence.  $Dec$  is a Transformer decoder that performs image captioning by predicting the next word’s probability from the masked word embeddings and the image features.

**Image-Captioning Transformer** Symmetric to TCIT, the inverse operation, in which  $f_t$  is revised by  $f_i$ , is leveraged for image captioning in  $Dec$ , as shown in Fig. 2-(b). Such a design has been widely used in recent captioning works. In TIME, we show that a simple 4-layer 4-head Transformer is sufficient to obtain high-quality captions and facilitate the consistency checking in the T2I task.

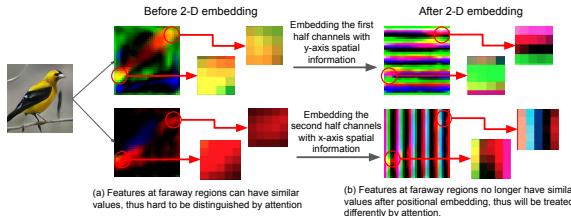


Figure 4: Visualization of 2-D positional encoding.

## 4.2 2-D Positional Encoding for Image Features

When we reshape the image features  $f_i$  for the attention operation, there is no way for the Transformer to discern spatial information from the flattened features. To take advantage of coordinate signals, we propose *2-D positional encoding* as a counter-part to the 1-D *positional encoding* in the Transformer (Vaswani et al. 2017). The encoding at each position has the same dimensionality as the channel size  $c$  of  $f_i$ , and is directly added to the reshaped image feature  $f_i^T \in \mathbb{R}^{d \times c}$ . The first half of dimensions encode the y-axis positions and the second half encode the x-axis, with sinusoidal functions of different frequencies. Such 2-D encoding ensures that closer



Figure 5: Samples generated during the training of TIME, note the visual features emerge in very early iterations.

visual features have a more similar representation compared to features that are spatially more remote from each other. An example  $32 \times 32$  feature-map from a trained TIME is visualized in Fig. 4, where we visualize three feature channels as an RGB image. In practice, we apply 2-D positional encoding on the image features for both TCIT and  $Dec$  in  $D$ . Please refer to the appendix for more details.

## 4.3 Objectives

**Discriminator Objectives** Formally, we denote the three kinds of outputs from  $D$  as:  $D_f()$ , the image feature at  $8 \times 8$  resolution;  $D_u()$ , the unconditional image real/fake score; and  $D_c()$ , the conditional image real/fake score. Therefore, the predicted next word distribution from  $Dec$  is:  $P_k = Dec(Enc(T_{1:k-1}^{\text{real}}), D_f(I_{\text{match}}))$ . Finally, the objectives for  $D$ ,  $Enc$ , and  $Dec$  to jointly minimize are:

$$\mathcal{L}_{\text{caption}} = - \sum_{k=1}^l \log(P_k(T_k^{\text{real}}, D_f(I_{\text{match}}))); \quad (1)$$

$$\begin{aligned} \mathcal{L}_{\text{uncond}} = & -\mathbb{E}[\log(D_u(I_{\text{match}}))] \\ & - \mathbb{E}[\log(1 - D_u(I_{\text{fake}}))]; \end{aligned} \quad (2)$$

along with  $\mathcal{L}_{\text{cond}}$ , which we shall discuss next.

**Annealing Image-Text Matching Loss** During training, we find that  $G$  can learn a good semantic visual translation at very early iterations. As shown in Fig. 5, while the convention is to train the model for 600 epochs on the CUB dataset, we observe that the semantic features of  $T^{\text{real}}$  begin to emerge on  $I_{\text{fake}}$  as early as after 20 epochs. Thus, we argue that it is not ideal to penalize  $I_{\text{fake}}$  by the conditional loss on  $D$  in

a static manner. Since  $I_{\text{fake}}$  is already very consistent to the given  $T^{\text{real}}$ , if we let  $D$  consider an already well-matched input as inconsistent, this may confuse  $D$  and in turn hurt the consistency-checking performance. Therefore, we employ a hinge loss (Lim and Ye 2017; Tran, Ranganath, and Blei 2017) and dynamically anneal the penalty on  $I_{\text{fake}}$  according to how confidently  $D$  predicts the matched real pairs:

$$s_{\text{pivot}} = \text{detach}(\mathbb{E}[D_c(I_{\text{match}}, Enc(T^{\text{real}}))]); \quad (3)$$

$$\begin{aligned} \mathcal{L}_{\text{cond}} &= \mathbb{E}[\min(0, 1 - D_c(I_{\text{match}}, Enc(T^{\text{real}})))] \\ &+ \mathbb{E}[\min(0, 1 + D_c(I_{\text{mismatch}}, Enc(T^{\text{real}})))] \\ &+ \mathbb{E}[\min(0, -s_{\text{pivot}} \times p + D_c(I_{\text{fake}}, Enc(T^{\text{real}})))]. \end{aligned} \quad (4)$$

Here,  $\text{detach}(\cdot)$  denotes that the gradient is not computed for the enclosed function, and  $p = i_{\text{epoch}}/n_{\text{epochs}}$  (current epoch divided by total number) is the annealing factor. The hinge loss ensures that  $D$  yields a lower score on  $I_{\text{fake}}$  compared to  $I_{\text{match}}$ , while the annealing term  $p$  ensures that  $D$  penalizes  $I_{\text{fake}}$  sufficiently in early epochs.

**Generator Objectives** On the other side,  $G$  considers random noise  $z$  and word embeddings from  $Enc$  as inputs, and is trained to generate images that can fool  $D$  into giving high scores on authenticity and semantic consistency with the text. Moreover,  $G$  is also encouraged to make  $D$  reconstruct the same sentences as provided as input. Thus, the objectives for  $G$  to minimize are:

$$\mathcal{L}_{\text{caption-g}} = - \sum_{k=1}^l \log(P_k(T_k^{\text{real}}, D_f(G(z, Enc(T^{\text{real}}))))); \quad (5)$$

$$\mathcal{L}_{\text{uncond-g}} = -\mathbb{E}[\log(D_u(G(z, Enc(T^{\text{real}}))))]; \quad (6)$$

$$\mathcal{L}_{\text{cond-g}} = -\mathbb{E}[D_c(G(z, Enc(T^{\text{real}})), Enc(T^{\text{real}}))]. \quad (7)$$

## 5 Experiments

In this section, we evaluate the proposed model from both the text-to-image and image-captioning directions, and analyze each module’s effectiveness individually. Moreover, we highlight the desirable property of TIME being a more controllable generator compared to other T2I models.

Experiments are conducted on two datasets: CUB (Welin-der et al. 2010) and MS-COCO (Lin et al. 2014). We follow the same convention as in previous T2I works to split the training/testing set. We benchmark the image quality by the Inception Score (IS) (Salimans et al. 2016) and Fréchet Inception Distance (FID) (Heusel et al. 2017), and measure the image–text consistency by R-precision (Xu et al. 2018) and SOA-C (Hinz, Heinrich, and Wermter 2019).

**Backbone Model Structure** Table 1 demonstrates the performance comparison between the StackGAN structure and our proposed “aggregating” structure. AttnGAN as the T2I backbone has been revised by recent advances in the GAN literature (Zhu et al. 2019; Qiao et al. 2019). Seeking a better baseline, we also incorporated recent advances. In particular, columns with “+new” imply that we train the model with equalized learning rate (Karras et al. 2017) and R-1 regularization (Mescheder, Geiger, and Nowozin 2018). “Aggr”

means we replace the “stacked”  $G$  and multiple  $D$ s with the proposed aggregated  $G$  and a single  $D$ . To show the computing cost, we list the relative training times of all models with respect to StackGAN. All models are trained with the optimal hyper-parameter settings and the same loss functions from StackGAN and AttnGAN respectively.

	Inception Score $\uparrow$	R-precision $\uparrow$	Training time $\downarrow$
StackGAN w/o stack	$3.42 \pm 0.05$	$9.25 \pm 3.12$	0.57
StackGAN	$3.82 \pm 0.06$	$10.37 \pm 5.88$	1.0
Aggr GAN	$3.78 \pm 0.03$	$10.21 \pm 5.42$	0.78
Aggr GAN +new	$4.12 \pm 0.03$	$12.26 \pm 4.76$	0.85
AttnGAN w/o stack	$4.28 \pm 0.02$	$64.82 \pm 4.43$	0.71
AttnGAN	$4.36 \pm 0.03$	$67.82 \pm 4.43$	1.14
Aggr AttnGAN	$4.34 \pm 0.02$	$67.63 \pm 5.43$	0.86
Aggr AttnGAN +new	$4.52 \pm 0.02$	$70.32 \pm 4.19$	1.0

Table 1: Comparison between **stacked** and **aggregated** model structures on the CUB dataset.

The aggregated structure with new GAN advances achieve the best performance/compute-cost ratio in both the image quality and the image–text consistency. Moreover, we find that the abandoned lower-resolution  $D$ s in StackGAN have limited effect on image–text consistency, which instead appears more related to the generated image quality, as a higher IS always yields a better R-precision.

**Attention Mechanisms** We conduct experiments to explore the best attention settings for the T2I task from the mechanisms discussed in Section 4.1.

	Inception Score $\uparrow$	R-precision $\uparrow$
AttnGAN	$4.36 \pm 0.03$	$67.82 \pm 4.43$
Tf-h1-1l	$4.38 \pm 0.06$	$66.96 \pm 5.21$
Tf-h4-1l	$4.42 \pm 0.06$	$68.58 \pm 4.39$
Tf-h4-12	$4.48 \pm 0.03$	$69.72 \pm 4.23$
Tf-h4-14	$4.33 \pm 0.02$	$67.42 \pm 4.31$
Tf-h8-14	$4.28 \pm 0.03$	$62.32 \pm 4.25$

Table 2: Comparison of different attention settings on CUB.

Table 2 lists the settings we tested, where all the models are configured the same based on AttnGAN, except for the attention mechanisms used in  $G$ . In particular, column 1 shows the baseline performance that employs the basic attention operation, described in Fig. 3-(a), from AttnGAN. The following columns show the results of using the Transformer illustrated in Fig. 3-(d) with different numbers of heads and layers (e.g., Tf-h4-12 means a Transformer with 4 heads and 2 layers). The results suggest that a Transformer with a more comprehensive attention yields a better performance than the baseline. However, when increasing the number of layers and heads beyond a threshold, a clear performance degradation emerges on the CUB dataset. More discussion and the results on MS COCO can be found in the appendix.

**Controllable G without Sentence-Embedding** Most previous T2I models rely on a sentence-level embedding  $f_s$  as a vital conditioning factor for  $G$  (Zhang et al. 2017; Xu et al. 2018; Qiao et al. 2019; Zhu et al. 2019; Li et al. 2019a).

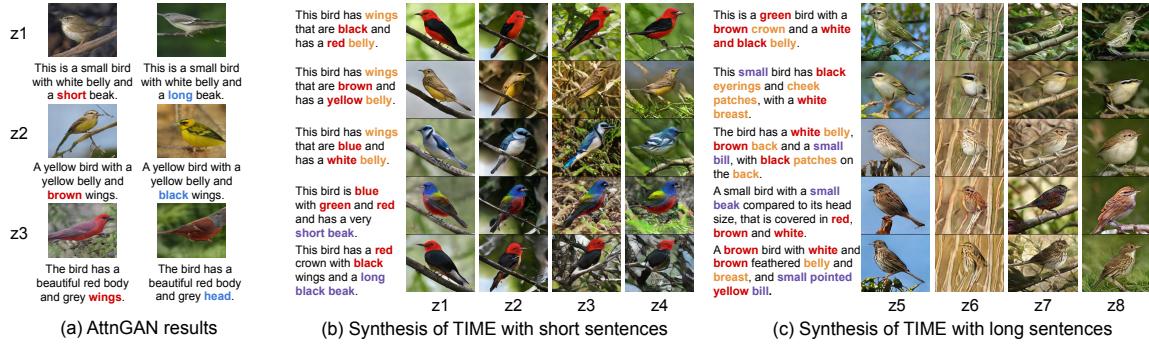


Figure 6: Images from TIME with fixed  $z$  and varied sentences

Specifically,  $f_s$  is concatenated with noise  $z$  as the input for  $G$ , and is leveraged to compute the conditional authenticity of the images in  $D$ . Sentence embeddings are preferred over word embeddings, as the latter lack contextual meaning and semantic concepts are often expressed in multiple words.

However, since  $f_s$  is a part of the input alongside  $z$ , any slight changes in  $f_s$  can lead to major visual changes in the resulting images, even when  $z$  is fixed. This is undesirable when we like the shape of a generated image but want to slightly revise it by altering the text description. Examples are given in Fig. 6-(a), where changing just a single word leads to unpredictably large changes in the image. In contrast, since we adopt the Transformer as the text encoder, where the word embeddings already come with contextual information,  $f_s$  is no longer needed in TIME. Via our Transformer text encoder, the same word in different sentences or at different positions will have different embeddings. As a result, the word embeddings are sufficient to provide semantic information, and we can abandon the sentence embedding.

In Fig. 6-(b) and (c), TIME shows a more controllable generation when changing the captions while fixing  $z$ . TIME provides a new perspective that naturally enables fine-grained manipulation of synthetic images via their text descriptions.

**Ablation Study** Based on AttnGAN objectives, we combine the model settings from Table 1 row.7 and Table 3 row.5 as the baseline, and perform ablation study in Table 3. First, we remove the image captioning text decoder  $Dec$  to show its positive impact. Then, we add  $Dec$  back and show that dropping the sentence-level embedding does not hurt the performance. Adding 2-D positional encoding brings improvements in both image–text consistency and the overall image quality. Lastly, the proposed hinge loss  $L_{hinge}$  (eq. 4) releases  $D$  from a potentially conflicting signal, resulting in the most substantial boost in image quality.

To emphasize the contribution of the proposed image-text hinge loss  $L_{hinge}$ , we conduct ablation study of it with different annealing schedules, including: stop training on  $L_{hinge}$  after 400 epoches (early-stop), start training on  $L_{hinge}$  after 100 epoches (late-begin), and annealing  $L_{hinge}$  with a constant factor 1. Fig. 7 records the model performance along the training iterations. Firstly, it shows the effectiveness of the proposed  $L_{hinge}$  within all the anneal schedules. Moreover, early-stop leads to a direct performance downgrade in

	Inception Score $\uparrow$	R-precision $\uparrow$
Baseline	$4.64 \pm 0.03$	$70.72 \pm 1.43$
B - img captioning	$4.58 \pm 0.02$	$69.72 \pm 1.43$
B - Sentence emb	$4.64 \pm 0.06$	$68.96 \pm 2.21$
B + 2D-Pos Encode	$4.72 \pm 0.06$	$71.58 \pm 2.39$
B + Hinged loss	$4.91 \pm 0.03$	$71.57 \pm 1.23$

Table 3: Ablation Study of TIME on CUB dataset

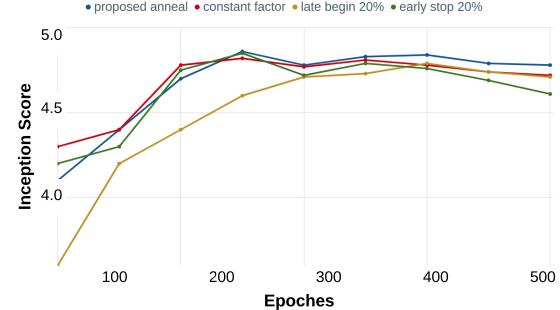


Figure 7: Performance comparison on different annealing schedules of the hinged image-text consistency loss.

late iterations, while late-begin performs the worst in early iterations. Annealing with a constant factor yields a similar performance as the dynamic annealing in early iterations, but falls back later when the models converge.

**Language Model Performance** Apart from a strong T2I performance, TIME also yields  $D$  as a well-performed stand-alone image captioning model.

model	Captioning BLEU-4 $\uparrow$	Image Retrieval@5 $\uparrow$	Text Retrieval@5 $\uparrow$
Bert Basic (with 24 TF)	0.389	69.3	82.2
UNITER (with 24 TF)	0.395	76.7	87.0
OSCAR (with 24 TF)	0.405	80.8	91.1
TIME (with 8 TF)	0.381	75.1	78.2

Table 4: Results on downstream Vision-Language tasks from TIME on COCO, compared with SOTA models.

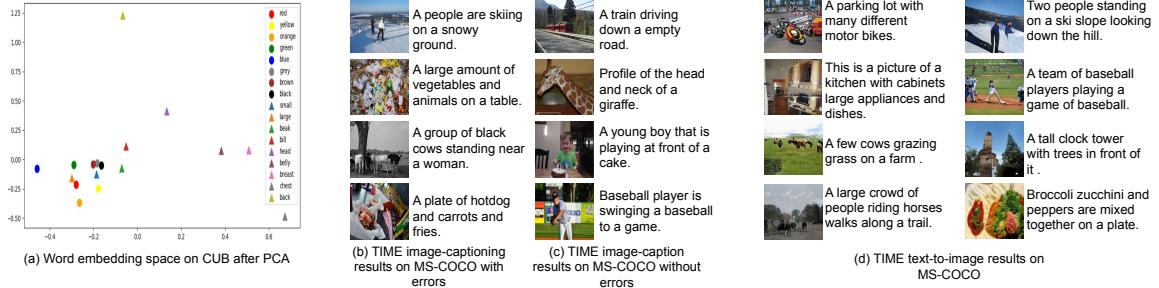


Figure 8: Learned word embeddings on CUB, and qualitative results on MS-COCO

		StackGAN	AttnGAN	ControlGAN	MirrorGAN	DMGAN	TIME	Real-Image
CUB	Inception Score $\uparrow$	3.82 $\pm$ 0.06	4.36 $\pm$ 0.03	4.51 $\pm$ 0.06	4.56 $\pm$ 0.05	4.71 $\pm$ 0.02	<b>4.91</b> $\pm$ 0.03	5.04
	FID $\downarrow$	N/A	23.98	N/A	N/A	16.09	<b>14.3</b>	0
	R-precision $\uparrow$	10.37 $\pm$ 5.88	67.82 $\pm$ 4.43	69.33 $\pm$ 3.21	69.58 $\pm$ 4.39	72.31 $\pm$ 0.91	71.57 $\pm$ 1.2	N/A
COCO	Inception Score $\uparrow$	8.45 $\pm$ 0.03	25.89 $\pm$ 0.47	24.06 $\pm$ 0.6	26.47 $\pm$ 0.4	30.49 $\pm$ 0.5	<b>30.85</b> $\pm$ 0.7	36.5
	FID $\downarrow$	N/A	35.49	N/A	N/A	32.64	<b>31.14</b>	0
	R-precision $\uparrow$	N/A	83.53 $\pm$ 0.43	82.43 $\pm$ 2.21	84.21 $\pm$ 0.39	91.87 $\pm$ 0.28	89.57 $\pm$ 0.9	N/A
	SOA-C $\uparrow$	N/A	25.88	25.64	27.52	<b>33.44</b>	32.78	74.97

Table 5: Text-to-Image performance comparison between TIME and other models.

Table 4 shows the comparison between TIME and more complicated NLP models, reflects the practicality and competence of TIME on the more general Vision-Language (VL) tasks. Note that we compete with Bert (Dai et al. 2019; Pan et al. 2020), UNITER (Chen et al. 2019) and OSCAR (Li et al. 2020) which all are large scale models with 24 Transformer (TF) blocks, and pre-trained on multiple VL tasks for an optimal performance.

In contrast, TIME is only trained on the studied text-image mutual translation task, with a tiny model size (only 8 TF blocks) and without any pre-training. It gains close performance to the SOTA models, which reveals a promising area for future research towards mutual-translation in a single framework. Fig. 8 shows the qualitative results from TIME on language tasks. In Fig. 8-(a), words with similar meanings reside close to each other. “Large” ends up close to “red”, as the latter often applies to large birds, while “small” is close to “brown” and “grey”, which often apply to small birds.

**Comparison on T2I with State-of-the-Arts** We next compare TIME with several SOTA text-to-image models. Qualitative results of TIME can be found in Figs. 1, 6, and 8. On CUB, TIME yields a more consistent image synthesis quality, while AttnGAN is more likely to generate failure samples. On MS-COCO, where the images are much more diverse and complex, TIME is still able to generate the essential contents that is consistent with the given text. The overall performance of TIME proves its effectiveness, given that it also provides image captioning besides T2I, and does not rely on any pre-trained modules.

As shown in Table 5, TIME demonstrates competitive performance on MS-COCO and CUB datasets with the new state-of-the-art IS and FID. Unlike the other models that require a well pre-trained language module and an Inception-v3 image encoder, TIME itself is sufficient to learn the cross-

modal relationships between image and language. Regarding the image-text consistency performance, TIME is also among the top performers on both datasets. Specifically, we do not tune the model structure to get an optimal performance on MS-COCO dataset. As our text decoder in  $D$  performs image captioning with image feature-map of size  $8 \times 8$ , such a size choice may not able to capture small objects in images from MS-COCO. On the other hand,  $8 \times 8$  is the suitable size to capture features of bird parts, for images from CUB dataset.

Importantly, TIME is considerably different from AttnGAN (no pre-training, no extra CNN/RNN modules, no stacked structure, no sentence embedding), while the other models all based on AttnGAN with orthogonal contributions to TIME. These technique contributions (e.g. DMGAN, SD-GAN, OP-GAN) could also be incorporated into TIME, with foreseeable performance boost. Due to the content limitation, we omit the integration of such ideas.

## 6 Conclusion

In this paper, we propose the Text and Image Mutual-translation adversarial nEtwork (TIME), a unified framework trained with an adversarial schema that accomplishes both the text-to-image and image-captioning tasks. Via TIME, we provide affirmative answers to the four questions we raised in Section 1. While previous works in the T2I field require pre-training several supportive modules, TIME achieves the new state-of-the-art T2I performance without pre-training. The joint process of learning both a text-to-image and an image-captioning model fully harnesses the power of GANs (since in related works,  $D$  is typically abandoned after training  $G$ ), yielding a promising Vision-Language performance using  $D$ . TIME bridges the gap between the visual and language domains, unveiling the immense potential of mutual translations between the two modalities within a single model.

## Bibliography

- Cai, Y.; Wang, X.; Yu, Z.; Li, F.; Xu, P.; Li, Y.; and Li, L. 2019. Dualattn-GAN: Text to Image Synthesis With Dual Attentional Generative Adversarial Network. *IEEE Access* 7: 183706–183716. ISSN 2169-3536.
- Chen, Y.-C.; Li, L.; Yu, L.; Kholy, A. E.; Ahmed, F.; Gan, Z.; Cheng, Y.; and Liu, J. 2019. Uniter: Learning universal image-text representations. *arXiv preprint arXiv:1909.11740*.
- Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q. V.; and Salakhutdinov, R. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.
- Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.
- Han, F.; Guerrero, R.; and Pavlovic, V. 2019. The art of food: Meal image synthesis from ingredients. *arXiv preprint arXiv:1905.13149*.
- 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. In *Advances in neural information processing systems*, 6626–6637.
- Hinz, T.; Heinrich, S.; and Wermter, S. 2019. Semantic Object Accuracy for Generative Text-to-Image Synthesis. *arXiv preprint arXiv:1910.13321*.
- Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8): 1735–1780.
- Huang, Q.; Zhang, P.; Wu, D.; and Zhang, L. 2018. Turbo learning for captionbot and drawingbot. In *Advances in neural information processing systems*, 6455–6465.
- Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.
- Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4401–4410.
- Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Li, B.; Qi, X.; Lukasiewicz, T.; and Torr, P. 2019a. Controllable Text-to-Image Generation. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d’Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems* 32, 2063–2073. Curran Associates, Inc.
- URL <http://papers.nips.cc/paper/8480-controllable-text-to-image-generation.pdf>.
- Li, W.; Zhang, P.; Zhang, L.; Huang, Q.; He, X.; Lyu, S.; and Gao, J. 2019b. Object-driven text-to-image synthesis via adversarial training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 12174–12182.
- Li, X.; Yin, X.; Li, C.; Hu, X.; Zhang, P.; Zhang, L.; Wang, L.; Hu, H.; Dong, L.; Wei, F.; et al. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. *arXiv preprint arXiv:2004.06165*.
- Lim, J. H.; and Ye, J. C. 2017. Geometric gan. *arXiv preprint arXiv:1705.02894*.
- 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.
- Mansimov, E.; Parisotto, E.; Ba, J. L.; and Salakhutdinov, R. 2015. Generating images from captions with attention. *arXiv preprint arXiv:1511.02793*.
- Mescheder, L.; Geiger, A.; and Nowozin, S. 2018. Which training methods for GANs do actually converge? *arXiv preprint arXiv:1801.04406*.
- Mikolov, T.; Karafiat, M.; Burget, L.; Černocky, J.; and Khudanpur, S. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.
- Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2020. Nerf: Representing scenes as neural radiance fields for view synthesis. *arXiv preprint arXiv:2003.08934*.
- Mirza, M.; and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.
- Nguyen, A.; Clune, J.; Bengio, Y.; Dosovitskiy, A.; and Yosinski, J. 2017. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4467–4477.
- Pan, Y.; Yao, T.; Li, Y.; and Mei, T. 2020. X-Linear Attention Networks for Image Captioning. *ArXiv abs/2003.14080*.
- Qiao, T.; Zhang, J.; Xu, D.; and Tao, D. 2019. Mirrorgan: Learning text-to-image generation by redescription. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1505–1514.
- Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multi-task learners. *OpenAI Blog* 1(8): 9.
- Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Schiele, B.; and Lee, H. 2016. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*.
- Reed, S.; van den Oord, A.; Kalchbrenner, N.; Colmenarejo, S. G.; Wang, Z.; Chen, Y.; Belov, D.; and De Freitas, N. 2017. Parallel multiscale autoregressive density estimation. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2912–2921. JMLR. org.

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

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Tay, Y.; Bahri, D.; Metzler, D.; Juan, D.-C.; Zhao, Z.; and Zheng, C. 2020. Synthesizer: Rethinking Self-Attention in Transformer Models. *arXiv preprint arXiv:2005.00743* .

Tran, D.; Ranganath, R.; and Blei, D. M. 2017. Deep and hierarchical implicit models. *arXiv preprint arXiv:1702.08896* 7: 3.

Van den Oord, A.; Kalchbrenner, N.; Espeholt, L.; Vinyals, O.; Graves, A.; et al. 2016. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, 4790–4798.

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 *Advances in neural information processing systems*, 5998–6008.

Welinder, P.; Branson, S.; Mita, T.; Wah, C.; Schroff, F.; Belongie, S.; and Perona, P. 2010. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology.

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.

Yin, G.; Liu, B.; Sheng, L.; Yu, N.; Wang, X.; and Shao, J. 2019. Semantics disentangling for text-to-image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2327–2336.

Zhang, H.; Goodfellow, I.; Metaxas, D.; and Odena, A. 2018. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318* .

Zhang, H.; Xu, T.; Li, H.; Zhang, S.; Wang, X.; Huang, X.; and Metaxas, D. N. 2017. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 5907–5915.

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 Conference on Computer Vision and Pattern Recognition*, 5802–5810.

## Appendix

### A Broader Impact

Generating realistic images from text descriptions is an important task with a wide range of real-world applications, such as reducing repetitive tasks in story-boarding, and film or video game scene editing. However, such technology may also lead to the spread of misinformation via synthesized images, and may enable infringement via massive re-production and simple manipulation of copyrighted images.

### B Implementation Details

The code that re-produces the results of TIME along with trained model checkpoints and pre-processed data are available at: <https://drive.google.com/drive/folders/1tMgEB7F7ZH0J2lfaPOsowwOJgq8ygUDT?usp=sharing>. Please check the detailed training configurations in the code. A “readme” file is as well included to link the proposed modules to the code snippets. Our training environment consists of two NVIDIA TITAN RTX GPUs, each of which has 24 GB memory, with the CUDA-10.1 driver and PyTorch framework running on the Ubuntu-19.10 operating system. During training, not all GPU capacity is necessarily used.

### C Vanilla Conditional Loss

Apart from the proposed image-captioning loss, which serves as a conditioning factor to train  $D$  and  $G$ , we also applied another simple conditional loss by combining the image feature and text embeddings (this vanilla conditional loss is reflected in Fig. 2 upper-left). Such conditional loss serves as a basic regulation term for  $D$ , just like the conditional loss used in StackGAN.

In practice, we can extract  $8 \times 8$  or  $4 \times 4$  image features from the CNN part of  $D$ , and concatenate the image features with the reshaped word embeddings along the channel dimension. Specifically, we re-shape the word embeddings by fitting each of them into one pixel location of an image feature map. For example, if we have an image feature with the shape of  $512 \times 4 \times 4$  ( $4 \times 4$  spatial size with 512 channels), and a text description of 16 words, where each word is represented in 512 dimensions, we reshape the word embeddings into  $512 \times 4 \times 4$ , where the top-left spatial location is the first embedding, the second top-left location is the second embedding, and the bottom-right contains the last embedding, etc. The image features and the reshaped word embeddings can then be concatenated into a image-context feature with the shape of  $1024 \times 4 \times 4$ . Such image-context feature is then processed by two convolution layers, giving rise to the final image-text matching score.

When the word sequence is longer than 16, we just select the first 16 words, and when the sequence is less than 16, we use 0-padding for the remaining space. This sort of procedure is fairly naive, and can potentially be improved. However, this is not the primary concern in this work and we find it already works sufficiently well across all of our experiments. Since using such a conditional loss is a common practice in prior works, we only conduct a simple ablation experiment on TIME and AttnGAN to confirm the effect of this loss. The

results shows a consistent small performance gain for both TIME and AttnGAN, where abandoning such a conditional loss only leads to about 2% IS loss on the CUB dataset. As a result, we pick this conditional loss for all models. Further details of the implementation can be found in the code.

### D 2-D Positional Encoding

The formal equation of the 2-D positional encoding is:

$$P_{i \in [1: \frac{c}{4}]}(y, 2i) = \sin\left(\frac{y}{10000^{\frac{4i}{c}}}\right); \quad (8)$$

$$P_{i \in [\frac{c}{4}: \frac{c}{2}]}(x, 2i) = \cos\left(\frac{x}{10000^{\frac{2i}{c}}}\right); \quad (9)$$

$$P_{i \in [1: \frac{c}{4}]}(y, 2i - 1) = \cos\left(\frac{y}{10000^{\frac{4i}{c}}}\right); \quad (10)$$

$$P_{i \in [\frac{c}{4}: \frac{c}{2}]}(x, 2i - 1) = \sin\left(\frac{x}{10000^{\frac{2i}{c}}}\right), \quad (11)$$

where  $x, y$  are the coordinates of each pixel location, and  $i$  is the dimension index along the channel.

One interesting question about 2-D positional encoding is: “if we cut off the boundary parts of the input image, which leads to major value changes of the position encoding on all pixels, how will it affect the captioning result?”

The purpose of position encoding is to re-value the feature vectors based on each spatial location, making the attention operation better at attending to the related positions. We believe the “relative value” between each spatial location is the key and it should be invariant in cases such as the one just described. I.e., the neighbor pixels’ **relative value difference** should not be changed even when the boundary pixels are cut off. Therefore, each embedding will still attend to its most relative embeddings that have similar embedding values, so the attention result and the resulting captions will show little change. We conducted experiments that confirmed our hypothesis, and showed that the 2D-positional-encoding based attention operation has such a “content invariant” property.

### E Controllability

The reason we gain controllability with TIME is fairly intuitive. Here, we extend the discussion in the paper on “Controllable G without Sentence-Level Embedding” with more implementation details. Unlike previous T2I models, the text information is introduced to the model only on high resolution feature-maps (e.g., 32x32 and 64x64 as illustrated in Fig. 2 of the paper), thus it is unlikely to change the content shape. This intuition is validated in many GAN works, a typical example being StyleGAN, where changing the input noise at the  $8 \times 8$  feature level leads to major changes in content, while changing the input noise at the  $32 \times 32$  level does not change the shapes in the image at all, but instead only leads to stylistic changes such as the coloring. Our experiments also confirmed this observation: when we tested feeding text embeddings to a lower-resolution layer (e.g.,  $8 \times 8$ ), the model controllability indeed disappeared.

Note that no additional tweaks were used (extra loss, extra labeled data) to gain such controllability, and thus the degree of controllability is highly biased on the data. Taking the

CUB dataset as an example, although the paired text description only describes how the bird looks like, there is a high correlation between the color of birds and the background, e.g., yellow birds tend to be found more in green trees and white birds more in the blue sky. Such “background & bird-style” correlation can easily be learned by the model without any extra guidance. As a result, the background color also changes when we change text that merely describes the bird, and hence the degree of controllability pertains more to the invariant shape of the bird but not to an invariant background.

## F Transformer configuration exploration

We conducted the same experiments to explore the best attention settings for the T2I task on the MS COCO dataset. Table 6 lists the settings we tested, where all the models are configured in the same way based on AttnGAN, except for the attention mechanisms used in  $G$ .

Table 6: Comparison between different attention settings on MS COCO dataset

	AttnGAN	Tf-h1-11	Tf-h4-11	Tf-h4-12	Tf-h4-14	Tf-h8-14
Inception Score $\uparrow$	25.89	26.58	26.42	27.48	27.85	27.153
R-precision $\uparrow$	83.53	86.46	88.58	89.72	89.57	88.32

The results are consistent with the experiment on the CUB dataset in showing that the “multi-head and multi-layer” (MHL) design boosts the performance. We hypothesize that the optimal number of heads and layers depends on the dataset, where the “4-heads 2-layers” and “4-heads-4-layers” settings are the sweet points for the CUB and COCO datasets, respectively. While the motivation and design of MHL is quite intuitive, it is still possible that the performance gain results from the increased number of parameters. This suspicion is beyond this work. On the other hand, as shown in the last two columns, making the Transformer too “big” leads to worse performance. A possible reason is that the increase in parameters makes it harder for the model to converge, and makes it more susceptible to overfitting the training data.

## G More on Related Work

In this section, we want to highlight the differences between TIME and previous work.

### G.1 StackGAN as the Image Generation Backbone

StackGAN adopts a coarse-to-fine structure that has shown substantial success on the T2I task. The generator  $G$  takes three steps to produce a  $256 \times 256$  image as shown in Figure 9-(a). In stage-I, a  $64 \times 64$  image with coarse shapes is generated. In stage-II and III, the feature maps are further up-sampled to produce more detailed images with better textures. Three discriminators are required, where the lowest-resolution  $D$  guides  $G$  with regard to coarse shapes, while localized defects are refined by the higher-resolution  $Ds$ .

However, there are several reasons for seeking an alternative architecture. First, the multi- $D$  design is memory-demanding and has a high computational burden during training. As the image resolution increases, the respective higher-

resolution  $Ds$  can raise the cost dramatically. Second, it is hard to balance the effects of the multiple  $Ds$ . Since  $Ds$  are trained on different resolutions, their learning paces diverge, and can result in conflicting signals when training  $G$ . In our experiments, we notice a consistently slower convergence rate of the stacked structure compared to a single- $D$  design.

### G.2 Dependence on Pre-trained modules

While the overall framework for T2I models resembles a conditional GAN (cGAN), multiple modules have to be pre-trained in previous works. As illustrated in Figure 9-(b), AttnGAN requires a DAMSM, which includes an Inception-v3 model (Szegedy et al. 2016) that is first pre-trained on ImageNet (Deng et al. 2009), and then used to pre-train an RNN text encoder. MirrorGAN further proposes a pre-trained STREAM module as shown in Fig. 9-(c).

Such pre-training has a number of drawbacks, including, first and foremost, the computational burden. Second, the additional pre-trained CNN for image feature extraction introduces a significant amount of weights, which can be avoided as we shall later show. Third, using pre-trained modules leads to extra hyper-parameters that require dataset-specific tuning. For instance, in AttnGAN, the weight for the DAMSM loss can range from 0.2 to 100 across different datasets. While these pre-trained models boost the performance, empirical studies in MirrorGAN and StackGAN show that they do not converge if jointly trained with cGAN.

### G.3 Difference to MirrorGAN

Our model shares a similar idea with MirrorGAN in “matching the image captions for a better text-to-image generation consistency”. However, there are substantial difference between our work and MirrorGAN. First, while MirrorGAN focused on the T2I task and uses a fixed captioning model to support the training of  $G$ , our idea can better be described as “image–text mutual translation”. The objective is not merely to train a T2I  $G$ , but also to train an image captioning model at the same time. Second, the study in MirrorGAN suggests that the image-captioning module with extra CNN and RNN has trouble converging if trained together with the T2I GAN, and thus can only be pre-trained. However, in our work, we show that the image-captioning module can be jointly trained under the same framework, and the joint training leads to better performance for  $G$  while does not hurt the performance on captioning for  $D$ . Third, we believe “image–text mutual translation” is a topic that can attract substantial research attention and that it is particularly intriguing, as humans can easily relate visual and lingual information together.

TIME differs from MirrorGAN in several respects to realize the joint training. i) MirrorGAN uses LSTMs to model the language, which suffers from vanishing gradient issues, in particular that the discrete word embeddings are not very compatible with image features, which are more continuous when trained jointly. In contrast, Transformer is more robust in dealing with the combined features of the text and image modalities. ii) MirrorGAN tries to train directly on word embeddings, which is common in T2I tasks. In contrast, in our model, the word embeddings that the CNNs ( $G$  and  $D$ ) take are not the initial word embeddings. We have a text encoder

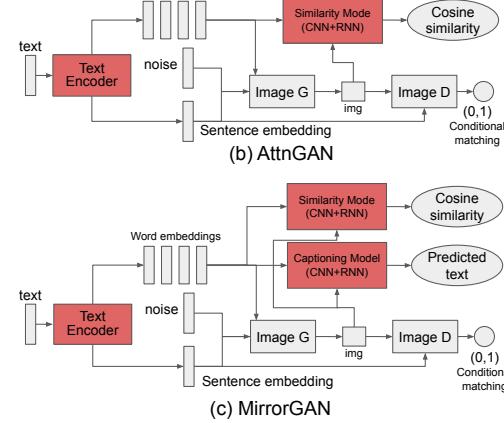
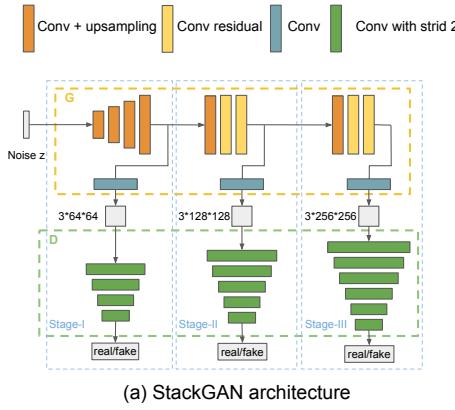


Figure 9: (a) The StackGAN structure that serves as the backbone in SOTA T2I models (Zhang et al. 2017; Xu et al. 2018; Qiao et al. 2019; Zhu et al. 2019; Cai et al. 2019; Li et al. 2019a; Hinz, Heinrich, and Wermter 2019). (b)&(c) Representative models build upon StackGAN, with red parts indicating modules that require pre-training. Note that our proposed model TIME does not: 1. require pre-training the red modules in (b) and (c); 2. require multiple discriminators (the green modules) in (a).

to endow the word sequence embeddings with contextual signals, making the resulting sentence representations much smoother, thereby facilitating training. iii) The language part of TIME is only trained with the discriminator on the image captioning task. The embeddings are not trained to help the Generator for better image quality. This focused objective frees the text encoder from potentially conflicting gradients, aiding convergence, and making mutual translation feasible with our model.

#### G.4 Contribution Report

In this paper, we explored the effectiveness of the “Multi-Layer and Multi-Head” (MLH) attention design based on a Transformer in the T2I task. Note that its use is definitely not meant for novelty. Moreover, it is still an open problem to fully understand how and why the attention mechanism works (see e.g. new discussions in (Tay et al. 2020)). Our aim is not to try to explore that problem, but instead we seek to know how the more comprehensive Transformer structure benefits generative image models compared to the currently used “single-layer and single-head” attention in models such as AttnGAN and SAGAN.

Apart from the MLH design, we would like to emphasize the “novelty” aspects in our work as follows: we propose 1. a training strategy for T2I without pre-training, 2. a 2-D positional encoding, 3. annealing hinge loss for adversarial training, 4. a new generator structure that is more lightweight and controllable, 5. a “text–image” mutual translation perspective of learning a model to bridge the image and text modalities. Importantly, the work of this paper was started in early 2019 and first submitted for peer review on March 2020 (withdrawn), one may find some of the claimed contributions not novel on the date of publication in 2021. To our best research on up-to-date related works, the integration of transformer into image models, and our aforementioned point 2. and 4. have also been studied in other works (Mildenhall et al. 2020), while the rest points remain unique and novel.

## H Image Captioning Qualitative Result



Figure 10: Image captioning results of TIME on the CUB dataset testing set.

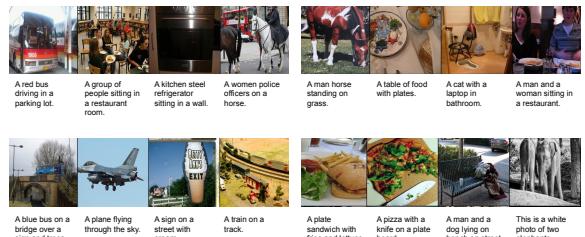


Figure 11: Image captioning results of TIME on the MS-COCO dataset testing set.

## I Text to Image Qualitative Results

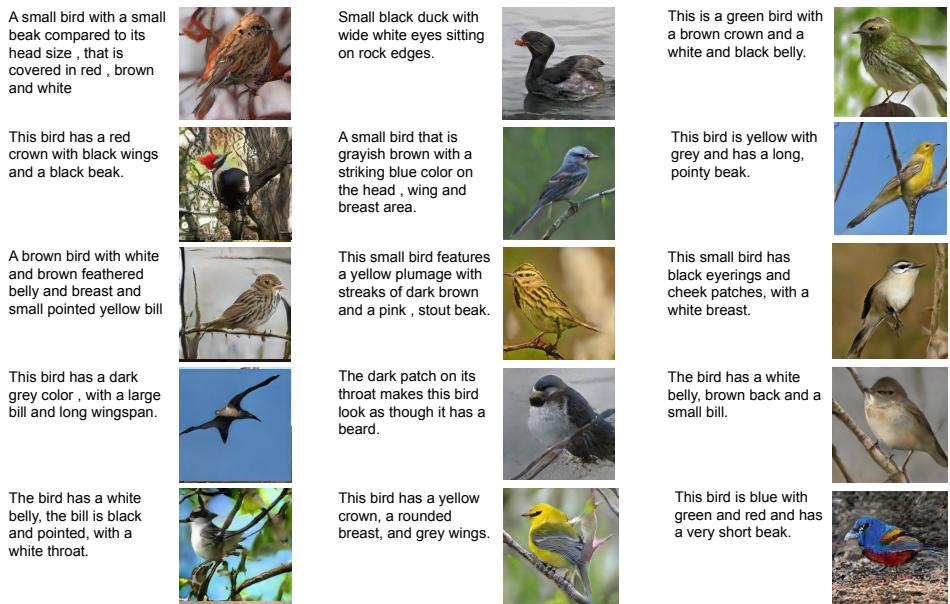


Figure 12: Uncurated Qualitative results of TIME on the CUB dataset: The images are generated by TIME given the captions from testing dataset.

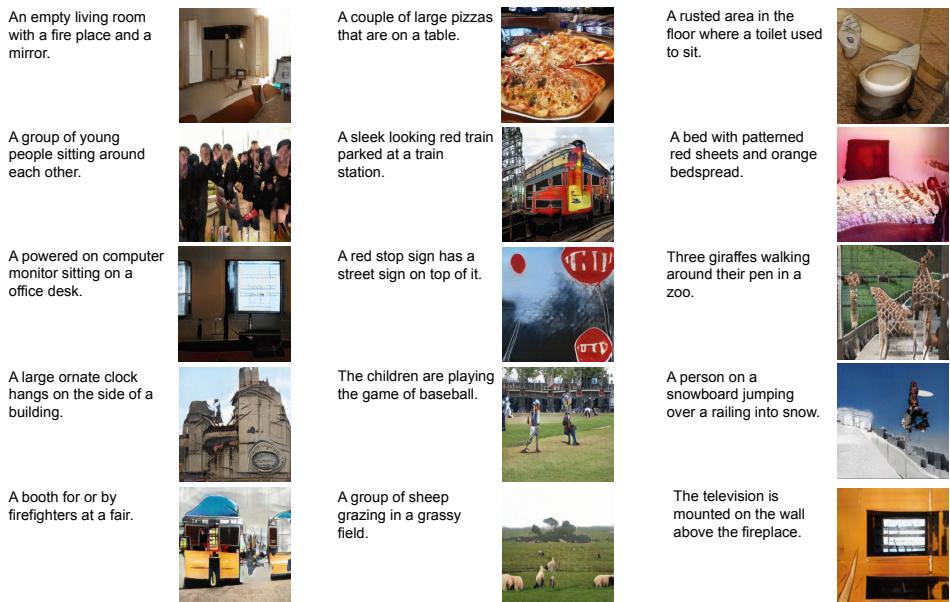


Figure 13: Uncurated Qualitative results of TIME on the MS-COCO dataset: The images are generated by TIME given the captions from testing dataset.



Figure 14: Uncurated Qualitative results of TIME on the CUB dataset: The images are generated by TIME given the captions from testing dataset, each row containing the generates samples from the same caption.



Figure 15: Uncurated Qualitative results of TIME on the MS-COCO dataset: The images are generated by TIME given the captions from testing dataset.