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, kunpeng.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 dataset (Inception Score of 4.91 and Fréchet Inception Distance of 14.3 on CUB), and shows promising performance on MS-COCO dataset on image captioning and downstream vision-language tasks.

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 Ds 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 an 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 Wermer 2019), the follow-up methods all share two common traits. First, they all adopt the same stacked structure of G that requires multiple Ds. 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 work 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 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.

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 (height \(\times\) width \(\times\) channel) in the CNN are flattened into a two-dimensional sequence (seq-length \(\times\) channel where seq-length = height \(\times\) 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.

**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 suitable pre-trained NLP model is not always available for a given image dataset. In cases where the given texts do not have a pre-trained NLP model, one can save the separate pre-training time and learn a model that translates in both 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 are learned together with $G$ and dynamically adjust the word embeddings for the training objective, leading to better image synthesis quality.

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

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-to-Text (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 $T$ along with: (a) matched real images $I_{\text{match}}$, (b) randomly mismatched real images $I_{\text{mis}}$, and (c) generated images $I_{\text{fake}}$ from $G$.

Model Structures

Text-Conditioned Image Transformer While prior studies (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 illustrate three main differences between TCIT and the form of attention widely used in previous T2I models such as AttnGAN. All attention modules take two inputs, consisting of captions $T$ alongside: (a) matched real images $I_{\text{match}}$, (b) randomly mismatched real images $I_{\text{mis}}$, and (c) generated images $I_{\text{fake}}$ from $G$.
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 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.

**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 \(f_i\) is revised by \(f_t\), 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.

**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 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 online appendix for further details.

**Objectives**

**Discriminator Objectives** Formally, we denote the three kinds of outputs from \(D\) as: \(D_t()\), 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: \(P_k = Dec(Enc(T_{real}^k), D_t(I_{match}))\). Finally, the objectives for \(D, Enc, \) and \(Dec\) to jointly minimize are:

\[
L_{caption} = - \sum_{k=1}^{l} \log(P_k(T_{real}^k, D_t(I_{match}))); \quad (1)
\]

\[
L_{uncond} = - \mathbb{E}[\log(D_u(I_{match}))] - \mathbb{E}[\log(1 - D_u(I_{fake}))]; \quad (2)
\]
very early iterations. As shown in Fig. 5, while the convention

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

along with \( L_{\text{cond}} \), 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:

\[
s_{\text{pivot}} = \text{detach}(\mathbb{E}[D_c(I_{\text{match}}, \text{Enc}(T_{\text{real}}))]);
\]

\[
L_{\text{cond}} = \mathbb{E}[(\min(0, 1 - D_c(I_{\text{match}}, \text{Enc}(T_{\text{real}}))))
+ \mathbb{E}[(\min(0, 1 + D_c(I_{\text{mismatch}}, \text{Enc}(T_{\text{real}}))))
+ \mathbb{E}[\min(0, -s_{\text{pivot}} \times p + D_c(I_{\text{fake}}, \text{Enc}(T_{\text{real}})))]).
\]

Here, \( \text{detach}() \) denotes that the gradient is not computed for the enclosed function, and \( p = \frac{\text{epoch}}{\text{epochs}} \) (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 \( \text{Enc} \) 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:

\[
L_{\text{caption-g}} = -\sum_{k=1}^{l} \log(P_k(T_{\text{real}}^c, D_t(G(z, \text{Enc}(T_{\text{real}}))));
\]

\[
L_{\text{uncond-g}} = -\mathbb{E}[\log(D_u(G(z, \text{Enc}(T_{\text{real}}))))];
\]

\[
L_{\text{cond-g}} = -\mathbb{E}[D_c(G(z, \text{Enc}(T_{\text{real}})), \text{Enc}(T_{\text{real}}))].
\]

**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 (Weninger 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 Wermt 2019).

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

<table>
<thead>
<tr>
<th></th>
<th>Inception Score ↑</th>
<th>R-precision ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttnGAN</td>
<td>4.36 ± 0.03</td>
<td>67.82 ± 4.43</td>
</tr>
<tr>
<td>Tf-h1-1l</td>
<td>4.38 ± 0.06</td>
<td>66.96 ± 5.21</td>
</tr>
<tr>
<td>Tf-h4-1l</td>
<td>4.42 ± 0.06</td>
<td>68.58 ± 4.39</td>
</tr>
<tr>
<td>Tf-h4-12</td>
<td>4.48 ± 0.03</td>
<td><strong>69.72 ± 4.23</strong></td>
</tr>
<tr>
<td>Tf-h4-l4</td>
<td>4.33 ± 0.02</td>
<td>67.42 ± 4.31</td>
</tr>
<tr>
<td>Tf-h8-l4</td>
<td>4.28 ± 0.03</td>
<td>62.32 ± 4.25</td>
</tr>
</tbody>
</table>

Table 1: Comparison of different attention settings on CUB.

Table 1 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 denotes a Transformer with 4 heads and 2 layers). The results suggest that a Transformer with a more comprehensive attention yields better results 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 corresponding results on MS-COCO can be found in the online 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). 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, which
yields word embeddings already reflecting the context, $f_x$ 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 representations suffice to capture pertinent 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** We consider our aggregated architecture with the setting from Table 3 row 5 and the AttnGAN objectives as the baseline, and perform an ablation study in Table 2. 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 $L_{\text{hinge}}$ (eq. 4) releases $D$ from a potentially conflicting signal, resulting in the most substantial boost in image quality.

<table>
<thead>
<tr>
<th></th>
<th>Inception Score $\uparrow$</th>
<th>R-precision $\uparrow$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>4.64 ± 0.03</td>
<td>70.72 ± 1.43</td>
</tr>
<tr>
<td>B - img captioning</td>
<td>4.58 ± 0.02</td>
<td>69.72 ± 1.43</td>
</tr>
<tr>
<td>B - Sentence emb</td>
<td>4.64 ± 0.06</td>
<td>68.96 ± 2.21</td>
</tr>
<tr>
<td>B + 2D-Pos Encode</td>
<td>4.72 ± 0.06</td>
<td>71.58 ± 2.39</td>
</tr>
<tr>
<td>B + Hinged loss</td>
<td>4.91 ± 0.03</td>
<td>71.57 ± 1.23</td>
</tr>
</tbody>
</table>

Table 2: Ablation Study of TIME on CUB dataset

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

![Figure 6: Images from TIME with fixed $z$ and varied sentences](image)

![Figure 7: Performance comparison on different annealing schedules of the hinged image-text consistency loss.](image)

![Table 3: Results on downstream Vision-Language tasks from TIME on COCO, compared with SOTA models.](table)
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 AttGAN 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 4, 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. As our text decoder in \( D \) performs image captioning with an image feature-map of size \( 8 \times 8 \), such a size choice may not be able to capture small objects in images from MS-COCO. In contrast, \( 8 \times 8 \) is a suitable size to capture features of bird parts for images from the CUB dataset.

Importantly, TIME is considerably different from AttnGAN (no pre-training, no extra CNN/RNN modules, no stacked structure, no sentence embedding), while other models based on AttnGAN have orthogonal contributions to TIME. Such technique contributions (e.g., DMGAN, SDGAN, OP-GAN) could also be incorporated into TIME, leading to likely performance boosts, though we consider such experiments beyond the scope of this paper.

Table 4: Text-to-Image performance comparison between TIME and other models.

<table>
<thead>
<tr>
<th></th>
<th>StackGAN</th>
<th>AttnGAN</th>
<th>ControlGAN</th>
<th>MirrorGAN</th>
<th>DMGAN</th>
<th>TIME</th>
<th>Real-Image</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CUB</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inception Score ↑</td>
<td>3.82 ± 0.06</td>
<td>4.36 ± 0.03</td>
<td>4.51 ± 0.06</td>
<td>4.56 ± 0.05</td>
<td>4.71 ± 0.02</td>
<td><strong>4.91 ± 0.03</strong></td>
<td>5.04</td>
</tr>
<tr>
<td>FID ↓</td>
<td>N/A</td>
<td>23.98</td>
<td>N/A</td>
<td>N/A</td>
<td>16.09</td>
<td><strong>14.3</strong></td>
<td>0</td>
</tr>
<tr>
<td>R-precision ↑</td>
<td>10.47 ± 5.88</td>
<td>67.82 ± 4.43</td>
<td>69.33 ± 5.21</td>
<td>69.58 ± 4.39</td>
<td>72.91 ± 0.91</td>
<td>71.57 ± 1.2</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>COCO</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inception Score ↑</td>
<td>8.45 ± 0.03</td>
<td>25.89 ± 0.47</td>
<td>24.06 ± 0.6</td>
<td>26.47 ± 0.4</td>
<td><strong>30.49 ± 0.5</strong></td>
<td>27.85 ± 0.7</td>
<td>36.5</td>
</tr>
<tr>
<td>FID ↓</td>
<td>N/A</td>
<td>35.49</td>
<td>N/A</td>
<td>N/A</td>
<td>33.72</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>R-precision ↑</td>
<td>83.53 ± 0.43</td>
<td>82.43 ± 2.21</td>
<td>84.21 ± 0.39</td>
<td>91.87 ± 0.28</td>
<td>89.57 ± 0.9</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>SOA-C ↑</td>
<td>N/A</td>
<td>25.88</td>
<td>25.64</td>
<td>27.52</td>
<td><strong>33.44</strong></td>
<td>32.78</td>
<td>74.97</td>
</tr>
</tbody>
</table>

Conclusion

In this paper, we propose the Text and Image Mutual-translation adversarial nEwork (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.


