

TISE: Bag of Metrics for Text-to-Image Synthesis Evaluation

Tan M. Dinh*, Rang Nguyen, and Binh-Son Hua

VinAI Research, Hanoi, Vietnam

Abstract. In this paper, we conduct a study on the state-of-the-art methods for text-to-image synthesis and propose a framework to evaluate these methods. We consider syntheses where an image contains a single or multiple objects. Our study outlines several issues in the current evaluation pipeline: (i) for image quality assessment, a commonly used metric, e.g., Inception Score (IS), is often either miscalibrated for the single-object case or misused for the multi-object case; (ii) for text relevance and object accuracy assessment, there is an overfitting phenomenon in the existing R-precision (RP) and Semantic Object Accuracy (SOA) metrics, respectively; (iii) for multi-object case, many vital factors for evaluation, e.g., object fidelity, positional alignment, counting alignment, are largely dismissed; (iv) the ranking of the methods based on current metrics is highly inconsistent with real images. To overcome these issues, we propose a combined set of existing and new metrics to systematically evaluate the methods. For existing metrics, we offer an improved version of IS named IS* by using temperature scaling to calibrate the confidence of the classifier used by IS; we also propose a solution to mitigate the overfitting issues of RP and SOA. For new metrics, we develop counting alignment, positional alignment, object-centric IS, and object-centric FID metrics for evaluating the multi-object case. We show that benchmarking with our bag of metrics results in a highly consistent ranking among existing methods that is well-aligned with human evaluation. As a by-product, we create AttnGAN++, a simple but strong baseline for the benchmark by stabilizing the training of AttnGAN using spectral normalization. We also release our toolbox, so-called TISE, for advocating fair and consistent evaluation of text-to-image models.

Keywords: language and vision, metrics, text-to-image synthesis

1 Introduction

The unprecedented growth of deep learning has sparked significant interest in tackling the vital vision-language task of text-to-image synthesis in recent years, with potential applications from computer-aided design, image editing with text-guided to image retrieval. This is a challenging task because of the wide semantic gap between two domains and the high many-to-many mapping (e.g., one text

* Corresponding author









Caption	DM-GAN	CPGAN	AttnGAN++	Real Images
Several plates of food include fry dough and salad.				
There are people standing on the sand at the beach.				
Inception Score	32.43	52.90	40.13	37.71
R-Precision	92.23	93.59	96.39	67.35
SOA-C	33.44	77.02	48.33	74.97
SOA-I	48.03	84.55	67.19	80.84

Fig. 1. Evaluating the text-to-image models is a challenging task. Many existing metrics are inconsistent especially for the case when an input sentence involves multiple objects. Values in red denote inconsistent evaluations, where the quantitative results are even higher than that of real photos, despite the fact that such generated images are not perceptually real.

caption can correspond to many image counterparts and vice versa). Many aspects of image synthesis, such as image fidelity, object relations, object counting have to be considered for generating complex scenes from a sentence.

In the past few years, key techniques for text-to-image synthesis are largely based on the evolution of generative adversarial networks (GANs) [8]. Tremendous achievements has been obtained in many domains, e.g. from unconditional image generation [19,20] to latent space mapping and manipulation [45,46]. Most of text-to-image synthesis approaches [55,24,49,57,62,25] are built upon GANs and jointly consider text and image features in the synthesis.

Despite excellent results have been achieved on particular datasets [33,53,27], the current evaluation pipeline is far from ideal. For single object case, image quality and text-image alignment are primary factors considered in a typical evaluation process. Some commonly evaluation metrics are Inception Score (IS) [44] and the Fréchet Inception Distance (FID) [12] for image fidelity and R-precision (RP) [55] for text-image alignment, which works well for most single-object cases. However, in complex scenes with multiple objects, adopting these metrics are not enough and causes some inconsistency issues. As can be seen in Figure 1, the ranking of GAN models based on the current metrics is not strongly correlated to their generated image qualities. The numbers reported from several GANs are even better than the one of corresponding real images, while it is clearly seen that the quality of generated images are still far from being real. Additionally, the existing evaluation system lacks the metrics for assessing other aspects like object fidelity, positional alignment, and counting alignment, among others. These aspects are critical in evaluating the performance of text-to-image models in the multi-object case. Furthermore, the absence of a unified evaluation tool-

box has resulted in inconsistent outcomes reported by different research works. These issues are also highlighted in the recent comprehensive survey [7], which raises a demand to devise a unified bag of metrics for text-to-image evaluation.

In this paper, we develop a systematic method for evaluating text-to-image synthesis approaches to tackle the challenges mentioned above. Our contributions are summarized as follows:

1. For existing metrics, we create IS* as an improved version of IS metric for image quality assessment, which alleviates the low confidence phenomenon due to miscalibrations in the pre-trained classifier used for IS. We also develop the robust versions for text relevance and object accuracy assessment (RP and SOA [13]) to mitigate their overfitting issues in multi-object case.
2. For new metrics, we develop O-IS and O-FID for object fidelity, PA for positional alignment, and CA for counting alignment to evaluate these lacking aspects in multi-object text-to-image synthesis.
3. Based on these metrics, we conduct a comprehensive, fair and consistent evaluation of the current state-of-the-art methods for both single- and multi-object text-to-image models.
4. Finally, we propose AttnGAN++, a simple but strong baseline that works well for both single- and multi-object scenarios. Our AttnGAN++ has competitive performance to current state-of-the-art text-to-image models.

On top of these contributions, we develop a *Python* assessment toolbox called **TISE** (**T**ext-to-**I**mage **S**ynthesis **E**valuation) implementing our bag of metrics in a unified way to facilitate, advocate fair comparisons and reproducible results for future text-to-image synthesis research. ¹

2 Background

Text-to-Image Synthesis is a vision-language task substantially benefit from the unprecedented evolutions of generative adversarial neural networks and language models. GAN-INT-CLS [42] is the first conditional GAN [29] designed for text-to-image generation, but images generated by GAN-INT-CLS only have 64×64 resolution. StackGAN and its successor StackGAN++ [60,61] enhanced the resolution of generated images by using a multi-stage architecture. These works, however, only consider sentence-level features for image synthesis; word-level features are completely dismissed, which causes poor image details. To fix this issue, an attention mechanism can be used to provide word-level features, notably used by AttnGAN [55] and DM-GAN [62], which significantly improves the generated image quality. Beyond modifying the network architecture, improving semantic consistency between image and caption is also an active research topic to gain better image quality. SD-GAN [57] and SE-GAN [49] guarantee text-image consistency by the Siamese mechanism; [36] proposes a

¹ TISE toolbox is available at <https://github.com/VinAIRResearch/tise-toolbox>.

text-to-image-to-text framework called MirrorGAN inspired by the cycle consistency, while [59,56] leverage contrastive learning in their text-to-image models. To improve the performance of model in the multi-object case, InferGAN [14] and Obj-GAN [24] introduce a two-step generation process including layout generation and image generation, while CPGAN [25] leverages the object memory features in developing the model. Regarding model scaling approach, DALL-E [39] and CogView [6] are two large scale text-to-image synthesis models with 12 and 4 billion parameters, respectively, synthesizing the image from the caption autoregressively by using a transformer [52] and VQ-VAE [40].

Evaluation. The rapid advancement of text-to-image generation necessitates the construction of a reliable and systematic evaluation framework to benchmark models and guide future research. However, assessing the quality of generative modeling tasks has proven difficult in the past [51]. Because none of the existing measures are perfect, it is usual to report many metrics, each of which assesses a different aspect. The performance assessment is even more challenging in the text-to-image synthesis task due to the multi-modal complexity of text and image, which motivates us to develop a new evaluation toolbox to compare text-to-image approaches fairly and confidently.

3 Single-Object Text-to-Image Synthesis

3.1 Existing Metrics

Most of existing metrics assess the quality of model based on two aspects: image quality and text-image alignment. For assessing the image quality of the model, Inception score (IS) [44] and Fréchet Inception Distance (FID) [12] are two common metrics. These metrics originally come from traditional GAN tasks for evaluating the image quality. For evaluating text-image alignment, R-precision [55] metric is utilized popularly.

Inception Score (IS) [44] leverages a pretrained Inception-v3 network [48] for calculating the Kullback-Leibler divergence (KL-divergence) between class-conditional distribution and class-marginal distribution of the generated images. The formula of IS is defined below.

$$\text{IS} = \exp(\mathbb{E}_x D_{KL}(p(y|x) \parallel p(y))), \quad (1)$$

where x is the generated image and y is the class label. The goal of this metric is to determine whether a decent generator can generate samples under two conditions: (i) The object in the image should be *distinct* $\rightarrow p(y|x)$ must have low entropy; (ii) Generated images should have the *diversity* of object class $\rightarrow p(y)$ must have high entropy. Combining these two considerations, we expect that the KL-divergence between $p(y)$ and $p(y|x)$ should be large. Therefore, higher IS value means better image quality and diversity.

Fréchet Inception Distance (FID) [12] calculates the Fréchet distance between two sets of images: generated and actual. To calculate FID, features from each set are firstly extracted by a pre-trained Inception-v3 network [48]. Then,

Table 1. Benchmark results for the single-object text-to-image synthesis models on the CUB dataset. In this benchmark, we only consider the methods, which have been released with officially source code and pre-trained weights by their authors. **Best** and runner-up values are marked in bold and underline.

Method	IS (\uparrow)	FID (\downarrow)	RP (\uparrow)
GAN-INT-CLS [42]	2.73	194.41	3.83
StackGAN++ [61]	4.10	27.40	13.57
AttnGAN [55]	4.32	24.27	65.30
AttnGAN + CL [56]	4.45	17.96	60.82
DM-GAN [62]	4.68	15.52	<u>76.25</u>
DF-GAN [50]	<u>4.77</u>	16.46	42.95
DM-GAN + CL [56]	<u>4.77</u>	14.57	69.80
AttnGAN++ (ours)	4.78	<u>15.01</u>	77.31

these two feature sets are modeled as two *multivariate Gaussian distributions*. Finally, the Fréchet distance is calculated between two distributions.

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{trace} \left(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{\frac{1}{2}} \right), \quad (2)$$

where $X_r \sim \mathcal{N}(\mu_r, \Sigma_r)$ and $X_g \sim \mathcal{N}(\mu_g, \Sigma_g)$ are the features of real images and generated images extracted by a pretrained Inception-v3 model. Lower FID value means better image quality and diversity.

R-precision (RP) [55] metric is used popularly to evaluate text-image consistency. The idea of RP is to use synthesized image query again the input caption. In particular, given a ground truth text description and 99 mismatching captions sampled randomly, an image is generated from ground truth caption. Then this image is used to query again input description among 100 candidate captions. This retrieval is marked as successful if the matching score of it and ground truth caption is the highest one. The cosine similarity between image encoding vector and caption encoding vector is used as matching score. RP is the ratio of successful retrieval and higher score means better quality.

3.2 Benchmark Results

In this section, we conduct an assessment to re-evaluate existing text-to-image models in the single-object case. For simplicity, CUB dataset [53] is selected for our mini-benchmark and used to generate images with only one object from fine-grained text description. CUB dataset [53] contains 11,788 images from 200 different bird species. We follow the same setup as mentioned in [60] to pre-process and prepare train/test data in zero-shot setting.

We suggest a new baseline approach for this benchmark based on recent breakthroughs in deep learning techniques, in addition to previous efforts. Particularly, we revise the architecture of AttnGAN [55] by adding the spectral

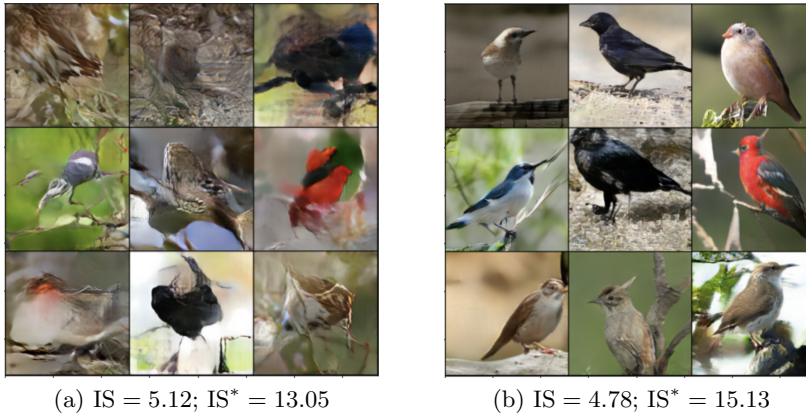


Fig. 2. Evaluating the single-object text-to-image synthesis models can be inconsistent with the IS score. (a) Generated images from the counter model are unrealistic but the IS score of this model is high; (b) Generated images of our AttnGAN++. As can be seen, our IS* fixes well this inconsistency issue.

normalization layers to the discriminator that helps stabilize the training process. We also hand-tune the hyperparameters of our baseline network, which we denote as AttnGAN++. The detail architecture and network setting of AttnGAN++ are shown in supplementary material. The quantitative results of our benchmark is reported in Table 1, which brings the following insights.

Insight 1: AttnGAN++ is a strong baseline. As can be seen in Table 1, our AttnGAN++ outperforms the original version (AttnGAN) with a large gap on all metrics for CUB dataset and has the comparable results with existing state-of-the-art works. It is worth noting that most of current state-of-the-art works [62,56,25,23] are built on AttnGAN. Therefore, this empirical finding would help create a very strong baseline for further improving the successor works. The qualitative results can be found in the supplementary.

Insight 2: IS scores are inconsistent. During the development of AttnGAN++, we discovered that it is feasible to design a generator that produces unrealistic images while yet having a high IS score, which we refer to as the *counter model*. Generated images from this counter model is shown in Figure 2(a). Note that the images from the counter model are randomly sampled and not curated. We describe the architecture of this counter model as well as how to reproduce these results in the supplementary.

Motivated by Insight 2, we revisited the definition of IS metric, and discovered that the inconsistency is due to a pitfall when the IS score is computed in the text-to-image synthesis task. From this observation, we proposed an improved version of IS that address such limitation, as follows.

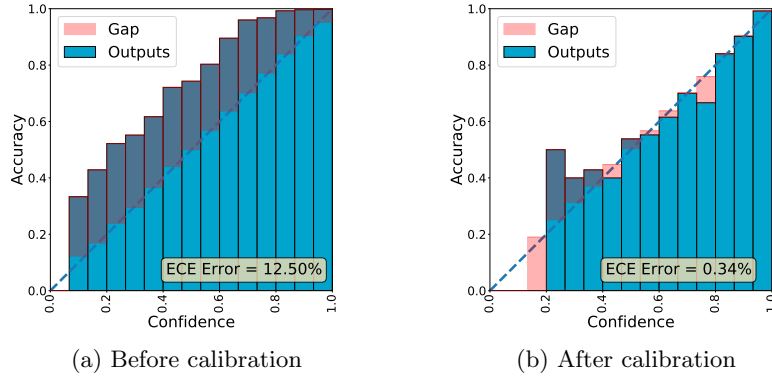


Fig. 3. Reliability diagrams of the fine-tuned Inception-v3 network on the CUB dataset before and after calibration.

3.3 Improved Inception score (IS*): Calibrating Image Classifiers

We found that the pretrained classifier based on the Inception network (used to calculate IS) is uncalibrated or mis-calibrated. As a result, the classifier tends to be either over-confident or under-confident. This is verified by using expected calibration error (ECE) [31] and reliability diagram [5,32]. ECE is the popular metric used to evaluate calibration whereas reliability diagram is a tool to visualize calibration quality. A classifier is well calibrated if they have a small ECE value and reliability diagram is close to identity. As can be seen in Figure 3(a), the Inception network, pretrained by StackGAN [60] for evaluating recent text-to-image models on CUB, is under-confident. When computing the IS, this leads to inconsistency due to erroneous distance between conditional and marginal probability distributions.

To tackle this issue, we propose to calibrate the confidence score of the classifier, which we opt to apply the popular network calibration method of temperature scaling [10]. Particularly, the classifier receives an input image x and output a logit vector z . Before this logit vector z is passed to a softmax layer to obtain probability values, we calibrate z by scaling it with a positive scalar value T for all classes. The conditional probability $p(y = k | x)^*$ with class label $k \in \{1..K\}$ after calibration is:

$$p(y = k | x)^* = \sigma(z/T)_k, \quad (3)$$

where K is the number of classes, T is the temperature, and σ represents the softmax function. We use the $p(y | x)^*$ vector for computing the divergence in IS*. The calibrated confidence score is $\max_k p(y = k | x)^*$. The value of T is obtained by optimizing the negative log-likelihood loss on the validation set used to train the classifier. After calibration on CUB, we get $T = 0.598$. Figure 3(b) showed that after calibration, the under-confident issue is greatly mitigated illustrated by a significant drop in ECE error and a nearly diagonal shape of the

Table 2. A comparison between the IS and IS* scores on the CUB dataset. Thanks to the calibration step, our IS* no longer suffers the problem of counter models being ranked high despite producing bad results.

Method	IS (\uparrow)	IS* (\uparrow)
GAN-INT-CLS [42]	2.73	7.51
StackGAN++ [61]	4.10	12.69
AttnGAN [55]	4.32	13.63
AttnGAN + CL [56]	4.45	14.42
DM-GAN [62]	4.68	15.00
DF-GAN [50]	4.77	14.70
DM-GAN + CL [56]	4.77	<u>15.08</u>
Counter Model	5.12	13.05
AttnGAN++ (ours)	<u>4.78</u>	15.13
<i>Real Images</i>	<i>24.16</i>	<i>46.27</i>

plot. The IS* score shown in Table 2 demonstrated that the inconsistent score causing by the countermodel is also addressed by using IS* instead of IS.

Summary. Single-object text-to-image synthesis is a relatively well-explored topic. Challenges still arise with new tasks, e.g., validating the models with novel word compositions [35]. Here we focused on the evaluation aspect and provided a unified benchmark with existing metrics and our IS* metric. Note that while both IS and FID are for image quality assessment, the benefit of IS (and our IS*) is that it does not require the distribution of real images for evaluation.

4 Multiple-Object Text-to-Image Synthesis

Evaluating text-to-image synthesis models with multiple objects is far more difficult than with a single object. The comprehensive survey by Frolov et al. [7] suggested many essential aspects for evaluating multiple-object text-to-image synthesis. We summarize these aspects in Table 3. As can be seen, simply using existing metrics as in the single-object case is insufficient because many critical aspects in the multi-object case have been implied or ignored, such as object count, relative position among objects, etc. In this section, we will describe a systematic approach for evaluating multi-object text-to-image models by revisiting and improving existing metrics and proposing new metrics for aspects that do not yet have a metric to quantify. Before we get into the specifics of the evaluation metrics, let us give an overview of the benchmark dataset that we use. Our benchmark is conducted on the MS-COCO version 2014 dataset [27], which contains photos with many objects and complex backgrounds. We choose MS-COCO since this dataset is used popularly in developing text-to-image model with multiple objects. The setup for preparing training and validation set in our experiments are same with [42]. In particular, we employ the official training set

Table 3. Demanding aspects for the evaluation of multi-object text-to-image models presented by [7] and our proposed metrics to assess the lacking criteria.

Metric	Image Realism	Object Fidelity	Text Relevance	Object Accuracy	Positional Alignment	Counting Alignment	Paraphrase Robustness	Explainable	Automatic
IS [44]	✓								✓
FID [12]	✓								✓
RP [55]			✓						✓
SOA [13]			✓	✓					✓
O-IS (Ours)		✓							✓
O-FID (Ours)		✓							✓
PA (Ours)					✓				✓
CA (Ours)						✓			✓
Human	✓	✓	✓	✓	✓	✓	✓	✓	

of MS-COCO (approximately 80K images) as the training set of text-to-image models, and we test models on the MS-COCO validation set (approximately 40K images).

4.1 Existing Metrics

Image Realism. FID and our IS* can be used to analyze the photorealism of multi-object synthetic images in the same way they have been used for single object images.

Text Relevance. Current studies use RP to assess the alignment between text and the generated image. However, this metric is shown to overfit in multiple-object synthesis, having inconsistent ranking with real images, which can be seen in Figure 1. One reason for this is that previous works have used the same image and text encoders from DAMSM [55] for training and computing RP. To alleviate this overfitting issue, we use an independent text encoder and image encoder for RP. We selected CLIP [37], a powerful text and image encoders trained on a very large-scale dataset with 400 million text-image pairs. This idea is also used by the concurrent work of Park et al. [35]. In our experiment, the overfitting problem of RP is mitigated using two new encoders, as demonstrated by the value of RP in real images have a large gap with the previous methods. A comparison between the traditional and our modified RP results can be found in supplementary material.

Object Accuracy. Semantic Object Accuracy (SOA) [13] is proposed to measure whether generate images having the objects mentioned in the caption. Specifically, the authors proposed two sub-metrics including SOA-I (average recall between images) and SOA-C (average recall between classes), which are formulated as:

$$\text{SOA-C} = \frac{1}{|C|} \sum_{c \in C} \frac{1}{|I_c|} \sum_{i_c \in I_c} \text{Object-Detector}(i_c), \quad (4)$$

$$\text{SOA-I} = \frac{1}{\sum_{c \in C} |I_c|} \sum_{c \in C} \sum_{i_c \in I_c} \text{Object-Detector}(i_c), \quad (5)$$

where C is the object category set; I_c is a set of images belonging to category c ; $\text{Object-Detector}(i_c) \in \{0, 1\}$ is a pretrained object detector returning 1 if the detector detect successfully an object belong to class c in i_c .

As can be seen, SOA is a plausible metric to evaluate the object accuracy factor in the text-to-image model. However, we found that both CPGAN [25] and SOA used the same pre-trained YOLO-v3 [41] in their implementation, which can potentially lead to overfitting. Empirically, the values of SOA-I and SOA-C of CPGAN are better than those for real images despite images from CPGAN are still non-realistic (Figure 1). To lessen the chance of overfitting, we choose Mask-RCNN [11] instead of YOLO-v3 to compute SOA. The empirical result in our experiment shows that this selection helps mitigate the inconsistency problem. A comparison between the SOA results when using YOLO-v3 and Mask-RCNN can be found in the supplementary material. In this paper, we solely report SOA values computed by Mask-RCNN.

We now turn to describe our new metrics. As shown in Table 3, several aspects in evaluating multi-object text-to-image models remain lacking. Unsolved aspects that we will tackle in this paper include *Object Fidelity*, *Positional Alignment* and *Counting Alignment*. Positional alignment measures the relative position among the objects in the image, e.g., when there is a man and a tree in an image, whether ‘a man stands in front of a tree’ and ‘a man stands behind a tree’ affects the positional alignment. Counting alignment measures the compatibility of the number of objects illustrated by the input sentence and the generated image. Object fidelity evaluates the quality of the object set extracted from generated images. In the survey by Frolov et al. [7], the authors simply provided a discussion without providing any concrete metrics for such aspects. In the following sections, we propose new metrics to address these shortcomings.

4.2 Object Fidelity

Object-centric IS (O-IS) and Object-centric FID (O-FID) are our straightforward extensions of IS and FID with the aim to measure object fidelity in the generated images. In the literature, SceneFID [47] is the closest metric that can assess this criteria and is proposed for evaluating layout-to-image models. However, SceneFID requires the ground truth object bounding boxes from the layout to extract objects in the images preventing them to apply for the text-to-image task. In this work, we replace the need of using ground truth bounding boxes by leveraging the bounding boxes predicted by an off-the-shelf object detection model. Specifically, we first use a well-trained object detector to localize and crop all object regions in each image in the generated image set. By treating all image regions as independently generated, we evaluate the fidelity by IS* and FID on the image regions, respectively. In our experiments, we used Mask-RCNN [11] pre-trained on MS-COCO as the object detector. We also fine-tune and then calibrate the Inception-v3 classifier on the object dataset cropped from the images in MS-COCO based on ground truth bounding boxes to obtain a classifier having 80 classes, equaling the number of classes in MS-COCO. The Inception-v3 network after fine-tuning is used for both computing O-IS and O-FID.

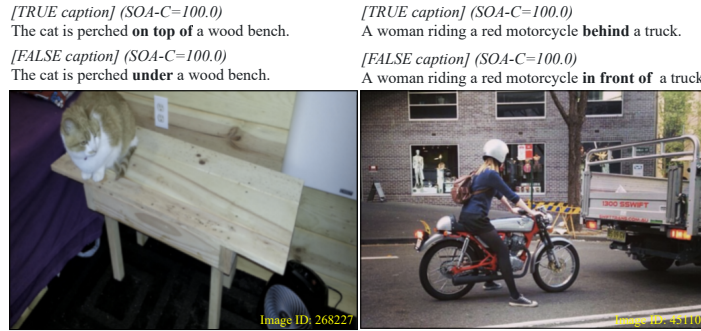


Fig. 4. Assessing positional alignment of the objects in the multi-object image is critical, yet it is still mostly ignored. This example shows a flaw in the existing metrics, such as SOA, which completely ignored the evaluation of positional alignment while maintaining good object accuracy. As can be seen, the SOA values for the image with the *true* caption and with the *false* caption are the same, which demonstrates that the SOA metric skips positional alignment. This weakness leads to the appearance of our Positional Alignment (PA) metric.

4.3 Positional Alignment

Text descriptions are used to describe an image and typically include phrases that convey the positioning information between objects, such as *behind*, *on top of*, *etc* (Figure 4). However, existing object-aware metrics like SOA do not penalize such incorrect relative object locations (e.g. generated images with inaccurate positional alignment still has high SOA scores). To tackle this issue, we propose a new metric to evaluate positional alignment, denoted by PA. First, we define the set of positional words as $W = \{above, right, far, outside, between, below, on top of, bottom, left, inside, in front of, behind, on, near, under\}$. For each word w in W , we filter the captions having word w in the evaluation set of the COCO dataset, and obtain the caption set P_w for each word w . Each caption in P_w is a matched caption, which means the image clearly explains the text. Given P_w , we build a mismatched caption by replacing w in the matched caption by its antonym and keeping other words. For example, the mismatched caption of "A man is *in front of* the blue car" is "A man is *behind* the blue car". Our evaluation begins by generating images from the matched captions in the test dataset. For each word w in W , we now have a set $D_w = \{(R_{wi}, P_{wi}, Q_{wi})\}_{i=1}^{N_w}$ where R_{wi} is a generated image from P_{wi} ; P_{wi} is matched caption; Q_{wi} is mismatched caption of P_{wi} ; N_w is the number of captions having word w . For each triplet in D_w , we use the image R_{wi} to query the input caption from the binary query set including matched caption P_{wi} and mismatched caption Q_{wi} . We mark a query as successful if the matched caption is successfully queried. The query success rate measures the positional alignment quality over all words:

$$PA = \frac{1}{|W|} \sum_{w \in W} \frac{k_w}{N_w}, \quad (6)$$

where k_w is the number of success cases, and $|W|$ is the total number of words. For image-to-text query, we use CLIP [37] as our text-image matching model.

4.4 Counting Alignment

In the multi-object case, counting alignment is an vital factor but so far disregarded in current text-to-image synthesis evaluation. Therefore, we propose a metric for counting alignment (CA metric) that measures how closely the number of objects in a generated image matches the text description.

To evaluate with CA, we first need to construct the test data by filtering from captions in MS-COCO validation set the captions mentioned counting aspect such as *a, one, two, three, four*. From these selected captions, we annotate the ground truth counting information for each one. It is worth noting that we only annotate the object types which can be counted by an object counter to avoid this metric to penalizing those object categories, which cannot be counted. For example, with a caption *"A group of seven people having a light meal and discussion at a single large table"*, the ground truth counting is {"person": 7.0, "dining table": 1.0}. Finally, we created a counting test set D with 1000 records. Each record has a form of (t, c) , in which t is an input text description, and c is the ground truth counting information.

We use a text-to-image model to generate images from each caption and use an off-the-shelf object counting model [3] to count the number of objects for each object class from generated images. To get CA value, we compare the object count to the ground truth and measure the counting error using root mean squared error averaged over the test images:

$$CA = \frac{1}{|D|} \sum_{i=1}^{|D|} \sqrt{\frac{1}{N_{ic}} \sum_{j=1}^{N_{ic}} (c_{ij} - \hat{c}_{ij})^2}, \quad (7)$$

where c_{ij} and \hat{c}_{ij} is the ground truth and predicted object count in the image i for object class j ; N_{ic} is the number of ground truth object types in image i , $|D|$ is the number of test samples.

4.5 Ranking Score

To facilitate the benchmark, we propose a simple formula to compute an average score for ranking purpose. The ranking score is calculated by summing all *rankings* of the considered metrics. To the best of our knowledge, a similar approach is used in the nuScenes challenge for autonomous driving [2] that ranks object detection methods by combining metrics for different bounding box properties such as center, orientation, and dimensions. In our case, since some evaluation aspects could have more than one metric variant, the ranking for each aspect is the average of the ranking of the variants. We treat all metrics and aspects equally, and thus use $\frac{1}{2}$ weight for IS and FID in image realism, O-IS and O-FID

Table 4. Benchmark performances of the multi-object text-to-image synthesis models on the MS-COCO dataset. The **best** and **runner-up** values are marked in bold and underline, respectively. As can be seen, our AttnGAN++ gains the competitive results compared to the current state-of-the-art text-to-image synthesis methods.

Method	IS* (↑)	FID (↓)	RP(↑)	SOA-C(↑)	SOA-I (↑)	O-IS (↑)	O-FID (↓)	CA (↓)	PA (↑)	RS (↑)
GAN-CLS [42]	8.10	192.09	10.00	5.31	5.71	2.46	51.13	2.51	32.79	7.0
StackGAN [60]	15.50	53.44	9.10	9.24	9.90	3.36	29.09	2.41	34.33	11.5
AttnGAN [55]	33.79	36.90	50.56	47.13	49.78	5.04	20.92	1.82	40.08	29.0
DM-GAN [62]	45.63	28.96	66.98	55.77	58.11	5.22	17.48	1.71	42.83	41.0
CPGAN [25]	59.64	50.68	69.08	81.86	83.83	6.38	20.07	2.07	43.28	43.0
DF-GAN [50]	30.45	21.05	42.44	37.85	40.19	5.12	14.39	1.96	40.39	31.5
AttnGAN + CL [56]	36.85	26.93	57.52	47.45	49.33	4.92	19.92	1.72	43.92	37.0
DM-GAN + CL [56]	46.61	<u>22.60</u>	<u>70.36</u>	58.68	61.05	5.09	15.50	<u>1.66</u>	49.06	<u>51.5</u>
DALLE-mini (zero-shot) [4]	19.82	62.90	48.72	26.64	27.90	4.10	23.83	2.31	47.39	23.5
AttnGAN++ (Ours)	<u>54.63</u>	26.58	72.48	<u>67.83</u>	<u>69.97</u>	<u>6.01</u>	<u>15.43</u>	1.57	<u>47.75</u>	56.0
<i>Real Images</i>	<i>51.25</i>	<i>2.62</i>	<i>83.54</i>	<i>90.02</i>	<i>91.19</i>	<i>8.63</i>	<i>0.00</i>	<i>1.05</i>	<i>100.0</i>	<i>65.0</i>

in object fidelity, SOA-I and SOA-C in object accuracy; other metrics have a unit weight. Our ranking score (RS) is computed as

$$\begin{aligned}
 \text{RS} = & \frac{1}{2}(\#\text{IS}^* + \#\text{FID}) + \frac{1}{2}(\#\text{O-IS} + \#\text{O-FID}) \\
 & + \frac{1}{2}(\#\text{SOA-I} + \#\text{SOA-C}) + \#\text{PA} + \#\text{CA} + \#\text{RP},
 \end{aligned} \tag{8}$$

where $\#(\text{metric}) \in \{1..N\}$ denotes the ranking by a particular metric with N is the number of considered methods.

4.6 Benchmark Results

We show the benchmark results in Table 4, from which we draw some following insights. Firstly, our proposed metrics (O-IS, O-FID, CA, PA) and two improved version of existing metrics (RP, SOA), properly rank real images as the best. An exception is IS* which ranks AttnGAN++ and CPGAN better than real images. However we opt to retain this metric due to its excellent properties on the single-object case, and the ranking score is consistent to human when including IS*. Second, our AttnGAN++ is ranked top for multi-object text-to-image synthesis in terms of overall performance, demonstrating that it is a substantial strong baseline for both single-object and multiple-object instances. Third, breaking down each part of our evaluation pipeline allows us to more clearly analyze each model’s flaws and strengths than earlier evaluations. For examples, CPGAN outperforms other techniques on SOA-I and SOA-C since it explicitly considers object-level information in the training phase. DM-GAN + CL is the most effective method for positional alignment. While our AttnGAN++ performs better in the remaining aspects. The details of aspect’s scores for each method are included in the supplementary material.

Table 5. Human evaluation results on the MS-COCO dataset. In this table, ranking scores (RS) are recalculated using just 5 considered techniques and real photos. As can be observed, RS is well-aligned with human decisions.

Method	Ranking Score (\uparrow)	Human Score (\uparrow)
StackGAN [60]	6.00	28.45
AttnGAN [55]	13.5	37.40
DM-GAN [62]	20.0	41.47
CPGAN [25]	23.0	43.73
AttnGAN++ (ours)	28.5	45.01
<i>Real Images</i>	<i>35.0</i>	<i>99.82</i>

4.7 Human Evaluation

To ensure that our evaluations are reliable, we conducted a user analysis to test the metrics against assessments done by humans. We opt for 5 methods including StackGAN, AttnGAN, DM-GAN, CPGAN, AttnGAN++ (ours), and real images to conduct our user survey. We sample 50 test captions from MS-COCO and use the above methods to generate an image for each caption. The IDs for these captions are provided in supplementary for reproducibility. We ask each human subject (40 participants in total) to score each method from 1 (worst) to 5 (best) based on two criteria: *plausibility* – whether the image is plausible based on the content of the caption (object accuracy, counting, and positional alignment, text relevance), and *naturalness* – whether the image looks natural. The score of each human subject for each method is the sum of score of 50 images and divide by 250 for normalization. The final score of each method is an average of the scores of each participant. Our evaluation result in Table 5 shows that our final ranking is well-aligned with human evaluation.

5 Conclusion

This paper performed an empirical study with benchmarks for text-to-image synthesis methods for both single-object and multiple-object scenario. The benchmark results reveal the inconsistency issues in the existing metrics, prompting us to propose the improved version of existing metrics as well as new metrics to evaluate many vital but lacking aspects in the multiple-object case. Our extensive experiments show that this bag of metrics provides a better and more consistent ranking with real images and human evaluation.

Our bag of metrics for text-to-image synthesis is by no means perfect. The proposed metrics can be further extended for complex cases, for example, to handle more positional words for positional alignment score and indefinite numeral adjectives (e.g., several, many) for counting alignment.

References

1. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096 (2018) **19**
2. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR (2020) **12**
3. Cholakkal, H., Sun, G., Khan, F.S., Shao, L.: Object counting and instance segmentation with image-level supervision. In: CVPR (2019) **12**
4. Dayma, B., Patil, S., Cuenca, P., Saifullah, K., Abraham, T., Le Khac, P., Melas, L., Ghosh, R.: Dall-e mini (2021), <https://github.com/borisdama/dalle-mini> **13, 24**
5. DeGroot, M.H., Fienberg, S.E.: The comparison and evaluation of forecasters. Journal of the Royal Statistical Society: Series D (The Statistician) **32**(1-2) (1983) **7**
6. Ding, M., Yang, Z., Hong, W., Zheng, W., Zhou, C., Yin, D., Lin, J., Zou, X., Shao, Z., Yang, H., et al.: Cogview: Mastering text-to-image generation via transformers. NeurIPS (2021) **4**
7. Frolov, S., Hinz, T., Raue, F., Hees, J., Dengel, A.: Adversarial text-to-image synthesis: A review. Neural Networks (2021) **3, 8, 9, 10, 21**
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014) **2**
9. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of wasserstein gans. arXiv preprint arXiv:1704.00028 (2017) **19**
10. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. arXiv preprint arXiv:1706.04599 (2017) **7**
11. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017) **10**
12. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS (2017) **2, 4, 9**
13. Hinz, T., Heinrich, S., Wermter, S.: Semantic object accuracy for generative text-to-image synthesis. arXiv preprint arXiv:1910.13321 (2019) **3, 9**
14. Hong, S., Yang, D., Choi, J., Lee, H.: Inferring semantic layout for hierarchical text-to-image synthesis. In: CVPR (2018) **4**
15. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015) **26**
16. Kang, M., Park, J.: Contragan: Contrastive learning for conditional image generation (2020) **19**
17. Karnewar, A., Wang, O.: Msg-gan: Multi-scale gradients for generative adversarial networks. In: CVPR (2020) **18**
18. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017) **30**
19. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: CVPR (2019) **2, 30**
20. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: CVPR (2020) **2**
21. Kodali, N., Abernethy, J., Hays, J., Kira, Z.: On convergence and stability of gans. arXiv preprint arXiv:1705.07215 (2017) **19**
22. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N (2015) **18**

23. Li, B., Qi, X., Lukasiewicz, T., H. S. Torr, P.: Controllable text-to-image generation. arXiv preprint arXiv:1909.07083 (2019) [6](#)
24. Li, W., Zhang, P., Zhang, L., Huang, Q., He, X., Lyu, S., Gao, J.: Object-driven text-to-image synthesis via adversarial training. In: CVPR (2019) [2](#), [4](#)
25. Liang, J., Pei, W., Lu, F.: Cpgan: Full-spectrum content-parsing generative adversarial networks for text-to-image synthesis. arXiv preprint arXiv:1912.08562 (2019) [2](#), [4](#), [6](#), [10](#), [13](#), [14](#), [21](#), [22](#), [24](#)
26. Lim, J.H., Ye, J.C.: Geometric gan. arXiv preprint arXiv:1705.02894 (2017) [19](#)
27. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014) [2](#), [8](#), [21](#), [22](#), [24](#), [28](#), [31](#)
28. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research **9**(Nov) (2008) [25](#), [32](#), [33](#), [34](#)
29. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014) [3](#)
30. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018) [19](#), [24](#), [26](#)
31. Naeini, M.P., Cooper, G.F., Hauskrecht, M.: Obtaining well calibrated probabilities using bayesian binning. In: AAAI (2015) [7](#)
32. Niculescu-Mizil, A., Caruana, R.: Predicting good probabilities with supervised learning. In: ICML (2005) [7](#)
33. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing (Dec 2008) [2](#)
34. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans. ICML (2017) [19](#)
35. Park, D.H., Azadi, S., Liu, X., Darrell, T., Rohrbach, A.: Benchmark for compositional text-to-image synthesis. In: NeurIPS Datasets and Benchmarks Track (2021) [8](#), [9](#)
36. Qiao, T., Zhang, J., Xu, D., Tao, D.: Mirrorgan: Learning text-to-image generation by redescription. In: CVPR (2019) [3](#)
37. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. arXiv preprint arXiv:2103.00020 (2021) [9](#), [12](#), [21](#)
38. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015) [19](#)
39. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: ICML (2021) [4](#)
40. Razavi, A., van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with vq-vae-2. In: NeurIPS (2019) [4](#)
41. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018) [10](#)
42. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text-to-image synthesis. In: ICML (2016) [3](#), [5](#), [8](#), [13](#), [24](#)
43. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV **115**(3) (2015) [18](#)

44. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: NeurIPS (2016) [2](#), [4](#), [9](#)
45. Shen, Y., Gu, J., Tang, X., Zhou, B.: Interpreting the latent space of gans for semantic face editing. In: CVPR (2020) [2](#)
46. Shen, Y., Yang, C., Tang, X., Zhou, B.: Interfacegan: Interpreting the disentangled face representation learned by gans. arXiv preprint arXiv:2005.09635 (2020) [2](#)
47. Sylvain, T., Zhang, P., Bengio, Y., Hjelm, R.D., Sharma, S.: Object-centric image generation from layouts. In: AAAI (2021) [10](#)
48. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR (2016) [4](#)
49. Tan, H., Liu, X., Li, X., Zhang, Y., Yin, B.: Semantics-enhanced adversarial nets for text-to-image synthesis. In: ICCV (2019) [2](#), [3](#)
50. Tao, M., Tang, H., Wu, F., Jing, X.Y., Bao, B.K., Xu, C.: Df-gan: A simple and effective baseline for text-to-image synthesis. In: CVPR (2022) [5](#), [8](#), [13](#), [24](#)
51. Theis, L., Oord, A.v.d., Bethge, M.: A note on the evaluation of generative models. arXiv preprint arXiv:1511.01844 (2015) [4](#)
52. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017) [4](#)
53. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-ucsd birds 200. Tech. Rep. CNS-TR-2010-001, California Institute of Technology (2010) [2](#), [5](#), [24](#), [27](#), [31](#), [32](#), [33](#), [34](#)
54. Wu, Y., Donahue, J., Balduzzi, D., Simonyan, K., Lillicrap, T.: Logan: Latent optimisation for generative adversarial networks. arXiv preprint arXiv:1912.00953 (2019) [19](#)
55. Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., He, X.: Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In: CVPR (2018) [2](#), [3](#), [4](#), [5](#), [8](#), [9](#), [13](#), [14](#), [21](#), [22](#), [24](#), [25](#), [26](#), [29](#), [30](#)
56. Ye, H., Yang, X., Takac, M., Sunderraman, R., Ji, S.: Improving text-to-image synthesis using contrastive learning. arXiv preprint arXiv:2107.02423 (2021) [4](#), [5](#), [6](#), [8](#), [13](#), [24](#)
57. Yin, G., Liu, B., Sheng, L., Yu, N., Wang, X., Shao, J.: Semantics disentangling for text-to-image generation. In: CVPR (2019) [2](#), [3](#)
58. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks. In: ICML (2019) [19](#)
59. Zhang, H., Koh, J.Y., Baldridge, J., Lee, H., Yang, Y.: Cross-modal contrastive learning for text-to-image generation (2021) [4](#)
60. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: ICCV (2017) [3](#), [5](#), [7](#), [13](#), [14](#), [21](#), [22](#), [24](#)
61. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: Stackgan++: Realistic image synthesis with stacked generative adversarial networks. TPAMI **41**(8) (2018) [3](#), [5](#), [8](#)
62. Zhu, M., Pan, P., Chen, W., Yang, Y.: Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In: CVPR (2019) [2](#), [3](#), [5](#), [6](#), [8](#), [13](#), [14](#), [21](#), [22](#), [24](#)

Supplementary Material

This supplemental document provides more details of our bag of metrics. We first present the details of our improved IS* metric including the impact of the calibration step on the IS* score (Section A.1) and the implementation details for the counter model for reproducing the inconsistency problem of IS (Section A.2). We then detailed our improved versions of RP and SOA and show how our metrics can mitigate the overfitting issues in the original versions in the multi-object text-to-image synthesis (Section B). Next, we detail our benchmark including the statistics of our test data (Section C.1), a complete benchmark of multi-object text-to-image models based on each assessment criterion (Section C.2), the architecture and network configurations of our AttnGAN++ baseline (Section D) as well as more visual examples (Section E) and t-SNE visualization (Section F). Finally, we provide the caption ids we used in our user study (Section G).

A Details for our IS* metric

A.1 Impact of calibration on the classifier and IS*

In the main paper, we showed that severe miscalibration of the classifier used to compute IS on the CUB dataset in previous methods led to inconsistent IS scores of a counter model, and we proposed IS* to fix this issue. In this section, we further conduct another experiment to verify the impact of the calibration step causing on computing IS*. In detail, this experiment is performed on the vanilla GAN task, which is the Tiny ImageNet [22] image generation. The classifier used to measure IS in these works is the Inception-v3 network pre-trained on ImageNet [43]. It is worth noting that this classifier is used popularly for measuring IS in the traditional GAN image generation task. The IS and IS* results are shown in Table A.1. We also plot the reliability diagrams and ECE errors of this classifier before and after calibration in Figure 5. As can be seen, even before calibration, this classifier is noticeably well calibrated. Hence, the effect of the calibration process on this classifier is negligible demonstrated through the temperature T after calibration is 0.909 ($T = 1$ means calibration does not have any effects on classifier). Therefore, we would only see the local ranking differs in IS and IS*.

A.2 Counter model implementation

This section details the development of our counter model, which was utilized to demonstrate the inconsistency problem of IS. Our counter model is built on AttnGAN++ and MSG-GAN [17]. Table 13 shows the network details of the counter model. The training and evaluation configurations can be found in Table 14. More random (not curated) visual samples synthesized by the counter model are also provided in Figure 6 to demonstrate that these samples from the counter model are quite poor in comparison to those from AttnGAN++.

Table 6. Comparing the ranking of IS and IS* on Tiny ImageNet dataset with various GAN models. The cases, which are ranked inconsistently between IS and IS*, are marked in **bold**. As we expect, only local ranking differs between IS and IS* appear due to the well-calibrated of the classifier even before calibration.

Method	IS (\uparrow)	IS* (\uparrow)
WGAN-GP [9]	1.64	1.79
GGAN [26]	5.22	7.00
DCGAN [38]	5.70	7.79
ACGAN [34]	6.51	9.30
BigGAN-LO [54]	7.83	11.29
SNGAN [30]	8.38	12.36
SAGAN [58]	8.48	12.34
WGAN-DRA [21]	9.35	14.00
BigGAN [1]	12.43	18.80
ContraGAN [16]	13.76	21.64

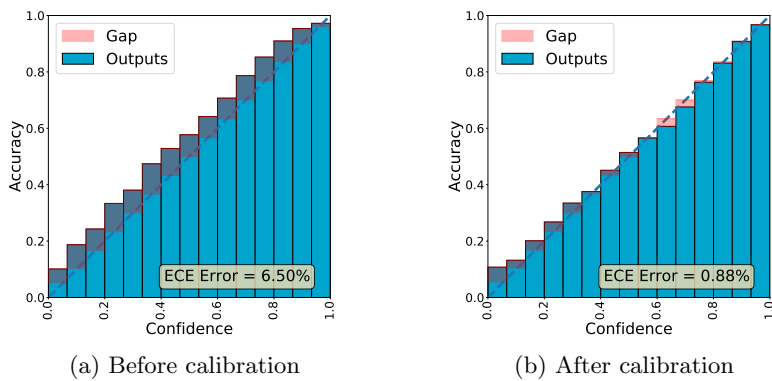


Fig. 5. Reliability diagrams of the Inception-v3 network pre-trained on the ImageNet dataset before and after calibration.

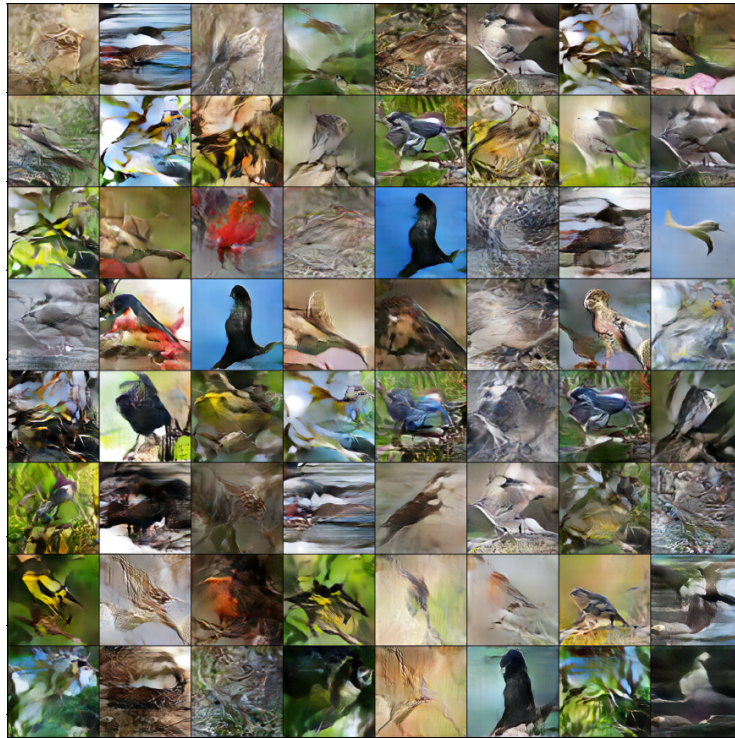


Fig. 6. More random (not curated) visual samples from our countermodel on the CUB dataset. As can be seen, the synthesized images are not realistic in most cases.

Table 7. Comparison of the original version of RP and our improved version one on the MS-COCO [27] dataset. Inconsistent results are marked in **bold**. As can be observed, the value of RP on real photographs is the lowest, showing the original RP’s heavily overfitting issue. Noticeably, our improved RP alleviated significantly by using CLIP [37]. Note that the values of RP in these experiments are calculated from 30,000 captions as most of the previous works do. In the main paper, we only sample 5,000 captions and calculate RP from these captions to save time but guarantee consistent scores.

Method	R-precision (original) (\uparrow)	R-precision (ours) (\uparrow)
StackGAN [60]	72.03	38.46
AttnGAN [55]	83.76	50.92
DM-GAN [62]	92.23	65.91
CPGAN [25]	93.59	70.36
AttnGAN++ (ours)	96.39	73.37
Real Images	67.35	83.65

B Details of our improved RP and SOA

In the main paper, we presented that the existing versions of RP and SOA overfit in the multi-object scenario of the MS-COCO dataset, as evidenced by the fact that the values from some methods on these metrics exceed the corresponding values from real photos although these methods produce images with poorer quality than real photos. We demonstrate this by comparing the values of the original and our modified versions of these metrics in Table 7 and Table 8. As can be seen, the overfitting phenomena on RP and SOA is fully eliminated in our enhanced versions.

C Details of our benchmark

C.1 Benchmark data

In the previous works, the inconsistency in the construction of testing data has caused many difficulties in benchmark models. A comprehensive survey [7] also pointed that there are some metrics are reported with inconsistent numbers between different research works. We find out that the non-unified input test data is one of the reasons leading to this issue. Therefore, we provide unified testing data in our TISE toolbox in order to compare techniques fairly.

The details of our test data is as follows. The number of captions used in each metrics are shown in Table 9 and Table 10 for CUB and MS-COCO, respectively. The distribution of per-class object count and positional words for counting alignment (CA) and positional alignment (PA) metric are visualized in Figure 7 and Figure 8, respectively.

Table 8. Comparison of the original version of SOA (including SOA-I and SOA-C) and our improved version of SOA on the MS-COCO [27] dataset. Inconsistent results are highlighted in **bold**, which shows that SOA-C and SOA-I of CPGAN are higher than real images and our SOA greatly migrated this phenomenon. Note that the values of SOA in this experiment are calculated on full captions provided by the authors of this metric, while the ones we report by our TISE toolbox are computed on the sample from them (about 16k captions) to save time but output the consistent scores.

Method	SOA-C (\uparrow) (original)	SOA-I (\uparrow) (original)	SOA-C (\uparrow) (ours)	SOA-I (\uparrow) (ours)
StackGAN [60]	21.09	30.35	31.34	49.97
AttnGAN [55]	25.88	39.01	47.26	62.02
DM-GAN [62]	33.44	48.03	55.40	68.76
CPGAN [25]	77.02	84.55	82.25	88.97
AttnGAN++(ours)	48.33	67.19	67.52	76.33
Real Images	74.97	80.84	89.98	92.92

Table 9. The number of test captions used in evaluation on the CUB dataset.

Metric	#Captions
Image Realism (IS, FID)	30,000
Text Relevance (RP)	30,000

Table 10. The number of test captions used in evaluating each evaluation aspect on the MSCOCO dataset.

Metric	#Captions
Image Realism (IS, FID)	10,000
Object Fidelity (O-IS, O-FID)	10,000
Text Relevance (RP)	5,000
Object Accuracy (SOA-C, SOA-I)	15,223
Positional Alignment (PA)	1,046
Counting Alignment (CA)	1,000

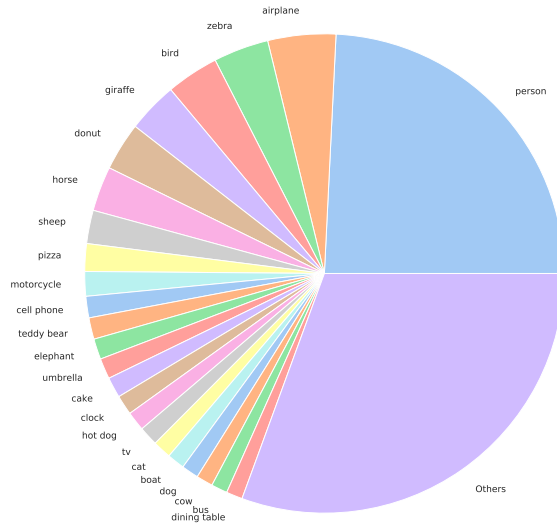


Fig. 7. Distribution of the number of object classes in our provided testing data for counting alignment factor in multi-object case. Best viewed in zoom.

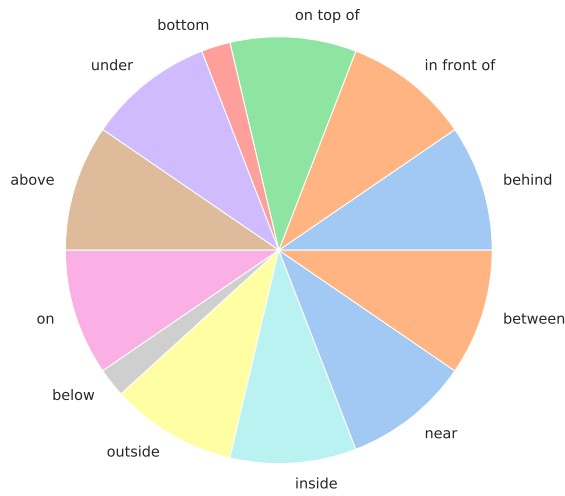


Fig. 8. Distribution of the number of test captions having corresponding positional words considered in our PA metric.

Table 11. The details of the ranking scores for each evaluation aspect of each method on the MS-COCO [27] dataset. **Best** and runner-up values are marked in bold and underline, respectively.

Method	Image Realism	Text Relevance	Object Accuracy	Object Fidelity	Counting Alignment	Positional Alignment
GAN-CLS [42]	1.0	2.0	1.0	1.0	1.0	1.0
StackGAN [60]	2.5	1.0	2.0	2.0	2.0	2.0
AttnGAN [55]	5.0	5.0	5.5	4.5	6.0	3.0
DM-GAN [62]	6.5	7.0	7.0	7.5	8.0	5.0
CPGAN [25]	7.5	8.0	10.0	7.5	4.0	6.0
DF-GAN [50]	7.0	3.0	4.0	<u>8.5</u>	5.0	4.0
AttnGAN + CL [56]	6.5	6.0	5.5	5.0	7.0	7.0
DM-GAN + CL [56]	<u>8.5</u>	<u>9.0</u>	8.0	7.0	<u>9.0</u>	10.0
DALLE-mini (zero-shot) [4]	2.5	4.0	3.0	3.0	3.0	8.0
AttnGAN++ (Ours)	9.0	10.0	<u>9.0</u>	9.0	10.0	<u>9.0</u>
<i>Real Images</i>	<i>10.0</i>	<i>11.0</i>	<i>11.0</i>	<i>11.0</i>	<i>11.0</i>	<i>11.0</i>

C.2 Benchmark on MS-COCO for each evaluation aspect

Table 11 provides the details of aspect’s ranking scores for each method on MS-COCO dataset. Here we show the performance of each model on six evaluation criteria, including Image Realism, Object Accuracy, Text Relevance, Object Accuracy, Object Fidelity, Counting Alignment, Positional Alignment.

D AttnGAN++ Architecture

Along with the assessment toolkit, we also offered our AttnGAN++, a new baseline based on AttnGAN [55]. The main difference between AttnGAN++ and AttnGAN is that we apply spectral normalization [30] to discriminators to stabilize the training process of GAN. With this simple technique, the performance of the model is boosted significantly comparing with the original version. The architecture of AttnGAN++ is shown in Figure 9. The network details and training settings of AttnGAN++ are demonstrated in Table 12 and Table 14 respectively.

E More visual results

Additionally, we show more visual examples of our AttnGAN++ comparing with the current state-of-the-art text-to-image models on CUB [53] dataset in Figure 10 and MS-COCO [27] dataset in Figure 11 for qualitative measuring.

F t-SNE Visualizations

To visualize the statistics of synthesized images, we utilize t-SNE [28]. Firstly, we extract feature vectors from these synthesized images using a pre-trained image encoder [55]. Then, we use t-SNE to convert these high dimensional feature vectors to 2-dimensional positions at which we display the images. The t-SNE visualization of generated images by AttnGAN++ and counter model on CUB can be found in Figure 12 and Figure 13, respectively. Additionally, we also show the t-SNE of all real images of the CUB test set in Figure 14 for reference. Note that the t-SNE image has a very high resolution so it is best viewed with an offline pdf viewer.

G User Study

To facilitate reproducibility, we provide the IDs of captions which we used in our human evaluation: *503647, 302716, 817708, 72017, 563987, 434439, 375212, 478341, 737362, 323692, 177535, 338067, 810717, 416305, 680452, 439866, 558122, 545601, 196294, 380857, 782291, 324845, 767124, 63597, 648878, 73383, 327849, 799148, 829090, 107333, 805428, 371195, 443142, 394904, 754057, 421896, 361352, 517666, 75305, 625131, 202787, 723526, 569736, 442834, 183253, 642468, 277787, 150568, 502193, 643215.*

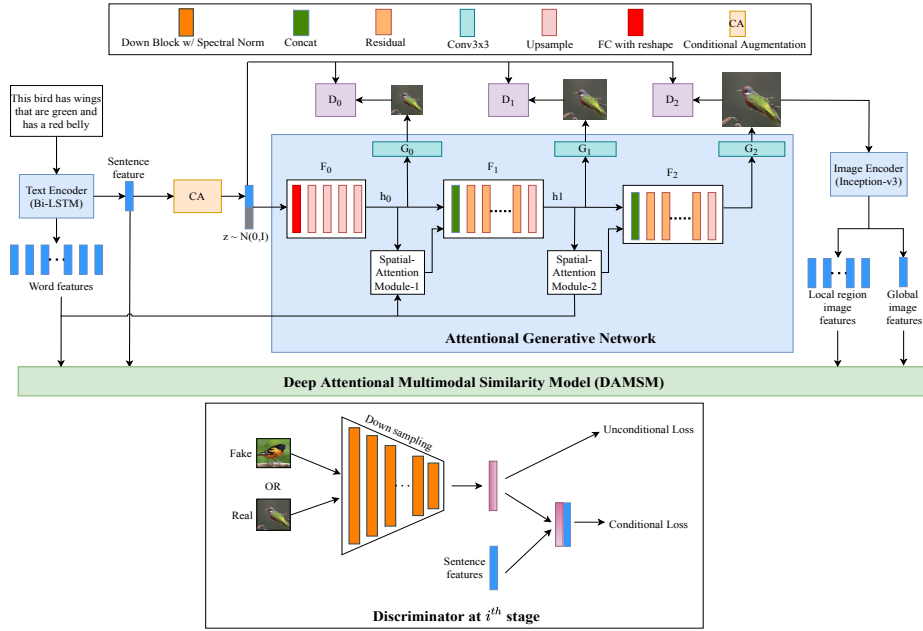


Fig. 9. The architecture of our AttnGAN++. In each discriminator, we employ spectral normalization [30] to each convolution layer instead of using batch normalization [15] as in the original AttnGAN [55]. Implementation details for each layer can be found in Table 12.



Fig. 10. Qualitative examples of the single object text-to-image generation models on the CUB [53] dataset. Best viewed in zoom.

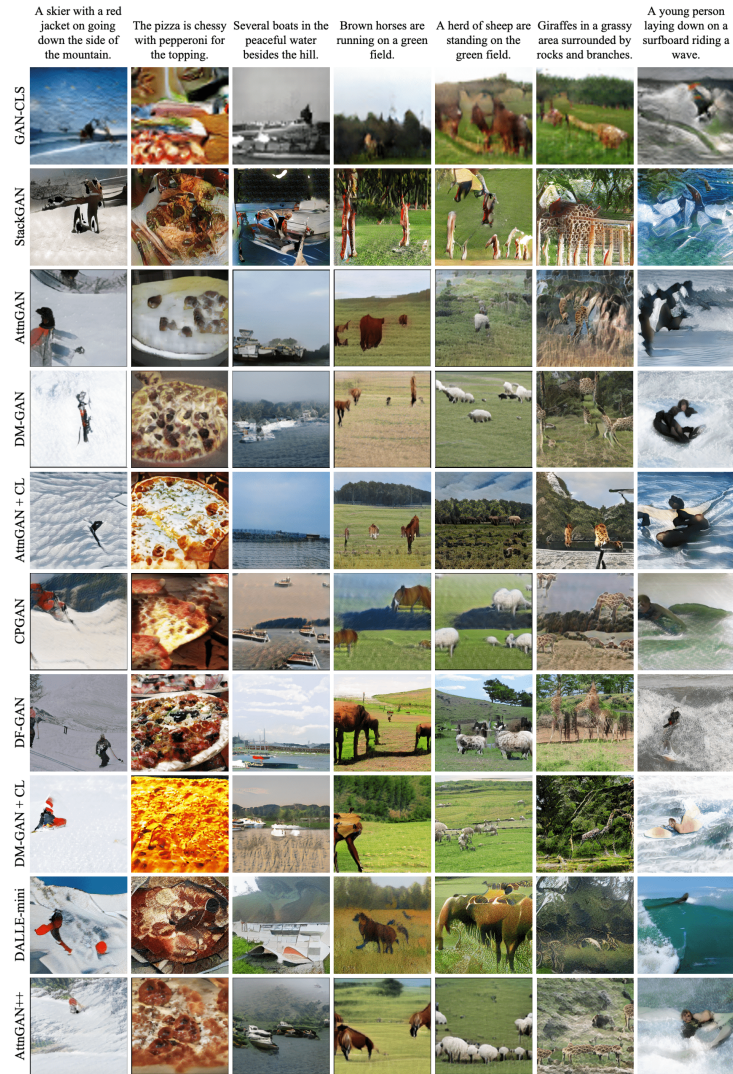


Fig. 11. Qualitative examples of the multi-object text-to-image synthesis models on the MS-COCO [27] dataset. Best viewed in zoom.

Table 12. Network details of our AttnGAN++. Some components which are not mentioned here such as text encoder, image encoder, DAMSM, its settings can be found in AttnGAN [55]. In the tables, k = kernel size, s = stride, p = padding, b = bias.

(a) Generator		(b) Discriminator	
Module	Output shape / Details	Module	Output shape / Details
Up Block		Down Block	
<i>Params:</i> (in_planes, out_planes)		<i>Params:</i> (in_planes, out_planes)	
Input shape	in_planes × h × w	Input shape	in_planes × h × w
Upsampling	Nearest Neighbor, scale factor = 2	SpectralNorm(Conv(k=4, s=2, p=1, b=True))	out_planes × h/2 × w/2
Conv(k=3, s=1, p=1, b=False)	2 × out_planes × h × 2 × w × 2	LeakyReLU(alpha=0.2)	No change shape
BatchNorm2D	No change shape		
Gated Linear Unit (GLU)	out_planes × h × 2 × w × 2		
Residual Block		Block3x3_LeakyReLU	
Input X		<i>Params:</i> (in_planes, out_planes)	
Conv(k=3, s=1, p=1, b=False)	Up channel size by 2,	Input shape	in_planes × h × w
BatchNorm2D	No change shape	SpectralNorm(Conv(k=3, s=1, p=1, b=True))	out_planes × h × w
Gated Linear Unit (GLU)	Down channel size by 2	LeakyReLU(alpha=0.2)	No change shape
Conv(k=3, s=1, p=1, b=False)	No change shape		
BatchNorm2D	No change shape	Discriminator 256 × 256	
Add output w/ X (skip connection)	No change shape	Input tensor	3 × 256 × 256
Spatial Attention layer	see AttnGAN	Down Block	ndf × 128 × 128
Conditional Augmentation (CA)	see AttnGAN	Down Block	2 * ndf × 64 × 64
Generator 64 × 64		Down Block	4 * ndf × 32 × 32
<i>Input</i>		Down Block	8 * ndf × 16 × 16
Input noise	nzf	Down Block	16 * ndf × 8 × 8
Caption Embedding	nef	Down Block	32 * ndf × 4 × 4
<i>Computation</i>		Block3x3_LeakyReLU	16 * ndf × 4 × 4
CA on caption embedding to get c	nef	Block3x3_LeakyReLU	8 * ndf × 4 × 4
Concat c w/ noise	ncf + nzf	<i>Unconditional logits</i>	
Linear(b=False)	ngf * 16 * 4 * 4 * 2	Conv(k=4, s=4, p=0, b=True)	1
BatchNorm1D	No change shape	Sigmoid	1
Gated Linear Unit (GLU)	ngf * 16 * 4 * 4 * 1	<i>Conditional logits</i>	
Reshape	16 * ngf × 4 × 4	Caption Embedding	nef
Up Block 1	8 * ngf × 8 × 8	Concat w/ replicated caption embedding	8 * ndf + nef × 4 × 4
Up Block 2	4 * ngf × 16 × 16	Block3x3_LeakyReLU	8 * ndf × 4 × 4
Up Block 3	2 * ngf × 32 × 32	Conv(k=4, s=4, p=0, b=True)	1
Up Block 4	ngf × 64 × 64	Sigmoid	1
Conv(k=3, s=1, p=1, b=False)	3 × 64 × 64	Discriminator 128 × 128	
Tanh	No change shape	Input tensor	3 × 128 × 128
Generator 128 × 128		Down Block	ndf × 64 × 64
<i>Input</i>		Down Block	2 * ndf × 32 × 32
Previous hidden features	ngf × 64 × 64	Down Block	4 * ndf × 16 × 16
Word Mask	word_num	Down Block	8 * ndf × 8 × 8
Word features	nef × word_num	Down Block	16 * ndf × 4 × 4
<i>Computation</i>		Down Block	8 * ndf × 4 × 4
Spatial Attention Layer	ngf × 64 × 64	Block3x3_LeakyReLU	
Residual Block × residual_num	ngf × 64 × 64	<i>Unconditional logits</i>	
Concat w/ previous hidden features	2 * ngf × 64 × 64	Conv(k=4, s=4, p=0, b=True)	1
Up Block	ngf × 128 × 128	Sigmoid	1
Conv(k=3, s=1, p=1, b=False)	3 × 128 × 128	<i>Conditional logits</i>	
Tanh	No change shape	Caption Embedding	nef
Generator 256 × 256		Concat w/ replicated caption embedding	8 * ndf + nef × 4 × 4
<i>Input</i>		Block3x3_LeakyReLU	8 * ndf × 4 × 4
Previous hidden features	ngf × 128 × 128	Conv(k=4, s=4, p=0, b=True)	1
Word Mask	word_num	Sigmoid	1
Word features	nef × word_num	Discriminator 64 × 64	
<i>Computation</i>		Input tensor	3 × 64 × 64
Spatial Attention Layer	ngf × 128 × 128	Down Block	ndf × 32 × 32
Residual Block × residual_num	ngf × 128 × 128	Down Block	2 * ndf × 16 × 16
Concat w/ previous hidden features	2 * ngf × 128 × 128	Down Block	4 * ndf × 8 × 8
Up Block	ngf × 256 × 256	Down Block	8 * ndf × 4 × 4
Conv(k=3, s=1, p=1, b=False)	3 × 256 × 256	<i>Unconditional logits</i>	
Tanh	No change shape	Conv(k=4, s=4, p=0, b=True)	1
		Sigmoid	1
		<i>Conditional logits</i>	
		Caption Embedding	nef
		Concat w/ replicated caption embedding	8 * ndf + nef × 4 × 4
		Block3x3_LeakyReLU	8 * ndf × 4 × 4
		Conv(k=4, s=4, p=0, b=True)	1
		Sigmoid	1

Table 13. Network details of our countermodel. Some components which are not mentioned here such as text encoder, image encoder, DAMSM, its settings can be found in AttnGAN [55]. In the tables, k = kernel size, s = stride, p = padding, b = bias.

(a) Generator		(b) Discriminator	
Module	Output shape / Details	Module	Output shape / Details
Up Block	see Table 12	Block3x3.LeakyReLU	see Table 12
Residual Block	see Table 12	DisGeneralConvBlock	
Spatial Attention Layer	see AttnGAN	<i>Params: in_planes, concat_planes, out_planes</i>	$in_planes + concat_planes \times h \times w$
Conditional Augmentation	see AttnGAN	MinibatchStdDev (see [18,19])	$in_planes \times h \times w$
Generator 4 × 4		Block3x3.LeakyRelu	$out_planes \times h \times w$
<i>Input</i>		AvgPool2d(k=2)	$out_planes \times h/2 \times w/2$
Input noise	nzf	Discriminator	
Caption Embedding	nef	<i>Input</i>	
<i>Computation</i>		Caption Embedding	nef
Conditional Augmentation on caption embedding	nef	Image scale 4 × 4	3 × 4 × 4
Concat w/ noise	$nef + nef$	Image scale 8 × 8	3 × 8 × 8
Linear(b=False)	$ngf * 16 * 4 * 4 * 2$	Image scale 16 × 16	3 × 16 × 16
BatchNorm1D	No change shape	Image scale 32 × 32	3 × 32 × 32
Gated Linear Unit (GLU)	$ngf * 16 * 4 * 4 * 1$	Image scale 64 × 64	3 × 64 × 64
Reshape	$16 * ngf \times 4 \times 4$	Image scale 128 × 128	3 × 128 × 128
Conv(k=3, s=1, p=1, b=False)	3 × 4 × 4	Image scale 256 × 256	3 × 256 × 256
Tanh	No change shape	<i>Computation</i>	
Generator 8 × 8		Image scale 256 × 256	3 × 256 × 256
Up Block	$8 * ngf \times 8 \times 8$	Conv(k=1, s=1, p=0, b=True)	$ndf \times 256 \times 256$
Conv(k=3, s=1, p=1, b=False)	3 × 8 × 8	DisGeneralConvBlock(ndf , 1, 2 * ndf)	$ndf * 2 \times 128 \times 128$
Tanh	No change shape	Concat w/ Image scale 128 × 128	$ndf * 2 + 3 \times 128 \times 128$
Generator 16 × 16		DisGeneralConvBlock(2 * ndf , 4, 4 * ndf)	$4 * ndf \times 64 \times 64$
Up Block	$4 * ngf \times 16 \times 16$	Concat w/ Image scale 64 × 64	$4 * ndf + 3 \times 64 \times 64$
Conv(k=3, s=1, p=1, b=False)	3 × 16 × 16	DisGeneralConvBlock(4 * ndf , 4, 8 * ndf)	$8 * ndf \times 32 \times 32$
Tanh	No change shape	Concat w/ Image scale 32 × 32	$8 * ndf + 3 \times 32 \times 32$
Generator 32 × 32		DisGeneralConvBlock(8 * ndf , 4, 8 * ndf)	$8 * ndf \times 16 \times 16$
Up Block	$2 * ngf \times 32 \times 32$	Concat w/ Image scale 16 × 16	$8 * ndf + 3 \times 16 \times 16$
Conv(k=3, s=1, p=1, b=False)	3 × 32 × 32	DisGeneralConvBlock(8 * ndf , 4, 8 * ndf)	$8 * ndf \times 8 \times 8$
Tanh	No change shape	Concat w/ Image scale 8 × 8	$8 * ndf + 3 \times 8 \times 8$
Generator 64 × 64		DisGeneralConvBlock(8 * ndf , 4, 8 * ndf)	$8 * ndf \times 4 \times 4$
Up Block	$ngf \times 64 \times 64$	<i>Unconditional logits</i>	
Conv(k=3, s=1, p=1, b=False)	3 × 64 × 64	Conv(k=4, s=4, p=0, b=True)	1
Tanh	No change shape	Sigmoid	1
Generator 128 × 128		<i>Conditional logits</i>	
<i>Input</i>		Caption Embedding	nef
Previous hidden features	$ngf \times 64 \times 64$	Concat w/ replicated caption embedding	$8 * ndf + nef \times 4 \times 4$
Word Mask	$word_num$	Block3x3.LeakyReLU	$8 * ndf \times 4 \times 4$
Word features	$nef \times word_num$	Conv(k=4, s=4, p=0, b=True)	1
<i>Computation</i>		Sigmoid	1
Spatial Attention Layer	$ngf \times 64 \times 64$		
Residual Block × residual_num	$ngf \times 64 \times 64$		
Concat w/ previous hidden features	$2 * ngf \times 64 \times 64$		
Up Block $ngf \times 128 \times 128$			
Conv(k=3, s=1, p=1, b=False)	3 × 128 × 128		
Tanh	No change shape		
Generator 256 × 256			
<i>Input</i>			
Previous hidden features	$ngf \times 128 \times 128$		
Word Mask	$word_num$		
Word features	$nef \times word_num$		
<i>Computation</i>			
Spatial Attention Layer	$ngf \times 128 \times 128$		
Residual Block × residual_num	$ngf \times 128 \times 128$		
Concat w/ previous hidden features	$2 * ngf \times 128 \times 128$		
Up Block	$ngf \times 256 \times 256$		
Conv(k=3, s=1, p=1, b=False)	3 × 256 × 256		
Tanh	No change shape		

Table 14. Training settings of both AttnGAN++ and counter model. Most of settings in evaluation process is the same with training process except word_num. In the evaluation process, word_num=25 for the CUB [53] dataset and word_num=20 for the MS-COCO [27] dataset.

Dataset	CUB [53]	MS-COCO [27]
Optimizer	Adam($\beta_1 = 0.5, \beta_2 = 0.999$)	Adam($\beta_1 = 0.5, \beta_2 = 0.999$)
Generator (G) Learning Rate	0.0002	0.0002
Discriminator (D) Learning Rate	0.0002	0.0002
G/D Update	1 : 1	1 : 1
γ_1	4.0	4.0
γ_2	5.0	5.0
γ_3	10.0	10.0
λ	5.0	50.0
residual_num	2	3
ngf	64	64
ndf	32	32
nef	256	256
nzf	100	100
ncf	100	100
max_epochs	800	200
word_num	18	12



Fig. 12. Visualization of generated images from captions on the test set of CUB [53] dataset by **our AttnGAN++** using t-SNE [28]. The number of clusters in the visualization shows that the photos generated by our model span a wide range of bird species. As a result of the similar appearances of various bird species, some clusters are near together and overlap slightly. We also found no intra-class mode dropping, indicating that the model does not create the same sample in each bird class over and over. As can be seen in each cluster, the samples are belonging to one bird class with a variety of poses, and backgrounds. Best viewed in zoom.

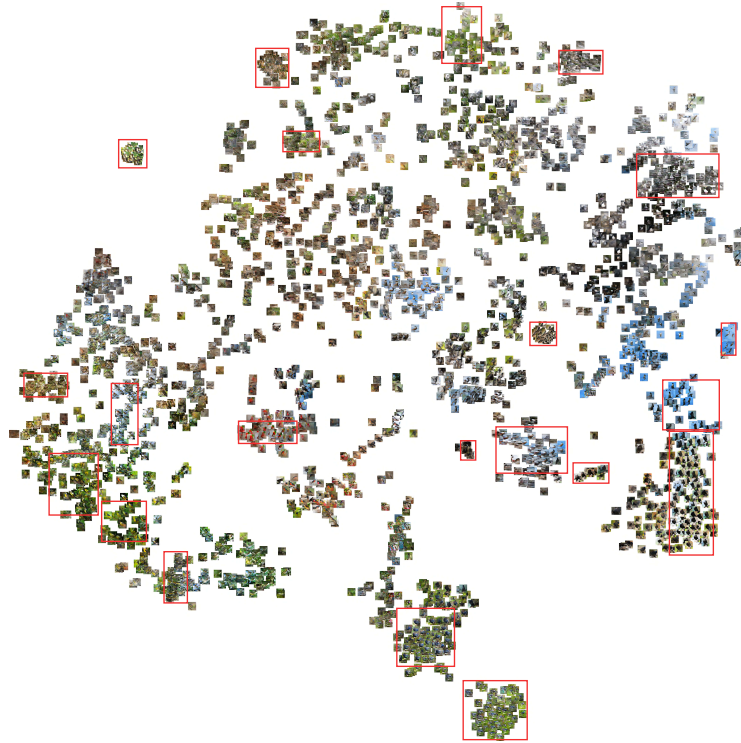


Fig. 13. Visualization of generated images from captions on the test set of CUB [53] dataset by **counter model** using t-SNE [28]. As can be seen, the images of counterexample are not realistic. The counter model tends to generate only one sample again and again per class that are surrounded by red squares in the visualization. Best viewed in zoom.

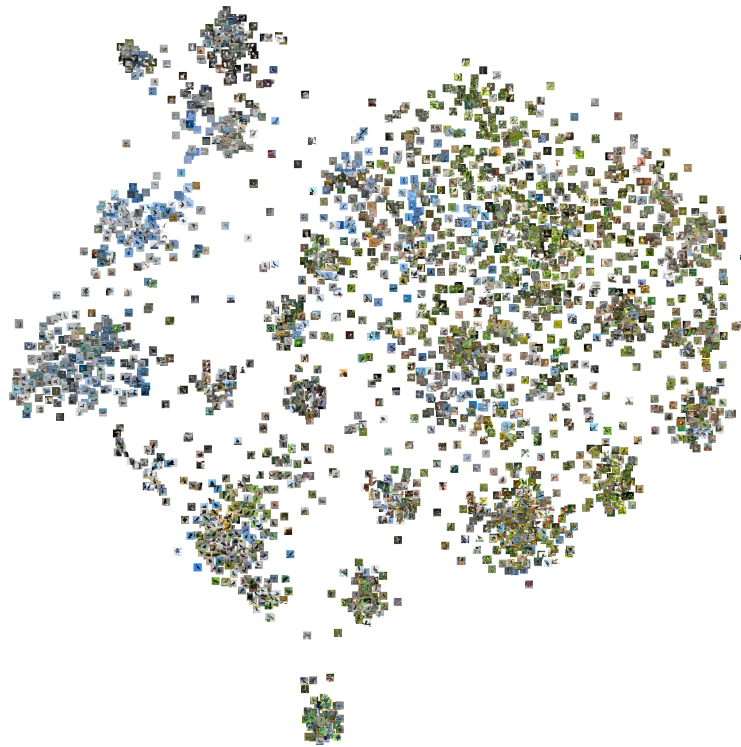


Fig. 14. Visualization of **real images** from CUB [53] test set by using t-SNE [28]. Best viewed in zoom.