TEXT AS NEURAL OPERATOR: IMAGE MANIPULATION BY TEXT INSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

In recent years, text-guided image manipulation has gained increasing attention in the image generation research field. Recent works have proposed to deal with a simplified setting where the input image only has a single object and the text modification is acquired by swapping image captions or labels. In this paper, we study a setting that allows users to edit an image with multiple objects using complex text instructions. In this image generation task, the inputs are a reference image and an instruction in natural language that describes desired modifications to the input image. We propose a GAN-based method to tackle this problem. The key idea is to treat text as neural operators to locally modify the image feature. We show that the proposed model performs favorably against recent baselines on three public datasets.

1 INTRODUCTION

Image synthesis from text has been a highly active research area. This task is typically set up as a conditional image generation problem where a Generative Adversarial Network (GAN) (Goodfellow et al., 2014) is learned to generate realistic images according to the text description in the format of natural languages (Zhang et al., 2018; Xu et al., 2018; Zhu et al., 2019; Li et al., 2019c), scene graphs (Johnson et al., 2018; Yikang et al., 2019), or other modalities (Li et al., 2019d; Nam et al., 2018; Li et al., 2020).

In this paper, we study *how to manipulate image content through complex text instruction*. In this setting, a user is able to apply various changes to a reference image to add, remove, or modify its content by sending text instructions. For example, Figure 1 shows the generated images by our model for three instructions: 1) adding a new object at a location, 2) removing an object, and 3) changing the object's attributes (size, shape, color, etc).

The closest related problem to ours is text-guided image manipulation (Nam et al., 2018; Li et al., 2020) which demonstrates promising image manipulation quality from the text. Sequential text-to-image generation known as GeNeVA (El-Nouby et al., 2019), which focuses on sequentially adding objects to a blank canvas given step-by-step text instructions, is also related, as each step can be seen as doing text-guided image manipulation on the intermediate image. However, the language in existing works is limited in complexity and diversity which consists of either descriptive attributes (Nam et al., 2018; Li et al., 2020) or a single "add" operation (El-Nouby et al., 2019). Different from previous works, this paper focuses on modeling the *complex instruction* for image manipulation. The studied text instructions involve adjectives (attributes), verbs (actions) and adverbs (locations) for three representative operations "add", "modify", and "remove". In addition, the complex instruction often specifies changes to only one of the many objects in the reference image as opposed to the single salient object in prior works (Nam et al., 2018; Li et al., 2020).

Image manipulation by complex instruction is inspired by cross-modal image retrieval which comprises a variety of applications such as product search (Kovashka et al., 2012; Zhao et al., 2017; Guo et al., 2018). In this retrieval setting (Vo et al., 2019), users search an image database using an input query that is formed of an image plus some text that describes complex modifications to the input image. Cross-modal retrieval is essentially the same as our problem except it aims at retrieving as opposed to generating the target image. Interestingly, as we will show, the generated image can be used to retrieve target images with competitive accuracy, providing a more explainable search experience that allows users to inspect the result before the retrieval.

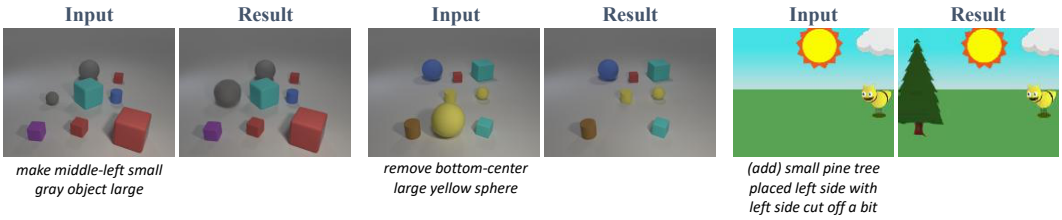


Figure 1: **Image manipulation by text instruction.** Each input contains a reference image and a text instruction. The results are synthesized images by our model.

The main research question studied in this paper is how to model the *complex text instructions* for effective conditional image manipulation. To this end, we propose an approach called Text-Instructed Manipulation GAN or TIM-GAN. The key idea is to treat language as *neural operators* to modify the image feature in a way such that the modified feature is useful in synthesizing the target image by the GAN model. The generation process is decomposed into where and how to edit the image feature. For "where to edit", we leverage existing attention mechanisms to ground words to a spatial region in the image. Although the use of spatial attention is not new, we find it allows for learning generic neural operators that can be decoupled from specific locations. For "how to edit", we introduce a novel text-adaptive routing network to generate text operators for complex instructions. For a text instruction, a route is dynamically created serving as a neural operator to modify the image feature. Since similar instructions perform similar operations, the text-adaptive routing network allows neural blocks to be shared among similar instructions, while still being able to distinguish among different operations.

Experimental results on three datasets, including Clevr (Vo et al., 2019), Abstract scene (Zitnick & Parikh, 2013), and Cityscapes (Cordts et al., 2016) demonstrate that image manipulation by the proposed approach outperforms baseline approaches by large margins in terms of Fréchet Inception Distance (FID) (Heusel et al., 2017) and Retrieval Scores (Xu et al., 2018). The user study validates our method's efficacy in generating more realistic and semantic relevant images. We also conduct ablation studies to substantiate the performance gain stems from the proposed neural operators.

2 Related Work

Conditional generative adversarial networks. Generative adversarial networks GANs (Goodfellow et al., 2014; Mao et al., 2017; Arjovsky et al., 2017; Brock et al., 2019) have made rapid progress on image generation in recent years. Built on the basis of GANs, the *conditional* GAN aims to synthesize the image according to the input context. The input context can be images (Isola et al., 2017; Zhu et al., 2017a; Lee et al., 2020; Huang et al., 2018; Mejjati et al., 2018), audio sequences (Lee et al., 2019), human poses (Ma et al., 2017), or semantic segmentation masks (Wang et al., 2018; Park et al., 2019; Li et al., 2019b). Particularly, text-to-image synthesis (Zhang et al., 2018; Johnson et al., 2018; Xu et al., 2018; Zhu et al., 2019; Li et al., 2019c; Yikang et al., 2019; Li et al., 2019a) learns a mapping from textual descriptions to images. Recently, GeNeVA (El-Nouby et al., 2019) extended the mapping for iterative image generation in which new objects are added one-by-one to a blank canvas following textual descriptions. Different from text-to-image synthesis, the proposed problem takes multimodal inputs, aiming at learning to *manipulate* image content through text instructions.

Conditional image manipulation. The goal is to manipulate image without degrading the quality of the edited images. To enable user-guided manipulation, a variety of frameworks (Zhang et al., 2016; 2017; Huang & Belongie, 2017; Li et al., 2018; Hung et al., 2018; Portenier et al., 2018; Chang et al., 2018; Nam et al., 2018; Li et al., 2020) have been proposed to use different control signals. For instance, Zhang et al. (Zhang et al., 2017) uses sparse dots to guide the image colorization process. There are additional works on image manipulation by bounding boxes subsequently refined as semantic masks (Hong et al., 2018) or by code (Mao et al., 2019). Numerous image stylization (Huang & Belongie, 2017; Li et al., 2018) and blending (Hung et al., 2018) approaches augment the images by referencing an exemplar image. Closest to ours are the TA-GAN (Nam et al., 2018) and ManiGAN (Li et al., 2020) schemes that take the image caption as input to describe attributes for conditional image manipulation. In this work, we propose to manipulate the images according to complex text *instructions*. Different from the image caption used by the TA-GAN and ManiGAN methods, the

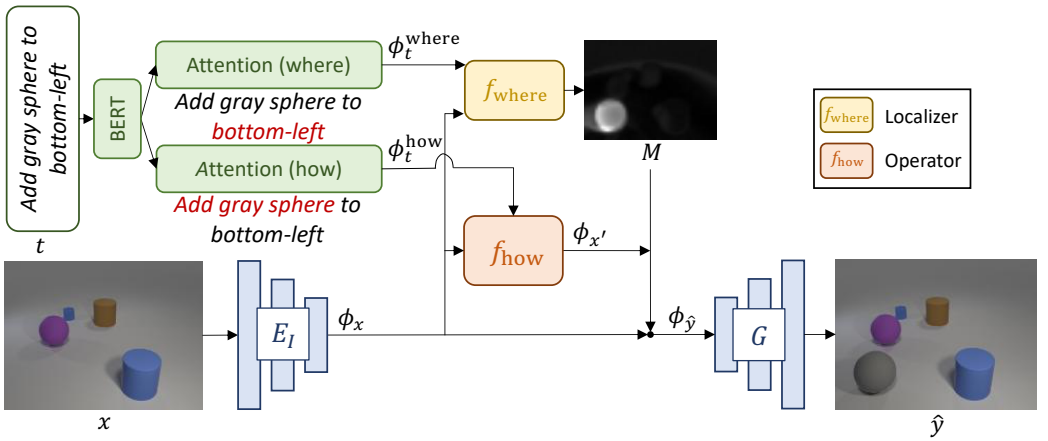


Figure 2: **Method overview.** Given an input image x and a text instruction t, the proposed TIM-GAN first predicts a spatial attention mask M (*where* to edit, Section 3.2) and a text operator f_{how} (*how* to edit, Section 3.1). The image feature ϕ_x is then modified by the text operator f_{how} on the predicted mask M. Finally, the edited image $\hat{y}$ is synthesized from the manipulated image feature $\phi_{\hat{y}}$.

instruction we take as input specifies 1) the region of the image to be edited (*where*) and 2) the type of editing to be conducted (*how*).

Feature Composition. The key idea of this work is to model text as operator. This can be seen as a feature composition function to combine the image and text features for image generation. Feature composition has been studied more extensively in other problems such as visual question answering (Kim et al., 2016; Noh et al., 2016; Chen et al., 2020; Liang et al., 2019), visual reasoning (Johnson et al., 2017b; Santoro et al., 2017), image-to-image translation (Zhu et al., 2017b; Lee et al., 2020), etc. In this work, we design a routing mechanism for image generation such that the intermediate neural blocks can be effectively shared among similar text operators. Our method is related to feature-wise modulation, a technique to modulate the features of one source by referencing those from the other. Examples of recent contributions are: text image residual gating (TIRG) (Vo et al., 2019), feature-wise linear modulation (FiLM) (Perez et al., 2018), and feature-wise gating (Ghosh et al., 2019). Among numerous existing works on feature composition, this paper compares the closely related methods including a state-of-the-art feature composition method for image retrieval (Vo et al., 2019) and three strong methods for conditional image generation (Zhu et al., 2019; Nam et al., 2018; El-Nouby et al., 2019), in additional to the standard routing mechanism (Rosenbaum et al., 2018) in the ablation study.

3 Methodology

Our goal is to manipulate a given reference image according to the modification specified in the input text instruction which specifies one of the three operations "add", "modify", and "remove" We accomplish this task by modeling instructions as neural operators to specify *where* and *how* to modify the image feature.

An overview of the proposed TIM-GAN method is illustrated in Figure 2. Given the input image x and text instruction t, we first extract the image feature ϕ_x along with the text features ϕ_t^{where} and ϕ_t^{how}. The text features ϕ_t^{where} and ϕ_t^{how} encode the *where* and *how* information about the modification, respectively. To indicate the region on the image x to be edited, we predict a spatial attention mask M from ϕ_t^{where}. Thereafter, we design a new network routing mechanism for building an operator f_{how}, from the feature ϕ_t^{how}, to modulate the feature editing. Finally, the resulting image $\hat{y}$ is generated from the manipulated image feature $\phi_{\hat{y}}$ using the generator G.

Although spatial attention has been commonly used in GAN models, we find that by disentangling how from where in the modification, our model learns more generic text operators that can be applied at various locations. To be more specific, let M be a learned spatial mask. The image feature ϕ_x is modified by:

$$\phi_{\hat{y}} = (1 - M) \odot \phi_x + M \odot f_{\text{how}}(\phi_x, \phi_t^{\text{how}}; \Theta_{\text{how}}(t)), \tag{1}$$

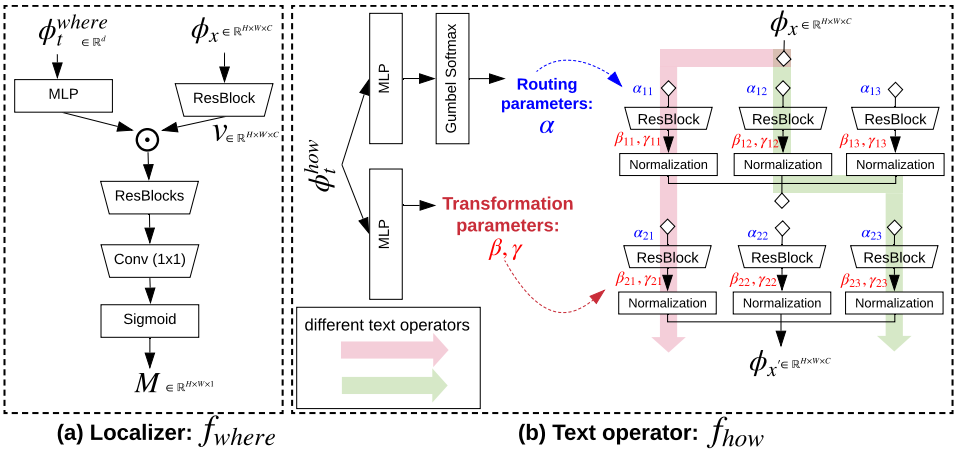


Figure 3: ***Where* and *how* to edit.** (a) The calculation of spatial mask M from text feature ϕ_t^{where} and image feature ϕ_x. (b) The proposed text-adaptive routing mechanism executes various paths as text operators. The operator is parameterized by (α, β, γ) generated from text feature ϕ_t^{how}.

where $\odot$ is element-wise dot product. The first term is a gated identity establishing the input image feature as a reference to the output modified feature.

The second term f_{how} is the proposed neural operator function which embodies the specific computation flow over the image feature (i.e., how to modify). We introduce a new text-adaptive router to execute a sequence of neural blocks dynamically for each text instruction. A route is parameterized by $\Theta_{\text{how}}(t)$ that is generated from ϕ_t^{how}; the remaining parameters are shared across all text instructions.

For training, we use the standard conditional GAN – the pix2pix (Isola et al., 2017) model, which consists of an adversarial loss $\mathcal{L}_{\text{GAN}}$ and an ℓ_1 reconstruction loss called $\mathcal{L}_{L1}$. The weights to $\mathcal{L}_{\text{GAN}}$ and $\mathcal{L}_{L1}$ are set to 1 and 10, receptively. In the rest of this section, we will detail the computation of M and f_{how}.

3.1 How to Edit: Text-Adaptive Routing

Instructions are not independent. Similar instructions perform similar operations, e.g., "add a large cylinder" and "add a red cylinder". Motivated by this idea, we model text operators in a routing network (Rosenbaum et al., 2018) where the text feature is used to dynamically select a sequence of neural blocks (or a path). Our routing network is illustrated in Figure 3b which has l layers of m blocks of identical structures. Each block consists of a `conv` layer followed by an instance normalization layer (Ulyanov et al., 2017). The routing parameter α_i decides to connect or disconnect a block in a layer. An execution path is hence parameterized by a series of α for all layers.

Different from prior routing mechanisms (Rosenbaum et al., 2018; Ahmed & Torresani, 2019; Newell et al., 2019), ours is text-adaptive which selects not only a path but also the associated parameters along the path. To be specific, in addition to α, text features also generate β and γ to perform text-specific normalization in the selected block. This design increases the learning capacity of text operators, while still allowing blocks to be shared among similar instructions. Our idea is partially inspired by the success of style transfer methods (Huang & Belongie, 2017).

Ideally, the path selector α can only take discrete values. However, this approach is not differentiable, and continuous approximation needs to be applied. To do so, we adopt the Gumbel-Max trick (Jang et al., 2017) to sample a block from a categorical distribution. Let $\pi \in \mathbb{R}^m_{>0}$ be the categorical variable with probabilities $P(\alpha = i) \propto \pi_i$ which indicates the probability for selecting block i. We have:

$$\arg\max_i [P(\alpha = i)] = \arg\max_i [g_i + \log \pi_i] = \arg\max_i [\hat{\pi}_i], \tag{2}$$

where $g_i = -\log(-\log(u_i))$ is a re-parameterization term, and $u_i \sim \text{Uniform}(0, 1)$. To make it differentiable, the argmax operation is approximated by a continuous softmax operation: $\alpha = \text{softmax}(\hat{\pi}/\tau)$, where τ is the temperature controlling the degree of the approximation.

Then, a text operator can be parameterized by $\Theta_{\text{how}}(t)$ defined in Equation 1 as:

$$\Theta_{\text{how}}(t) = f_{\text{MLP}}(\phi_t^{\text{where}}) = \{(\alpha_i, \beta_i, \gamma_i) | \alpha_i \in [0, 1]^m, \gamma_i, \beta_i \in \mathbb{R}^{m \times p}, i \in \{1, \cdots, l\}\}, \tag{3}$$

where the text feature ϕ_t^{where} generates real vectors $\alpha_i, \beta_i, \gamma_i$ for text-adaptive routing for all layers, p is the number of normalization parameters for each block.

Finally, as shown in Figure 3, the image feature is modified by:

$$a^{(i+1)} = \sum_{j=1}^{m} \alpha_{ij}(\gamma_{ij}\frac{o_{ij} - \mu(o_{ij})}{\delta(o_{ij})} + \beta_{ij}), \tag{4}$$

where o_{ij} is the output of the j-th `conv` block in layer i. δ and μ compute channel-wise mean and variance across spatial dimensions, and are applied at test time unchanged. The operator in Equation 4 takes the input of $a^{(1)} = \phi_x$ and outputs the modified image feature as $a^{(l)}$.

3.2 WHERE TO EDIT: SPATIAL MASK

We use the standard scaled dot-product self-attention (Vaswani et al., 2017) to summarize the location-indicative words in an instruction. Let $S = [w_1, \cdots, w_l] \in \mathbb{R}^{l \times d_0}$ denote the instruction where $w_i \in \mathbb{R}^{d_0}$ is the BERT embedding (Devlin et al., 2018) for the i-th word. The query, key and value in the attention are computed by:

$$Q = SW_Q, \qquad K = SW_K, \qquad V = SW_V \tag{5}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d_0 \times d}$ are weight matrices to learn, and d is the output dimension. After reducing matrix Q to a column vector $\hat{q}$ by average pooling along its first dimension, we obtain the attended text embedding by:

$$\phi_t^{\text{where}} = V^T \text{softmax}(\frac{K\hat{q}}{\sqrt{d}}), \tag{6}$$

in which the softmax function is supposed to assign higher attention weights for locational words. Likewise, we obtain the text feature ϕ_t^{how} for salient operational words in the instruction (e.g., "add", "red", "cylinder"), computed by a separate self-attention head similar to that of ϕ_t^{where}.

After that, we pass the image feature ϕ_x to a convolution block (e.g., a ResBlock (He et al., 2016)) to get the output $v \in \mathbb{R}^{H \times W \times C}$. The spatial mask is then computed from ϕ_t^{where} using image features as the context:

$$M = f_{\text{where}}(\phi_x, \phi_t^{\text{where}}) = \delta(W_m * (f_{\text{MLP}}(\phi_t^{\text{where}}) \odot v)) \in [0, 1]^{H \times W \times 1}, \tag{7}$$

where σ is the sigmoid function, $*$ represents the 2d-convolution product with kernel W_m (see Figure 3a). We use two layers of MLP with the ReLU activation. The spatial attention can be derived from M by performing ℓ_1 normalization over the spatial dimensions. In this paper, we choose to use the unnormalized mask for improved generalization performance.

In training, we also use an ℓ_1 loss to penalize the distance between the predicted mask M and the noisy true mask, and assign it the same weight as the $\mathcal{L}_{L1}$ reconstruction loss. Note that computing this loss needs no additional supervision as the noisy mask is automatically computed by comparing the difference between the input and ground-truth training images.

4 EXPERIMENTAL RESULTS

We conduct experiments to quantitatively and qualitatively compare the proposed method with baseline approaches. Additional qualitative results are presented in the supplementary material. We will release the source code and dataset to facilitate further research in this field.

4.1 EXPERIMENTAL SETUPS

Datasets. We use three public datasets: Clevr (Vo et al., 2019), Abstract scene (Zitnick & Parikh, 2013), and Cityscapes (Cordts et al., 2016). All datasets consist of images of multiple objects accompanied by complex text instructions. Since there is no dataset of text instructions on real-world RGB images (i.e., providing the ground-truth manipulated images for the inputs of a text instruction and a reference image), existing works (El-Nouby et al., 2019; Li et al., 2019e) were only able to

Table 1: **Quantitative comparisons.** We use the FID scores to measure the realism of the generated images, and the retrieval score (RS) to estimate the correspondence to text instructions.

Method	Clevr			Abstract scene			Cityscape		
	FID ↓	RS@1 ↑	RS@5 ↑	FID ↓	RS@1 ↑	RS@5 ↑	FID[1] ↓	RS@1 ↑	RS@5 ↑
DM-GAN	**27.9**	$1.6_{\pm 0.1}$	$5.6_{\pm 0.1}$	53.8	$2.1_{\pm 0.1}$	$6.6_{\pm 0.1}$	18.7	$4.6_{\pm 0.2}$	$15.7_{\pm 0.2}$
TIRG-GAN	34.0	$\underline{48.5}_{\pm 0.2}$	$\underline{68.2}_{\pm 0.1}$	52.7	$23.5_{\pm 0.1}$	$38.8_{\pm 0.1}$	$\underline{6.1}$	$25.0_{\pm 0.3}$	$\underline{88.9}_{\pm 0.3}$
TA-GAN	58.8	$40.8_{\pm 0.1}$	$64.1_{\pm 0.1}$	$\underline{44.0}$	$\underline{26.9}_{\pm 0.2}$	$\underline{46.3}_{\pm 0.1}$	6.7	$\underline{36.8}_{\pm 0.4}$	$79.8_{\pm 0.3}$
GeNeVA	46.1	$34.0_{\pm 0.1}$	$57.3_{\pm 0.1}$	72.2	$17.3_{\pm 0.2}$	$31.6_{\pm 0.2}$	10.5	$14.5_{\pm 0.4}$	$46.1_{\pm 0.3}$
Ours	$\underline{33.0}$	$\mathbf{95.9}_{\pm 0.1}$	$\mathbf{97.8}_{\pm 0.1}$	**35.1**	$\mathbf{35.4}_{\pm 0.2}$	$\mathbf{58.7}_{\pm 0.1}$	**5.9**	$\mathbf{77.2}_{\pm 0.4}$	$\mathbf{99.9}_{\pm 0.1}$
Real images	17.0	100	100	14.0	100	100	4.4	100	100

test on synthetic images. Therefore we extend our method to manipulate semantic segmentation in Cityscapes. By doing so, we show the potential of our method for synthesizing RGB images from the modified segmentation mask. We describe details about these datasets in the Appendix.

Baselines. We compare to the following baseline approaches in our experiments. All methods including ours are trained and tested on the same datasets, implemented by their official code or adapted official code. More details about the baseline comparison are discussed in the Appendix.

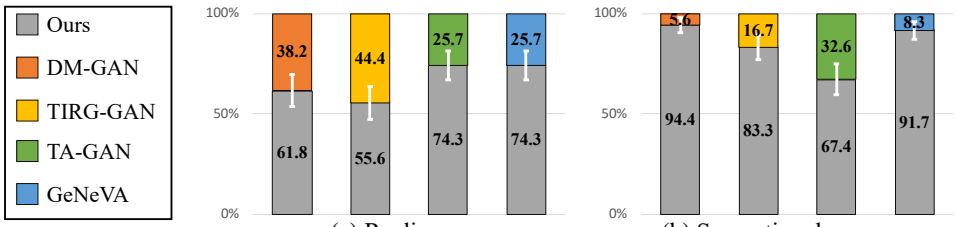
- **DM-GAN**: The DM-GAN (Zhu et al., 2019) model is a recent text-to-image synthesis framework. To adapt it to our task, we use our image encoder to extract the image feature and concatenate it with its original text feature as its input signal.
- **TIRG-GAN**: TIRG (Vo et al., 2019) is a state-of-the-art method for the cross-modal image retrieval task. It takes the same input as ours but only produces the image feature for retrieval. We build a baseline TIRG-GAN based on TIRG by using our image decoder G to synthesize the image from the feature predicted by the TIRG model.
- **TA-GAN**: TA-GAN (Nam et al., 2018) is trained by learning the mapping between the caption and the image. The manipulation is then conducted by changing the caption of the image. Since there is no image caption in our task, we concatenate the pre-trained features of the input image and text instruction as the input caption feature for the TA-GAN model.
- **GeNeVA**: GeNeVA (El-Nouby et al., 2019) learns to generate the image step-by-step according to the text description. To adapt it to take the same input as all the other methods, we use it for single-step generation over the real input image.

Metrics. In all experiments, we use the Fréchet Inception Distance score (FID) to measure the realism of the edited images Heusel et al. (2017), and the retrieval score (RS) to estimate the correctness of the manipulation. For the retrieval score, we employ the evaluation protocols similar to (Vo et al., 2019; Xu et al., 2018). Specifically, we use the generated image as a query to retrieve the target images in the test set. We extract the image features of all query and target images by an autoencoder pre-trained on each dataset and use simple cosine similarity between their feature embedding as our retrieval metric. The score RS@N indicates the recall of the ground-truth image in the top-N retrieved images. The computations of FID and RS scores are detailed in the Appendix.

4.2 Quantitative Results

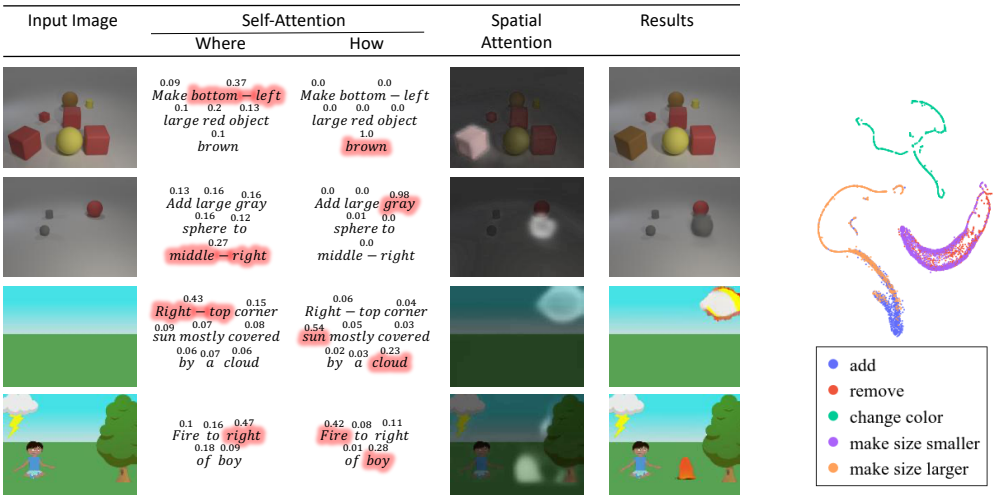
Realism and Retrieval Score. The results are shown in Table 1. The proposed method performs favorably against all baseline approaches across datasets. Although DM-GAN appears to generate more realistic images on the Clevr dataset, its retrieval scores are very poor ($< 2\%$), indicating it merely memorizes random images without properly editing the input image. In comparison, our approach achieves a decent realism score as well as significantly higher retrieval scores.

User preference study. We conduct two user studies to understand the visual quality and semantic relevance of the generated content. Given a pair of images generated by two different methods, users are asked to choose 1) which one looks more *realistic* while ignoring the input image and text; 2) which one is more relevant to the text instruction by comparing the *content* of the generated and the ground-truth image. In total, we collect 960 answers from 30 users. As shown in Figure 4, the proposed TIM-GAN outperforms other methods by a large margin in both metrics.



(a) Realism (b) Semantic relevance

Figure 4: **User preference studies.** We present manipulated images on the Clevr and abstract scene datasets and ask the users to select the one which (a) is more *realistic* and (b) is more *semantically relevant* to the ground-truth image.



(a) Text and spatial attention (b) Text Operators

Figure 5: **Where and how to edit.** (a) We visualize the predicted self-attention weights and spatial attention masks. The self-attention weights are labeled above each word, and highlighted if the weights are greater than 0.2. (b) We show the t-SNE visualization of the routing parameters α predicted from different types of instructions on the Clevr dataset.

Ablation study. Results are shown in Table 2. We verify three of our key designs by leaving the module out from the full model. (1) the learned mask M is removed and replaced with an identity matrix; (2) the f_{how} operator is substituted with a fixed network with the same number of layers and parameters that takes the input of concatenated features of image and text; (3) We examine the standard routing by treating the text-adaptive parameters β, γ as latent variables in our full model.

The results show the following. First, removing f_{how} from our approach leads to worse performance than the baseline methods in Table 1, which indicates the performance gain is primarily resulted from the proposed text operator as opposed to the network backbone or BERT embedding. Second, there is a sharp drop when text-adaptive routing is removed. This is more evident on Clevr in which text instructions are more diverse. These results substantiate the efficacy of text-adaptive in modeling complex text instructions. Finally, though f_{where} is not a crucial component, it complements the learning of generic f_{how} that is decoupled from specific spatial locations.

4.3 Qualitative Results

Qualitative results are shown in Figure 6. As shown, TA-GAN and TIRG-GAN tend to copy the reference images. DM-GAN often generates random objects following similar input layouts. GeNeVA can make local modifications to images, but often does not follow the text instructions. By comparison, our model generates images guided by the text instructions with better quality.

Figure 5 visualizes our intermediate results for where and how to edit. The former is shown by the text self-attention and spatial attention in Figure 5a. Figure 5b shows the t-SNE plot of the routing parameters. As shown in Figure 5b, instructions of similar types are grouped together, suggesting

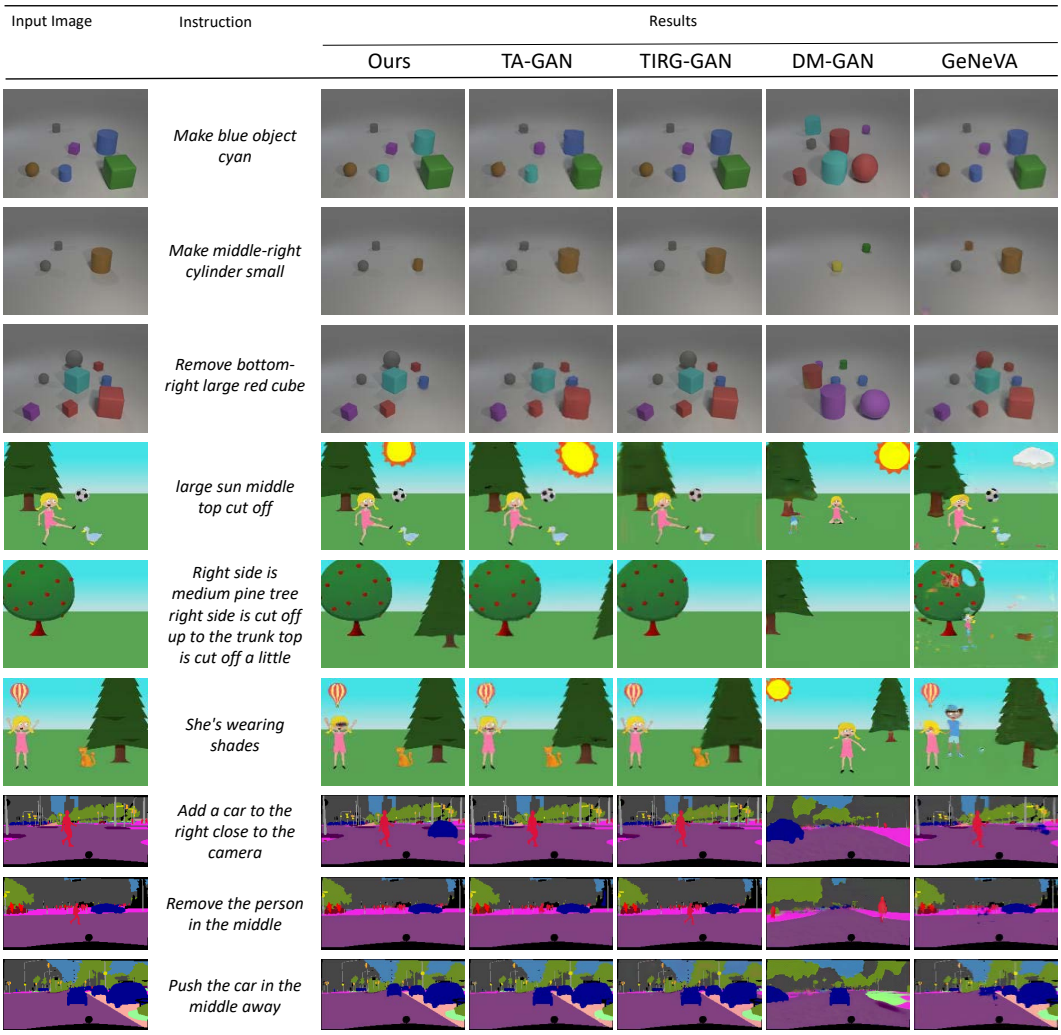


Figure 6: **Selected generation results.** We show the manipulation results by different approaches on the Clevr (*top*), Abstract scene (*middle*), and Cityscapes (*bottom*) datasets.

Table 2: **Ablation Studies.** Performance on ablated versions of our model.

Methods	f_{where}	f_{how}		Clevr			Abstract scene		
		text-adaptive	non-adaptive	FID ↓	R@1 ↑	R@5 ↑	FID ↓	R@1 ↑	R@5 ↑
Ours Full	✓	✓	✗	33.0	$95.9_{\pm 0.1}$	$97.8_{\pm 0.1}$	35.1	$35.4_{\pm 0.2}$	$58.7_{\pm 0.1}$
no f_{where}	✗	✓	✗	34.8	$81.7_{\pm 0.1}$	$89.6_{\pm 0.1}$	48.7	$28.7_{\pm 0.1}$	$44.4_{\pm 0.1}$
no f_{how}	✓	✗	✗	34.7	$49.5_{\pm 0.1}$	$67.4_{\pm 0.1}$	36.0	$33.8_{\pm 0.2}$	$56.7_{\pm 0.2}$
no text-adaptive	✓	✗	✓	45.9	$29.9_{\pm 0.2}$	$49.1_{\pm 0.1}$	37.4	$33.1_{\pm 0.2}$	$54.5_{\pm 0.1}$

neural blocks are shared among similar text operators. It is interesting to find our method can automatically uncover the subtle relationship between operators, e.g., "add" and "make size larger" operators are closer indicating more neural blocks are shared between these similar operations.

5 Conclusion

In this paper, we studied a conditional image generation task that allows users to edit an input image using complex text instructions. We proposed a new approach treating text instructions as neural operators to locally modify the image feature. To learn more genetic operators, our method decomposes "where" from "how" to apply the modification (text operator), introducing a new text-adaptive network routing mechanism. We evaluate our method on three datasets and show competitive results with respect to metrics on image quality, semantic relevance, and retrieval performance.

REFERENCES

Karim Ahmed and Lorenzo Torresani. Star-caps: Capsule networks with straight-through attentive routing. In *Neural Information Processing Systems*, 2019.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. In *International Conference on Machine Learning*, 2017.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

Huiwen Chang, Jingwan Lu, Fisher Yu, and Adam Finkelstein. Pairedcyclegan: Asymmetric style transfer for applying and removing makeup. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Yanbei Chen, Shaogang Gong, and Loris Bazzani. Image search with text feedback by visiolinguistic attention learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alaaeldin El-Nouby, Shikhar Sharma, Hannes Schulz, Devon Hjelm, Layla El Asri, Samira Ebrahimi Kahou, Yoshua Bengio, and Graham W. Taylor. Tell, draw, and repeat: Generating and modifying images based on continual linguistic instruction. In *IEEE International Conference on Computer Vision*, 2019.

Arnab Ghosh, Richard Zhang, Puneet K Dokania, Oliver Wang, Alexei A Efros, Philip HS Torr, and Eli Shechtman. Interactive sketch & fill: Multiclass sketch-to-image translation. In *IEEE International Conference on Computer Vision*, 2019.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Neural Information Processing Systems*, 2014.

Xiaoxiao Guo, Hui Wu, Yu Cheng, Steven Rennie, Gerald Tesauro, and Rogerio Feris. Dialog-based interactive image retrieval. In *Neural Information Processing Systems*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Neural Information Processing Systems*, 2017.

Seunghoon Hong, Xinchen Yan, Thomas S Huang, and Honglak Lee. Learning hierarchical semantic image manipulation through structured representations. In *Neural Information Processing Systems*, 2018.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE International Conference on Computer Vision*, 2017.

Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *European Conference on Computer Vision*, 2018.

Wei-Chih Hung, Jianming Zhang, Xiaohui Shen, Zhe Lin, Joon-Young Lee, and Ming-Hsuan Yang. Learning to blend photos. In *European Conference on Computer Vision*, 2018.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017a.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Inferring and executing programs for visual reasoning. In *IEEE International Conference on Computer Vision*, 2017b.

Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Jin-Hwa Kim, Sang-Woo Lee, Donghyun Kwak, Min-Oh Heo, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Multimodal Residual Learning for Visual QA. In *Neural Information Processing Systems*, 2016.

Jin-Hwa Kim, Nikita Kitaev, Xinlei Chen, Marcus Rohrbach, Byoung-Tak Zhang, Yuandong Tian, Dhruv Batra, and Devi Parikh. Codraw: Collaborative drawing as a testbed for grounded goal-driven communication. *arXiv preprint arXiv:1712.05558*, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Adriana Kovashka, Devi Parikh, and Kristen Grauman. Whittlesearch: Image search with relative attribute feedback. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

Hsin-Ying Lee, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz. Dancing to music. In *Neural Information Processing Systems*, 2019.

Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Singh, and Ming-Hsuan Yang. Drit++: Diverse image-to-image translation via disentangled representations. *International Journal of Computer Vision*, 2020.

Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip Torr. Controllable text-to-image generation. In *Neural Information Processing Systems*, 2019a.

Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. Manigan: Text-guided image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Ke Li, Tianhao Zhang, and Jitendra Malik. Diverse image synthesis from semantic layouts via conditional imle. In *IEEE International Conference on Computer Vision*, 2019b.

Wenbo Li, Pengchuan Zhang, Lei Zhang, Qiuyuan Huang, Xiaodong He, Siwei Lyu, and Jianfeng Gao. Object-driven text-to-image synthesis via adversarial training. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019c.

Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In *European Conference on Computer Vision*, 2018.

Yijun Li, Lu Jiang, and Ming-Hsuan Yang. Controllable and progressive image extrapolation. *arXiv preprint arXiv:1912.11711*, 2019d.

Yitong Li, Zhe Gan, Yelong Shen, Jingjing Liu, Yu Cheng, Yuexin Wu, Lawrence Carin, David Carlson, and Jianfeng Gao. Storygan: A sequential conditional gan for story visualization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019e.

Junwei Liang, Lu Jiang, Liangliang Cao, Yannis Kalantidis, Li-Jia Li, and Alexander G Hauptmann. Focal visual-text attention for memex question answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1893–1908, 2019.

Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Neural Information Processing Systems*, 2017.

Jiayuan Mao, Xiuming Zhang, Yikai Li, William T Freeman, Joshua B Tenenbaum, and Jiajun Wu. Program-guided image manipulators. In *IEEE International Conference on Computer Vision*, 2019.

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *IEEE International Conference on Computer Vision*, 2017.

Youssef Alami Mejjati, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim. Unsupervised attention-guided image-to-image translation. In *Neural Information Processing Systems*, 2018.

Seonghyeon Nam, Yunji Kim, and Seon Joo Kim. Text-adaptive generative adversarial networks: manipulating images with natural language. In *Neural Information Processing Systems*, 2018.

Alejandro Newell, Lu Jiang, Chong Wang, Li-Jia Li, and Jia Deng. Feature partitioning for efficient multi-task architectures. *arXiv preprint arXiv:1908.04339*, 2019.

Hyeonwoo Noh, Paul Hongsuck Seo, and Bohyung Han. Image question answering using convolutional neural network with dynamic parameter prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NeurIPS workshop*, 2017.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI Conference on Artificial Intelligence*, 2018.

Tiziano Portenier, Qiyang Hu, Attila Szabo, Siavash Arjomand Bigdeli, Paolo Favaro, and Matthias Zwicker. Faceshop: Deep sketch-based face image editing. *ACM Transactions on Graphics*, 37(4): 99, 2018.

Clemens Rosenbaum, Tim Klinger, and Matthew Riemer. Routing networks: Adaptive selection of non-linear functions for multi-task learning. In *International Conference on Learning Representations*, 2018.

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *Neural Information Processing Systems*, 2017.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *CVPR*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, 2017.

Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, and James Hays. Composing text and image for image retrieval-an empirical odyssey. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

LI Yikang, Tao Ma, Yeqi Bai, Nan Duan, Sining Wei, and Xiaogang Wang. Pastegan: A semi-parametric method to generate image from scene graph. In *Neural Information Processing Systems*, 2019.

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1947–1962, 2018.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision*, 2016.

Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics*, 9(4), 2017.

Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. Memory-augmented attribute manipulation networks for interactive fashion search. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision*, 2017a.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Neural Information Processing Systems*, 2017b.

Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

C Lawrence Zitnick and Devi Parikh. Bringing semantics into focus using visual abstraction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

A More Qualitative Results

A.1 More images generated by our model

We show additional images generated by our model on the three experimental datasets. See Figure 7, Figure 8, and Figure 9 for details. Generally, our model can handle complex text instructions. But we also observe cases in which our method can fail: (a) when the location of the target is not well-specified, see Figure 10 the 8-th row; and when the attribute of the target is not detailed enough, see Figure 10 the 7-th and 9-th row.

A.2 Retrieval results

We use the generated image by our model as a query to retrieve the target image. Figure 10 shows the top-5 retrieved images on the Clevr dataset. We show the successful retrieval cases in the first 5 rows and failure cases in the rest of the rows. The quantitative retrieval performance is reported in Table 1

B Implementation Details

B.1 Dataset Processing

Clevr. We use the CSS dataset (Vo et al., 2019) which was created for the image retrieval task. The dataset is generated using the Clevr toolkit (Johnson et al., 2017a) and contains 3-D synthesized images with the presence of objects with different color, shape, and size. Each training sample includes an input image, an output image and a text instruction specifying the modification. There are three types of modifications: add a new object, remove an existing object, and change the

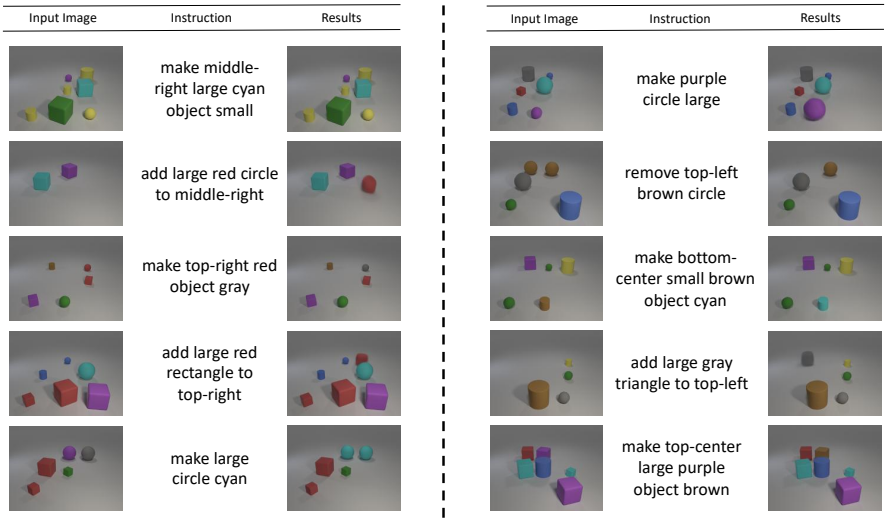


Figure 7: **Examples of the generated image by our model on Clevr.**

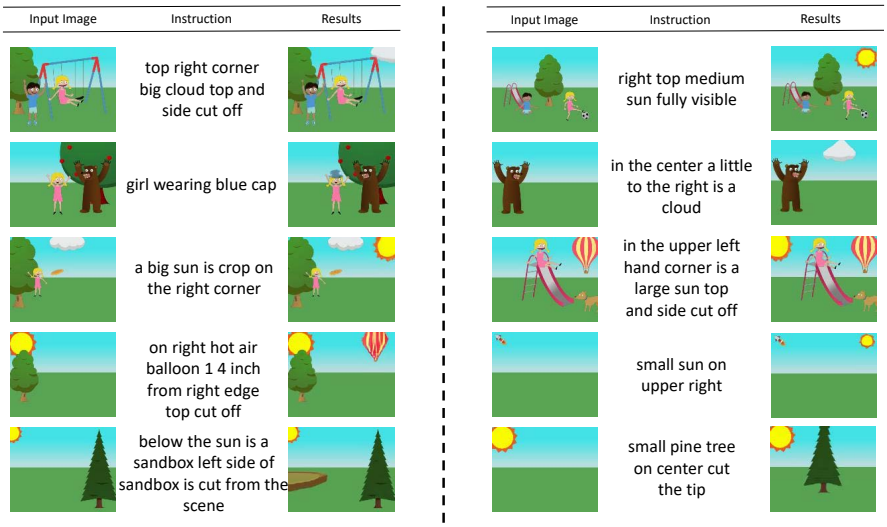


Figure 8: **Examples of the generated image by our model on Abstract Scene.**

attribute of an object. Each text instruction specifies the position of the target object and the expected modification. The dataset includes 17K training data pairs and 17K tests. Note the original rendering of this dataset contains significant camera and object displacements which fail GAN model training of all the methods. In our experiments, we use the re-rendered CSS obtained from Vo et al. (2019) with reduced misalignment for unchanged objects. As a result, we can train meaningful GAN models and compare all methods fairly on the same CSS benchmark.

Abstract scene. CoDraw(Kim et al., 2017) is a synthetic dataset built upon the Abstract Scene dataset(Zitnick & Parikh, 2013). It is formed by sequences of images of children playing in the park. For each sequence, there is a conversation between a Teller and a Drawer. The teller gives text instructions on how to change the current image and the Drawer can ask questions to confirm details and output images step by step. To adapt it to our setting, we extract the image and text of a single step. The dataset consists of 30K training and 8K test instances. Each training sample includes an input image, an output image and a text description about the object to be added to the input image.

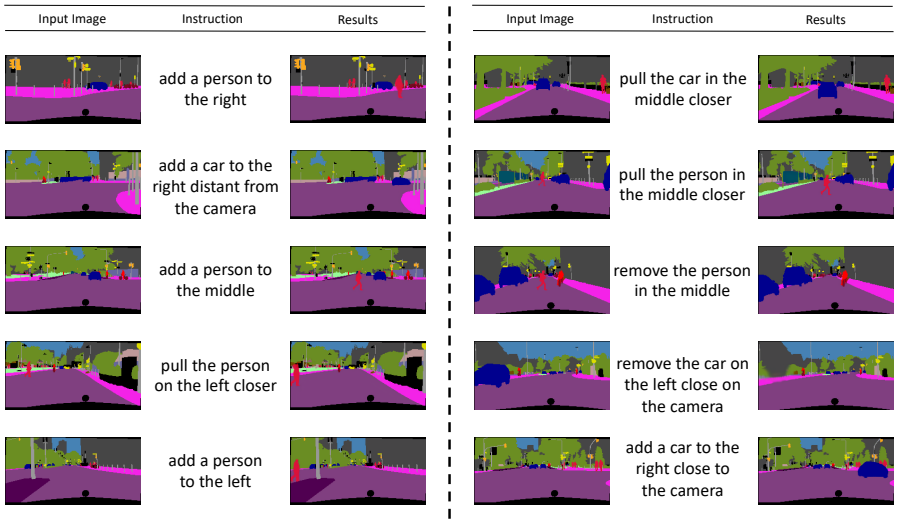


Figure 9: **Examples of the generated image by our model on Cityscapes.**

Cityscapes. We create a third dataset based on Cityscapes segmentation masks. The dataset consists of 4 types of text modifications: "add", "remove", "pull an object closer", and "push an object away". The ground-truth images are manually generated by pasting desired objects on the input image at appropriate positions. We crop out various object prototypes (cars, people, etc.) from existing images. Specifically, adding is done by simply pasting the added object. Removing is the inverse of adding. Pulling and pushing objects are done by pasting the same object of different sizes (with some adjustment on location as well to simulate depth changing effect). The dataset consists of 20K training instances and 3K examples for testing.

B.2 Evaluation Details

FID We employ the standard FID (Heusel et al., 2017) metric based on the InceptionV3 model for the Clevr dataset and the Abstract Scene dataset. On Cityscapes, the FID scores are computed using a pretrained auto-encoder on segmentation masks. We use the encoder to extract features for distance computation, and keep the feature dimension to be the same as the original Inception V3 network to provide a similar scale of the final score.

Retrieval Score First, we extract the features of the edited images using the learned image encoder E_i to get the queries. For each query, we take the ground-truth output image and randomly select 999 real images from the test set, and extract the features of these images using the same model to form a pool for the retrieval task. Second, we compute the cosine similarity between the queries and image features from the pool. We then select the top-N most relevant images from the pool as the candidate set for each query. We report RS@1 and RS@5 scores in our experiments, in which RS@N indicates the recall of the ground-truth image in the top-N retrieved images.

B.3 Experiment Setting

We implement our model in Pytorch (Paszke et al., 2017). For the image encoder E_i, we use three down-sampling convolutional layers followed by Instance Normalization and ReLU activation. We use 3x3 kernels and a stride of 2 for down-sampling convolutional layers. We construct the decoder G by using two residual blocks followed by three up-sampling layers (transposed-convolutional layers) followed by Instance Normalization and ReLU activation. We use 3x3 kernels and a stride of 2 for up-sampling layers.

As for the text encoder E_t, we use the BERT (Devlin et al., 2018) model. We use the cased version of *BERT-Base* released by the authors of the paper for the BERT (Devlin et al., 2018) text encoder. The parameters are initialized by pretraining on a large corpus (Wikipedia + BookCorpus).

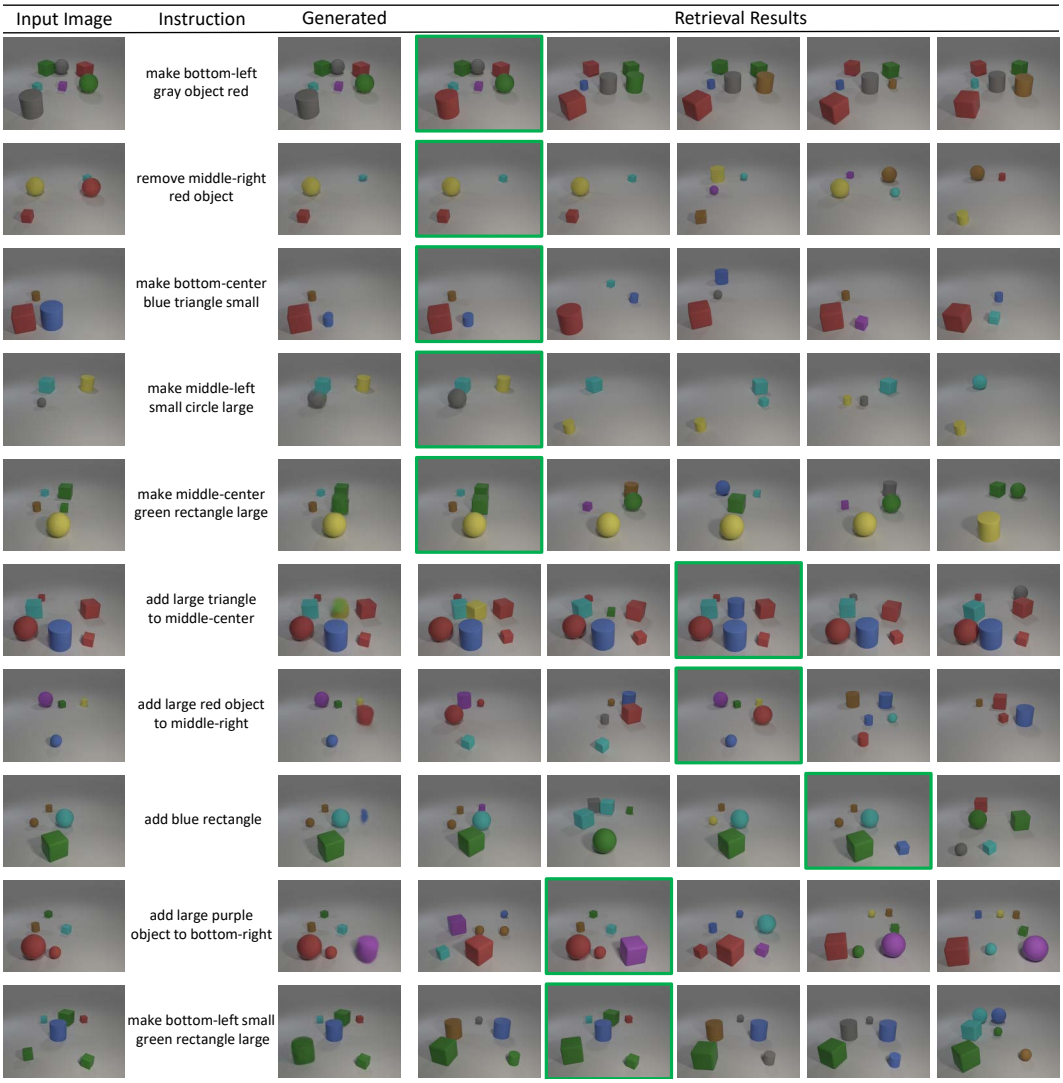


Figure 10: **Retrieval Results.** For each row, top-5 retrieved images are shown. The correct image is highlighted in the green box.

The parameters in the image encoder E_i and decoder G are initialized by training an image autoencoder. Specifically, for each dataset, we pre-train the image encoder and decoder on all images of the dataset. After the initialization, we fix the parameters in the image encoder E_i and optimize the other parts of the network in the end-to-end training. During pretraining of the autoencoder, we use the Adam optimizer (Kingma & Ba, 2015) with a batch size of 8, a learning rate of 0.002, and exponential rates of $(\beta_1, \beta_2) = (0.5, 0.999)$ and train the model for 30 epochs.

The encoded image feature has 256 channels. The BERT output text embedding dimension $d_0 = 768$, and the attended text embedding dimension $d = 512$. The routing network has $l = 2$ layers and $m = 3$ blocks for each layer.

For the training, we use the Adam optimizer (Kingma & Ba, 2015) with a batch size of 16, a learning rate of 0.002, and exponential rates of $(\beta_1, \beta_2) = (0.5, 0.999)$. We use a smaller learning rate of 0.0002 for BERT as suggested in (Devlin et al., 2018). The model is trained for 60 epochs.

C NOTES ON BASELINE MODELS

Implementation: the selected baselines are among the state-of-the-art methods in text-to-image synthesis: DM-GAN[2] (Zhu et al., 2019), iterative text-to-image synthesis GeNeVA[3] (El-Nouby et al., 2019), attribute-based text-guided image manipulation TA-GAN[4] (Nam et al., 2018). TIRG is a recent cross-modal retrieval[5] (Vo et al., 2019) and is adapted for conditional image generation. For these baseline methods, we stick to using their original or adapted official implementation (including their backbone networks and text embeddings) to avoid performance degradation.

DM-GAN is originally used for unconditional text-to-image synthesis and hence has no image input. To adapt it to our task, we add an image encoder to the model and concatenate the image feature and the text feature as the model input. However, to minimize modification on the architecture, the image feature is squeezed into a vector by using global average pooling. Therefore, significant spatial information of the input image is lost, resulting in low consistency between the generated image and the input image. We add the ℓ_1 reconstruction loss and find it improves the performance.

GeNeVA is a sequential image synthesis model. We compare it under one-shot generation on the same Abstract scene dataset used in their paper (El-Nouby et al., 2019). While GeNeVA is only tested on the "add" operation, our method is also verified on other datasets with more diverse and complex text instructions. Applying our method in the sequential generation is non-trivial as it requires the design of extra memory for sequential modeling. Since all baseline methods except GeNeVA do not use memory/state for sequential modeling, we do not evaluate multi-shot generation but leave it as our future work.

[2]code available on `https://github.com/MinfengZhu/DM-GAN`

[3]code available on `https://github.com/Maluuba/GeNeVA`

[4]code available on `https://github.com/woozzu/tagan`

[5]code available on `https://github.com/google/tirg`