Reject Decoding via Language-Vision Models for Text-to-Image Synthesis

Fuxiang Wu[1,2], Liu Liu[3], Fusheng Hao[1,2], Fengxiang He[4], Lei Wang[1, 2], Jun Cheng*[1,2]

[1] Guangdong Provincial Key Laboratory of Robotics and Intelligent System,
Shenzhen Institute of Advanced Technology, CAS, China
[2] The Chinese University of Hong Kong, Hong Kong, China
[3] School of Computer Science, Faculty of Engineering, The University of Sydney, Australia
[4] JD Explore Academy, JD.com Inc., Beijing, China
{fx.wu1, fs.hao, lei.wang1,jun.cheng}@siat.ac.cn, liu.liu1@sydney.edu.au, fengxiang.f.he@gmail.com.

Abstract

Transformer-based text-to-image synthesis generates images from abstractive textual conditions and achieves prompt results. Since transformer-based models predict visual tokens step by step in testing, where the early error is hard to be corrected and would be propagated. To alleviate this issue, the common practice is drawing multi-paths from the transformer-based models and re-ranking the multi-images decoded from multi-paths to find the best one and filter out others. Therefore, the computing procedure of excluding images may be inefficient. To improve the effectiveness and efficiency of decoding, we exploit a reject decoding algorithm with tiny multi-modal models to enlarge the searching space and exclude the useless paths as early as possible. Specifically, we build tiny multi-modal models to evaluate the similarities between the partial paths and the caption at multi scales. Then, we propose a reject decoding algorithm to exclude some lowest quality partial paths at the inner steps. Thus, under the same computing load as the original decoding, we could search across more multi-paths to improve the decoding efficiency and synthesizing quality. The experiments conducted on the MS-COCO dataset and large-scale datasets show that the proposed reject decoding algorithm can exclude the useless paths and enlarge the searching paths to improve the synthesizing quality by consuming less time.

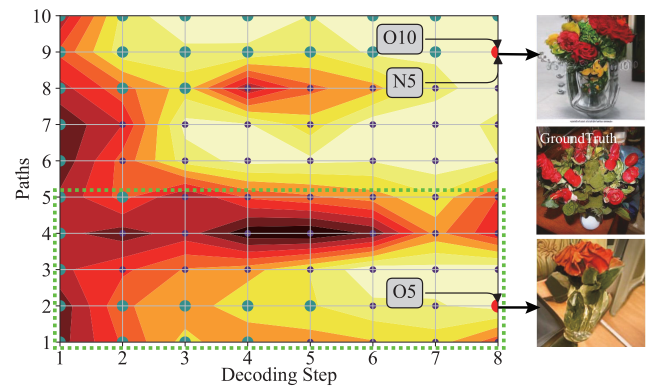

Figure 1: The reject decoding algorithm can explore a larger size of paths under a similar computing load and generate better results. The green box indicates the five paths exploited in traditional decoding, and "O5" is the traditional result. For the rejecting decoding algorithm, there are ten paths for search and the large green circles are the active routines. "N10" is our final result, which is better than "O5" and outside the green box.

Introduction

Text-to-image synthesis is a multimodal task, in which vivid images can be generated from the given textual descriptions (Reed et al. 2016). Many models make use of the Generative Adversarial Networks (GANs) to generate the images (Reed et al. 2016; Zhang et al. 2019; Xu et al. 2018; Zhu et al. 2019; Tao et al. 2022) and have achieved highly promising results. However, GANs are known to have difficulty in achieving stable convergence and suffer from the problem of mode collapse in training. Recently, many researchers have employed transformers (Vaswani et al. 2017) to achieve significant progress in generating high-quality images (Ding et al. 2021; Ramesh et al. 2021; Wu et al. 2022b; Yu et al. 2022a).

In text-to-image synthesis, the transformer-based methods exploit Vector Quantized Variational AutoEncoders (VQ-VAE) (van den Oord, Vinyals, and Kavukcuoglu 2017) to transform the given image into a low-dimensional image tokens. Then, the methods model the joint distribution between the image tokens and the language tokens to predict the image tokens in testing. In testing, the transformer-based methods sample visual tokens from the joint distribution step by step, which would bring much noise and suffer from error propagation. The current common practice is reranking multi-samples drawn from the transformer with a pre-trained multi-modal model. For example, given a caption, DALL-E (Ramesh et al. 2021) generates 512 images and re-ranks them to search for the best image, and Parti (Yu et al. 2022a) samples 16 images for searching. Besides, many sophisticated transformer-based methods rely on large-scale models. The parameters in DALL-E are up to 12 billion parameters, and CogView consists of 4 billion parameters. Thus, slow and inefficient inference may be the main bottleneck of the

*J. Cheng is the corresponding author.

tasks applied in real life.

The inference stage includes three phases: (a) sampling tokens of multi N_{ref} paths (a path represents the visual tokens of a image for simplifying) via the large transformer model; (b) transforming N_{ref} paths to N_{ref} images by using VQ-GAN; (c) re-ranking N_{ref} generated images to select the best image. In phase (c), the inferior images with number of $N_{\text{ref}} - 1$ will be dropped, and the corresponding computing load of (a) and (b) is nearly useless. Thus, to improve the inference efficiency and synthesizing quality, we propose a *reject decoding algorithm* to reduce the inefficient computing load and enlarge the searching space to improve the final results. As shown in Figure 1, "O5" and "O10" indicate the images generated by the original decoding via setting $N_{\text{ref}} = 5$ and $N_{\text{ref}} = 10$, respectively. The reject decoding algorithm generates the final result "N5", which is the same as"O10" and better than "O5", under a similar computing load of "O5". In phase (a), we exploit the vision models to guide the decoding to reject the lower-quality tokens as early as possible, reducing the unnecessary calculation. Thus, we can enlarge the initial searching paths as marked in large blue circles, which would improve the decoding quality. In Figure 1, the reject decoding algorithm outputs "N5" that is the same image as "O10", which has twice computing load. Besides, since the final paths of "N5" will be smaller than those of $O5$ in phase (a), some following operations of phase (b) and phase (c) can be eliminated, which further improves the efficiency.

To summarize, we propose a *reject decoding algorithm* to reduce the inefficient calculation and enlarge the initial searching paths for covering a larger searching space and improving the decoding quality, and the contributions are threefold:

- To improve the efficiency in decoding, we propose a reject decoding algorithm, where the language-vision models are employed to guide the decoding to reject the lower-quality paths as early as possible.
- To measure the alignment between the given textual description and the full or part of image tokens, we introduce tiny transformer-based multimodal language-vision models and train them with a contrastive loss.
- We conduct extensive experiments with a base model trained on the MS-COCO dataset and a large-scale model trained on large-scale datasets to verify the efficiency of the reject decoding algorithm and the effectiveness of the multimodal vision models.

Related Work

GAN-Based Text-to-Image generation

Reed *et al*. (Reed et al. 2016) proposed GANs to generate plausible images from text. Then, Stacked GANs *et al*. (Zhang et al. 2019; Zhang, Xie, and Yang 2018) are proposed to gradually synthesize images and improve the generating quality. Attentional models (Xu et al. 2018; Zhu et al. 2019; Cheng et al. 2020) are introduced to focus on different words when handing different parts of an image. Wu *et al.* (Wu et al. 2022a) exploited the attribute pairs to improve the controllability, and Qiao *et al.* (Qiao et al. 2019) proposed a MirrorGAN to improve semantic consistency. Tan *et al.* (Tan et al. 2021) and Yuan and Peng (Yuan and Peng 2020) proposed transferring methods to improve the association between the given text and the synthesized image.

For object-oriented generating, Hinz *et al*. (Hinz, Heinrich, and Wermter 2019; Hinz Heinrich) proposed object-level generators to synthesize the complex scenes. Sylvain *et al*. (Sylvain et al. 2020) exploited object-centric generators to fuse the object layout, and Li *et al*. (Li et al. 2019) introduced two-step object-driven GANs to exploit bounding boxes to improve the quality. Besides, many works (Li, Zhang, and Malik 2019; Pavllo, Lucchi, and Hofmann 2020; Sun and Wu 2019; Li et al. 2020) implicitly decomposed complex scenes to fuse the layouts.

Transformer-Based Text-to-Image Synthesis

Recent works employ Vector Quantized Variational AutoEncoders (VQ-VAE) (van den Oord, Vinyals, and Kavukcuoglu 2017) to compress the high-resolution dense image into low-dimensional discrete codes, and the decoder of VQ-VAE can recover the dense image from the discrete codes. Then, the transformer (Esser, Rombach, and Ommer 2021; Ramesh et al. 2021; Ding et al. 2021; Huang et al. 2021b; Wu et al. 2022b) models the prior of the discrete codes and predicts the codes in an auto-regressive manner, which greatly improves the synthesizing quality. Zhang *et al*. (Zhang et al. 2021b) proposed the new two-stage UFC-BERT that exploited the progressive non-autoregressive generation to improve the holistic consistency and support preserving operation. The UFC-BERT decoded B parallel paths similar to the beam search. Then, it dropped some low probability tokens of a path and re-predicted them in each iterative step. Zhang *et al*. (Zhang et al. 2021a) introduced ERNIE-ViLG to model image-text bidirectional generation in an autoregressive generating manner. Kim *et al*. (Kim et al. 2022) proposed the L-Verse with a feature-augmented variational autoencoder and bidirectional auto-regressive transformers for the image-text bidirectional generating. Huang *et al*. (Huang et al. 2021a) exploited a transformer to synthesize high-quality images conditioned on multiple captions. Esser *et al*. (Esser et al. 2021) proposed the ImageBART to synthesize images in a coarse-to-fine manner by using autoregressive models and the multinomial diffusion process. Yu *et al*. (Yu et al. 2022b) built the Parti to synthesize high-fidelity photorealistic images by using large language models and the large transformer model.

Diffuse-Based Text-to-Image Synthesis

Tang *et al*. (Tang et al. 2022) introduced vector quantized diffusion models with classifier-free guidance sampling and used a high-quality inference method. Ramesh *et al*. (Ramesh et al. 2022) built a two-stage model by generating the CLIP image embedding and synthesizing the corresponding images via diffusion models. Nichol *et al*. (Nichol et al. 2021) proposed the GLIDE to synthesize high-quality images via exploiting diffusion models with CLIP guidance and classifier-free guidance. Gu *et al*. (Gu et al. 2022) introduced the vector quantized diffusion model with the mask-

Algorithm 1: Original Decoding in Transformer

Input: The given caption T; The referent predicted size N_{ref}; The number of one image tokens L.
Output: A set of paths $\hat{c}$

1: $\hat{\mathcal{G}} \leftarrow \{\{\}_1, \{\}_2, \cdots, \{\}_{N_{\text{ref}}}\}$
2: **for** $j \in \{1, ..., L\}$ **do**
3:     **for** $\hat{g} \in \hat{\mathcal{G}}$ **do**
4:         $\hat{c}_j \leftarrow \texttt{Multinomial}(p_\Theta(\hat{c}_j|\hat{g}, T))$
5:         $\hat{g} \leftarrow \hat{g} \cup \{\hat{c}_j\}$
6:     **end for**
7: **end for**
8: Return the set of predicted tokens $\hat{\mathcal{G}}$

Algorithm 2: Reject Decoding in Transformer

Input: The given caption T; The initial predicted size N_b;The end predicted size N_e; The reject threshold $\{\sigma_1, \sigma_2, \cdots, \sigma_K\}$; The size of predicted group M.
Output: A set of paths $\hat{c}$

$\hat{\mathcal{G}} \leftarrow \{\{\}_1, \{\}_2, \cdots, \{\}_{N_b}\}$
for $i \in [1, K]$ **do**
    for $\hat{g} \in \hat{\mathcal{G}}$ **do**
        for $j \in [1, M)$ **do**
            $\hat{c}_j \leftarrow \texttt{Multinomial}(p_\Theta(\hat{c}_j|\hat{g}, T))$
            $\hat{g} \leftarrow \hat{g} \cup \{\hat{c}_j\}$
        end for ▷ predicted M tokens for each group
    end for
    $\hat{\mathcal{S}} \leftarrow \{\}$;
    for $\hat{g} \in \hat{\mathcal{G}}$ **do** ▷ the alignment between group and T
        $s_g \leftarrow \mathcal{M}_i(\hat{g}, T)$
        $\hat{\mathcal{S}} \leftarrow \hat{\mathcal{S}} \cup (\hat{g}, s_g)$
    end for
    $R_c \leftarrow \texttt{max}(\lceil|\hat{\mathcal{G}}| \times \sigma_i\rceil, N_e)$
    $\hat{\mathcal{G}} \leftarrow \texttt{TopK}(\hat{\mathcal{S}}, R_c)$ ▷ reject $|\hat{\mathcal{G}}| - R_c$ groups with lowest scores
end for
Return the set of predicted tokens $\hat{\mathcal{G}}$

and-replace diffusion strategy to generate the tokens conditioned on a caption, then decoded the tokens into the synthesized images. Saharia *et al*. (Saharia et al. 2022) proposed the Imagen by exploiting large transformer language models to better understand captions and synthesize high-quality images, which may complement to Parti and generate similar photorealistic images.

Difference to Existing Works

Transformer-based methods need to generate multi-images in search of the best one to alleviate the error propagation and the exposure bias. The computing load of excluding paths may be a waste, and the reject decoding algorithm tries to skip those paths as earlier as possible. Current works, like UFC-BERT (Zhang et al. 2021b), would exploit beam search decoding under some useful constraints (Susanto, Chollampatt, and Tan 2020) to select the k-best paths at each step. However, for image synthesis via image tokens, a smaller part of tokens may be more unreliable to select the k-best candidates as in beam search decoding. Thus, we propose the reject decoding algorithm to exclude some lowest quality partial paths instead of selecting the k-best paths to improve the decoding effectiveness and efficiency.

Methodology

In this section, we propose a new decoding algorithm to improve decoding efficiency and synthesizing quality. First, we present the *reject decoding algorithm* to eliminate the useless partial paths as early as possible. After that, we describe multi-modal language vision models to measure the alignment between the given textual description and the partial paths for finding the useless paths.

Given an image I and the corresponding caption T, we define $c = \gamma(I)$ as the corresponding residual quantization discrete tokens, where γ is the encoder of the RQ-VAE (Lee et al. 2022). Let $\hat{\mathcal{C}}$ be predicted image tokens from caption T, the transformer model predicts the current tokens $\hat{c}_j \in \hat{\mathcal{C}}$, $j \in \{1, ..., |\hat{\mathcal{C}}|\}$[1], based on the previous image tokens $\hat{c}_{1:j-1} \subset \hat{c}$ as follows,

$$\hat{c}_j = \texttt{Multinomial}(p_\Theta(\hat{c}_j|\hat{c}_{1:j-1}, T)), \tag{1}$$

where Θ is the parameter of the transformer. The function $\texttt{Multinomial}(\cdot)$ samples a token from the multinomial probability distribution p_Θ, which can be implemented as the truncation sampling (Gu et al. 2022). To improve the synthesizing quality and semantic consistency, the transformer-based model normally synthesizes multi images and re-ranks them to select the best one. Thus, as shown in Algorithm 1, the transformer model auto-regressively generates N_{ref} paths, where each path consists of one image tokens $\hat{\mathcal{C}}$ that can be transformed into the corresponding image $\hat{I} = \phi(\hat{\mathcal{C}})$ by using the decoder ϕ of the RQ-VAE.

Since only one of N_{ref} images would be selected, the original algorithm may be inefficient. Thus, to improve the decoding efficiency and synthesizing quality, we introduce a reject decoding method as depicted in the following parts.

Reject Decoding in Transformer

First, we analyze the computing load of the reject decoding compared with the original decoding and show that the reject decoding algorithm can touch more candidates at the beginning. Besides, we investigate the probability of preserving the ground-truth to show that the rejecting elements would unlikely contain the ground-truth. In all, we provide an algorithm to configure the reject threshold automatically.

Computing Load To improve the decoding efficiency and quality, we can evaluate the partial tokens $\hat{g}$ in line 5 of Algorithm 1. However, considering the computing load, we split full L tokens into K groups (each group includes $M = L/K$ tokens) as shown in Algorithm 2, and evaluate the partial tokens $\hat{g}$ consisting of several groups, which is similar to the situations that the sentence can be split into

[1] $|\hat{\mathcal{C}}|$ denotes the number of tokens in $\hat{\mathcal{C}}$.

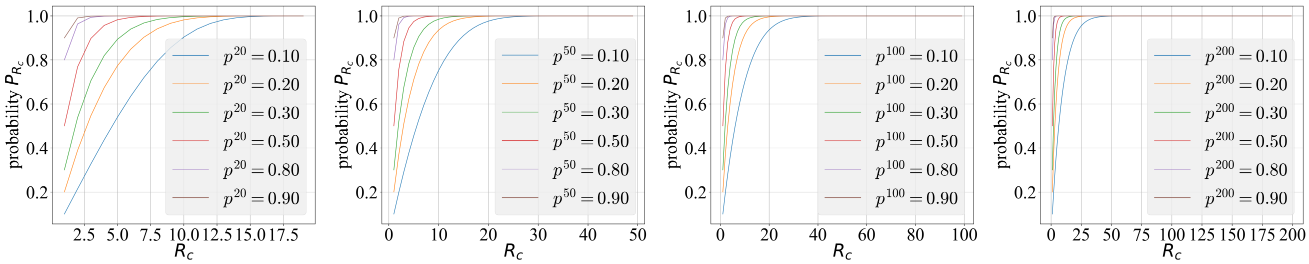

Figure 2: Diagrams of reserving count R_c and the probability of reserving ground-truth under different p^x of scorers $\mathcal{M}_i$: the figures indicate that the ground-truth would unlikely be dropped when R_c is large.

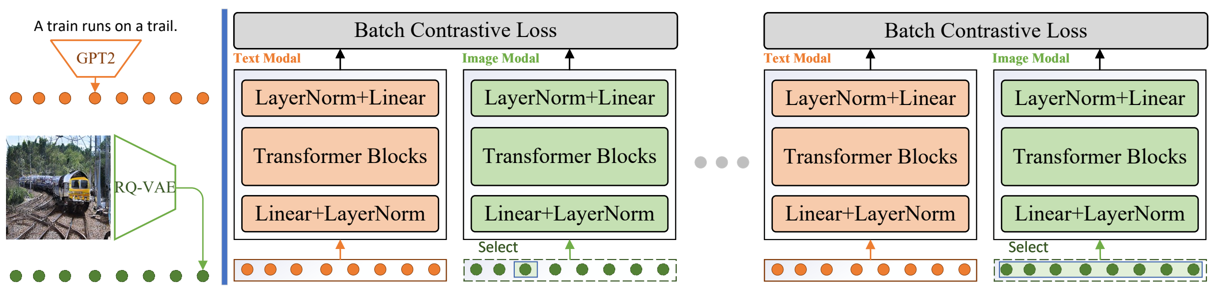

Figure 3: Multimodal Vision Models: they are trained with the random subset of visual tokens and the caption and contrastive loss.

words.

In Algorithm 1, the total executing times in computing Eq. (1) is $C_{\text{org}} = N_{\text{ref}} L = N_{\text{ref}} KM$, where N_{ref} is the number of generated images corresponding to the caption T. In Algorithm 2, the total executing times in computing Eq. (1) is as follows,

$$C_{\text{reject}} = M \sum_{i=1}^{K} N_i, \tag{2}$$

where $N_1 = N_b$, and $N_{i+1} = \max(\lceil N_i \sigma_i \rceil, N_e)$. N_b, N_e and σ_i are the initial size, end size of tokens, and the reject threshold at i-th iteration, respectively. Specifically, for simplicity, when $\sigma_i = \xi$ is constant, and $N_{i+1} = N_i \sigma_i$, we can see that

$$C_{\text{reject}} = M \sum_{i=1}^{K} N_b \xi^{i-1} = M N_b \frac{(1-\xi^K)}{(1-\xi)}. \tag{3}$$

In order to ensure that the executing times are not increasing, we want to have $C_{\text{reject}} \leq C_{\text{org}} = N_{\text{ref}} KM$, then we need to require $N_b \leq N_{\text{ref}} K \frac{1-\xi}{1-\xi^K}$. Define $f_{\text{scale}}(K, \xi) = K \frac{1-\xi}{1-\xi^K}$. If $f_{\text{scale}}(K, \xi) > 1$, and $N_b > N_{\text{ref}}$ is feasible and we can search across more candidates at the beginning. In particualr, we can obtain the lower bound of f_{scale} as,

$$f_{\text{scale}}(K, \xi) = \frac{K(1-\xi)}{1-\xi^K} \geq K(1-\xi), \tag{4}$$

where the last formulate is driven by $1 - \xi^K \leq 1$. Thus, the scale factor f_{scale} is proportion to K. Besides, the partial derivative of f_{scale} with respect to ξ is,

$$\frac{\partial f_{\text{scale}}(K, \xi)}{\partial \xi} = -\frac{K}{(1-\xi^K)^2}(1+(K-1)\xi^K - K\xi^{(K-1)}), \tag{5}$$

where the partial derivative of $1 + (K-1)\xi^K - K\xi^{(K-1)}$ with respect to ξ is $(K-1)K\xi^{K-2}(\xi - 1) \leq 0$, i.e., $1 + (K-1)\xi^K - K\xi^{(K-1)} \geq 1 + (K-1)1^K - K1^{(K-1)} = 0$. Thus, we can obtain that

$$\frac{\partial f_{scale}(K, \xi)}{\partial \xi} \leq 0. \tag{6}$$

Thus, the scale factor f_{scale} will be a decreasing function with respect to ξ. In Eq. (4), given ξ, and set $K \geq 1/(1-\xi)$, $C_{\text{reject}} \leq C_{\text{org}}$, $N_b \geq N_{\text{ref}}$, our decoding method can search across *more candidates in the beginning* at the same computing load, which means the result would be better than the original Algorithm 1, to alleviate *the error propagation for incremental decoding*.

Preserving the best candidate by $\mathcal{M}_i$ Given a set of paths $\hat{\mathcal{G}}$, we presume that $\hat{\mathcal{G}}_*$ is the ground-truth element of $\hat{\mathcal{G}}$, and the scorer $\mathcal{M}_i$ can classify $\hat{\mathcal{G}}_*$ from other elements with a probability p, i.e., $\mathcal{M}_i(\hat{\mathcal{G}}_*, T) \geq \mathcal{M}_i(\hat{\mathcal{G}}_k, T), k \in \{1, ..., N_{\hat{\mathcal{G}}}\}$, and $N_{\hat{\mathcal{G}}} = |\hat{\mathcal{G}}|$. Thus, the probability of

Algorithm 3: Searching Reject Threshold

Input: The referent counter N_{ref}; the begin and end predicted sizes N_b and N_e, respectively; the number of groups K; the preferring reject probability p_r
Output: the set of using count $\{N_i\}_{i=1}^{K}$

```
N_i ← N_e,  ∀i ∈ {2, ..., K};
N_1 ← N_b
for i ∈ {2, ..., K} do
    C_res ← max(N_ref K − Σ_j N_j, 0)
    N_i ← min(⌊N_{i−1} p_r⌋, C_res + N_i)
end for
if Σ_j N_j ≠ N_ref K then
    Return {N_i}_{i=1}^K, Fail
end if
Return {N_i}_{i=1}^K, Successful
```

$\mathcal{M}_i(\hat{\mathcal{G}}_*, T) \geq \mathcal{M}_i(\hat{\mathcal{G}}_k, T), \forall \hat{\mathcal{G}}_k \in \hat{\mathcal{G}}$ is $p^{N_{\hat{\mathcal{G}}}-1}$. In line 15 of Algorithm 2, we can split $\hat{\mathcal{G}}$ into two sets $\hat{\mathcal{G}}^0 = \mathtt{TopK}(\hat{\mathcal{S}}, R_c)$ and $\hat{\mathcal{G}}^1 = \hat{\mathcal{G}}/\hat{\mathcal{G}}^0$, where R_c is size of preserving the paths and function $\mathtt{TopK}(\hat{\mathcal{S}}, R_c)$ is to reject the $|\hat{\mathcal{G}}| - R_c$ groups with the lowest scores, $\hat{\mathcal{S}}$ is score set of $\hat{\mathcal{G}}$. Thus, the probability of preserving the ground truth $\hat{\mathcal{G}}_*$ in $\hat{\mathcal{G}}^0 \subset \hat{\mathcal{G}}$ is P_{R_c}, which is defined as follows,

$$P_{\{R_c=1\}} = p^{N_{\hat{\mathcal{G}}}-1}, \tag{7}$$

$$P_{\{R_c=j+1\}} = (1 - P_{\{R_c=j\}}) p^{N_{\hat{\mathcal{G}}}-j-1} + P_{\{R_c=j\}}, \tag{8}$$

where the first term of Eq. (8) is the probability of $P(\hat{\mathcal{G}}^0_{R_c} = \hat{\mathcal{G}}_*)$. The probability $P_{R_c} \to 1$ when $R_c \to N_{\hat{\mathcal{G}}}$, and $P_{\{R_c=N_{\hat{\mathcal{G}}}\}} = 1$. For examples, in Figure 2, the plots indicate that a larger $p^{N_{\hat{\mathcal{G}}}-1}$ would make P_{R_c} increasing faster and can exclude more trailing elements. When the partial path is short and $p^{N_{\hat{\mathcal{G}}}-1}$ is small, we could reject a few elements at a slight cost. In short, the ground-truth would be unlike in the trailing elements of sorted $\hat{\mathcal{G}}$, which could be dropped with little cost.

Configure of Reject Counters In Algorithm 3, given the reference counter N_{ref} for Algorithm 1,we can search the reject thresholds for Algorithm 2 under the same computing loads of the transformer. Specially, given the initial and end predicted counter $N_b > N_{\text{ref}}$ and $N_e < N_{\text{ref}}$, we can enumerate $p_r \in [0, 1]$ and find out the smallest p_r which lets Algorithm 3 return "Successful". Note that the reject threshold can be set as N_i/N_{i-1}.

Multimodal Language-Vision Models

In Algorithem 2, we utilize tiny language-vision models to filter out the low-quality partial paths. First, we exploit the transformer-based structure to get the embeddings of a partial path and the given caption in a common space. Then, we train the tiny models with a contrastive loss.

Embedding Caption and Tokens As shown in the Figure 3, given a caption T, we exploit GPT2, noted as $\mathcal{E}_{\text{GPT2}}$, to get the embeddings $\omega = \mathcal{E}_{\text{GPT2}}(T) \in \mathbb{R}^{|T| \times N_\omega}$, where N_ω is the dimension of word embedding and $|T|$ is the size of caption T. Thus, the representation of T is calculated as follows,

$$f_{\mathcal{M}_{\text{T}}}(\omega) = L_{\text{mean},1}(L_{\text{L+N}}(L_{\text{Trans}}(L_{\text{N+L}}(\omega)))), \tag{9}$$

where $L_{\text{N+L}}$ denotes a linear layer to translate the embedding ω into the hidden features, followed by a layer normalization; L_{Trans} includes several transformer blocks, consisting of a multi-head self-attention layer, layer normalization, and a multi-layer perceptron; $L_{\text{L+N}}$ indicates a layer normalization followed by a linear layer; $L_{\text{mean},1}$ computes the mean across the second dimension, namely the mean of embeddings words.

Given residual quantization tokens $\mathcal{C} \in \mathbb{R}^{L \times 4}$ of an image, we employ the decoder ϕ of the RQ-VAE to get the embedding $\nu = \mathcal{E}_\phi(\mathcal{C}) \in \mathbb{R}^{L \times 4 \times N_\nu}$, where N_ν is the dimension of the visual embedding i.e.,

$$f_{\mathcal{M}_{\text{c}}}(\nu) = L_{\text{mean},1}(L_{\text{L+N}}(L_{\text{Trans}}(L_{\text{N+L}}(L_{\text{mean},1}(\nu))))). \tag{10}$$

Training with Contrastive Loss Given the k-th caption T and residual quantization tokens $\mathcal{C}$ in a batch, their embeddings are,

$$\hat{\omega}_k = \mathtt{Norm}(f_{\mathcal{M}_{\text{T}}}(\mathcal{E}_{\text{GPT2}}(T))), \tag{11}$$

$$\hat{\nu}_k = \mathtt{Norm}(f_{\mathcal{M}_{\text{c}}}(\mathcal{E}_\phi(\mathcal{C}))), \tag{12}$$

where the function $\mathtt{Norm}$ is L_2-normalization, and the training loss of a batch L_{contrast} is defined as,

$$-\sum_i \left\{ \frac{\exp(\hat{\omega}_i \cdot \hat{\nu}_i)}{\sum_{k \neq i} \exp(\hat{\omega}_i \cdot \hat{\nu}_k)} + \frac{\exp(\hat{\omega}_i \cdot \hat{\nu}_i)}{\sum_{k \neq i} \exp(\hat{\omega}_k \cdot \hat{\nu}_i)} \right\}. \tag{13}$$

Besides, as shown in the Figure 3, we train and evaluate the alignment between part image tokens and the given caption. For model $\mathcal{M}_i = \{f_{\mathcal{M}_{\text{T}}}, f_{\mathcal{M}_{\text{c}}}\}$, we sample a subset of tokens $\mathcal{C}' \subset \mathcal{C}$ and exploit Eq. (13) to compute the contrastive loss of a given batch. In Algorithm 2, we choose the group size $M = 8$, and the total size of tokens is 64. Thus, we construct 8 similarity models as $\{\mathcal{M}_i\}_{i=1}^{8}$.

Experiments

We conduct experiments by using the RQ-Transformer (Lee et al. 2022) as baselines and train the RQ-Transformer on the MS-COCO dataset (Lin et al. 2014) as the normal model denoted by the superscript "coco". To verify the experiments on large-scale datasets, we exploit the large-scale pre-trained RQ-Transformer with 3.9B parameters[2] denoted by the superscript "pre", which is trained by CC-3M[3], CC-12M[4], and YFCC-subset[5].

Evaluating Metrics

a) Inception score (IS): IS (Salimans et al. 2016) is an automatic metric and popular to evaluate the quality, which

[2] github.com/kakaobrain/rq-vae-transformer
[3] github.com/google-research-datasets/conceptual-captions
[4] github.com/google-research-datasets/conceptual-12m
[5] github.com/openai/CLIP/blob/main/data/yfcc100m.md

Methods	IS↑	FID↓	RP_{cnn} ↑	RP_{trans} ↑	Consuming Time(s)↓
DALL-E	17.90	27.50	N/A	N/A	N/A
CogView	18.20	27.10	N/A	N/A	N/A
RQ-Transformer$_5^{\text{pre}}$	24.55 ± 0.51	16.5989	63.75 ± 0.90	56.10 ± 0.62	17.13
Our$_5^{\text{pre}}$	$\mathbf{25.39 \pm 0.42}$	**14.5853**	$\mathbf{69.27 \pm 0.92}$	$\mathbf{67.37 \pm 0.91}$	**16.68**
RQ-Transformer$_{10}^{\text{pre}}$	24.37 ± 0.51	16.4212	66.61 ± 0.42	58.45 ± 0.65	33.54
Our$_{10}^{\text{pre}}$	$\mathbf{25.64 \pm 0.65}$	**14.3066**	$\mathbf{70.91 \pm 0.59}$	$\mathbf{69.42 \pm 0.73}$	**29.22**
RQ-Transformer$_5^{\text{coco}}$	26.36 ± 0.28	8.3623	68.42 ± 0.72	69.38 ± 0.98	6.93
Our$_5^{\text{coco}}$	$\mathbf{27.19 \pm 0.56}$	**8.2824**	$\mathbf{71.98 \pm 0.66}$	$\mathbf{76.64 \pm 0.99}$	**6.76**
RQ-Transformer$_{10}^{\text{coco}}$	27.27 ± 0.43	8.2035	70.37 ± 0.63	71.09 ± 0.77	10.74
Our$_{10}^{\text{coco}}$	$\mathbf{27.98 \pm 0.63}$	**8.1090**	$\mathbf{74.45 \pm 0.63}$	$\mathbf{77.64 \pm 0.43}$	**10.32**

Table 1: Inception Score (IS), Fréchet Inception Distance (FID), R-precision(RP_{cnn} and RP_{trans}), and Consuming Time

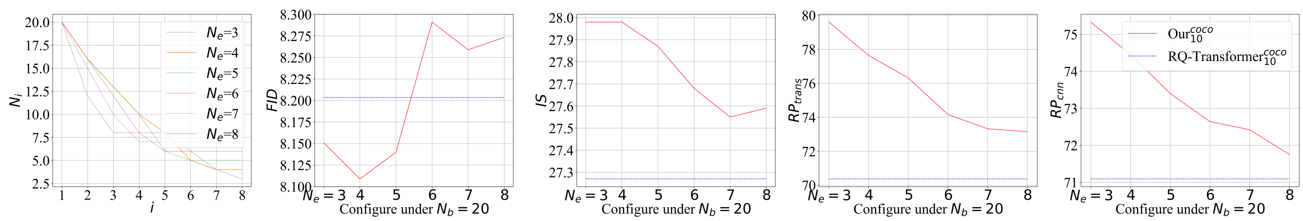

Figure 4: Diagrams of FID, IS, RP_{cnn}, and RP_{trans} for our models by exploiting different N_e under similar computing loads.

favors meaningful and diverse images. Followed the works in (Zhang et al. 2019; Xu et al. 2018; Wu et al. 2022b; Tao et al. 2022; Li et al. 2022), although it has some flaws (Barratt and Sharma 2018), we report the metric to compare the quality of synthesized images.

b) Fréchet Inception Distance (FID): FID measures the Fréchet distance between the features of 30K generated images and real images. A lower FID indicates that the model generates higher-quality images. Thus, we report the FID to compare those models.

c) R-precision: To measure the semantic consistency between the given caption and the synthesized image, we employ R-precision (Xu et al. 2018) to evaluate the alignment, denoted as RP_{cnn}. Since transformer models could extract high-quality features, similar to works (Park et al. 2021), we utilize the language-vision transformer, like $\mathcal{M}_i$, to extract base features to compute the R-precision, denoted as RP_{trans}.

Quantitative Comparison

In Table 1, the subscript "5" and "10" indicate the referent computing load for $N_{\text{ref}} = 5$ and $N_{\text{ref}} = 10$ in the original decoding, respectively. For the large-scale pre-trained models, compared with RQ-Transformer$_{10}^{\text{pre}}$ decoding under $N_{\text{ref}} = 10$ and trained by CC-3M, CC-12M, and YFCC-subset, the IS of Our$_{10}^{\text{pre}}$ increases 1.27. the FID of Our$_{10}^{\text{pre}}$ largely decreases 2.11. The RP_{cnn} and RP_{trans} increases by **4.30%** and **10.97%**, respectively. The consuming time, including the whole process from feeding the text embedding to returning the final image, of Our$_{10}^{\text{pre}}$ reduces by about 4.32 seconds. The consuming times are evaluated under the same batch sizes on the same device and may be further improved with some engineering optimization. Compared with DALL-E (trained with CC-3M and YFCC-subset) and CogView (trained with WudaoCorpora), the IS of Our$_{10}^{\text{pre}}$ increases at least 17.44, and the FID decreases at least 12.79. For the models trained with MS-COCO, compared with RQ-Transformer$_{10}^{\text{coco}}$, the IS of Our$_{10}^{\text{coco}}$ increases by 0.71. the FID of Our$_{10}^{\text{coco}}$ decreases 0.09. The RP_{cnn} and RP_{trans} increases **4.08%** and **6.55%**, respectively. The consuming time of Our$_{10}^{\text{coco}}$ reduces by about 0.42 seconds. The results indicate that the reject decoding could synthesize better images while maintaining a similar computing load.

a) IS, FID, RP_{cnn}, and RP_{trans} under Different N_e:
In Figure 4, the results demonstrate the influence of N_e with $N_b = 20$: the first figure shows that the number N_i will be dropping faster with larger N_e. With increasing N_e, the FID will be decreasing firstly, which indicates that the re-ranking phase is important, and it is beneficial to provide several images for the re-ranking. However, when increasing N_e further, the decoding may be hard to cover the high-quality path, because the number of inner paths N_i is dropping faster. The results indicate that the reject threshold is important, and the N_e prefers a relatively small number. However, IS, RP_{cnn}, and RP_{trans} are better than those of the corresponding baseline.

b) IS, FID, RP_{cnn}, and RP_{trans} under Different Language-Vision Models :
In Figure 5, we exploit the multimodal vision models, consisted of 8 layers with 4 heads, under different training epochs to evaluate the performance. In the first figure, by using more tokens "#x", $x \in \{8, 16, \ldots, 64\}$, RP_{trans} will increase, which shows that it is more reliable to measure the similarity between the captions and the larger part of tokens. Besides, the first figure shows that RP_{trans} will increase with more training epochs. By using the vision models with higher RP_{trans}, the FID of our model will decrease,

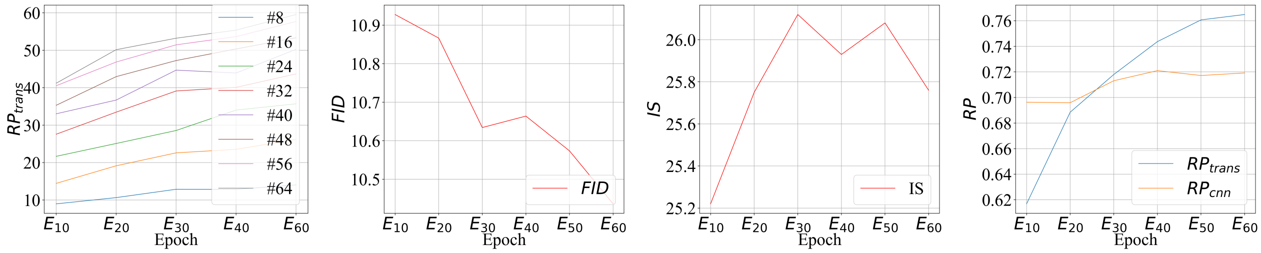

Figure 5: Diagrams of FID, IS, RP_{cnn} and RP_{trans} for our models by exploiting different language-vision models.

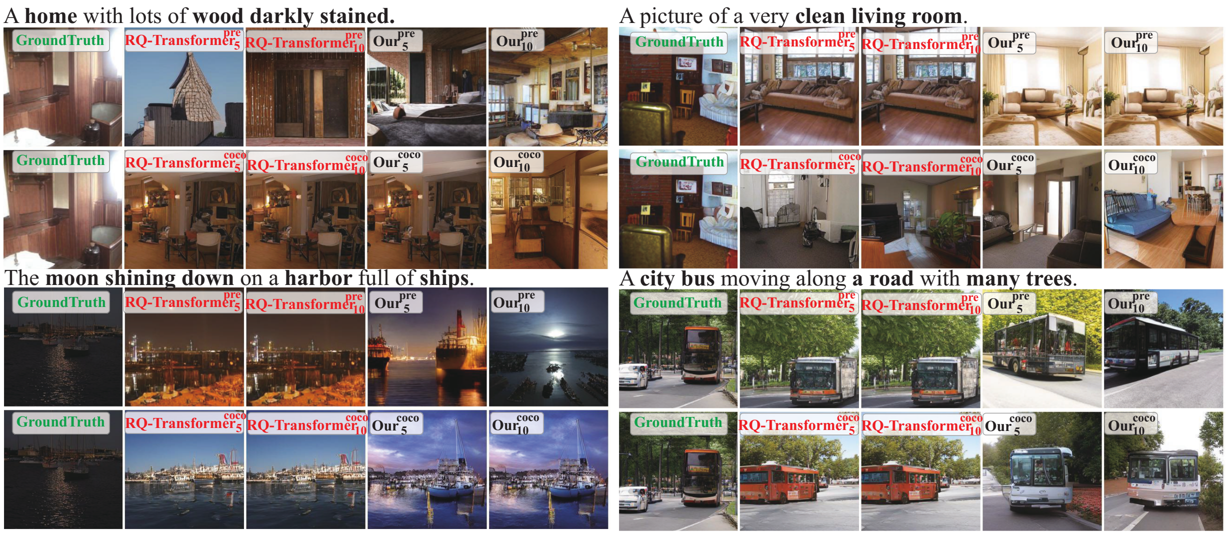

Figure 6: Synthesized examples: the caption is above the corresponding image, and the prominent features are marked as bold.

and IS is on an ascending trend. By exploiting different vision models, the last figure shows that RP_{cnn} and RP_{trans} will increase with more training epochs of vision models (with higher RP_{trans}), which indicates that language-vision models could provide effective guidance to retrieve images with higher quality and to improve semantic similarity. Furthermore, when we exploit the similarity computed by the original transformer, $RP_{\text{cnn}} = 68.06$, $RP_{\text{trans}} = 67.89$, $FID = 8.63$, and $IS = 26.86$, the scores are worse than those with language-vision models, which shows the importance of the models.

Qualitative Comparison

In Figure 6, the results show that images of $\text{Our}_5^{\text{pre}}$ and $\text{Our}_5^{\text{coco}}$ are better than those of $\text{RQ-Transformer}_5^{\text{pre}}$ and $\text{RQ-Transformer}_5^{\text{coco}}$, respectively. For the referent number $N_{\text{ref}} = 10$, images of $\text{Our}_{10}^{\text{pre}}$ and $\text{Our}_{10}^{\text{coco}}$ are also better than those of $\text{RQ-Transformer}_{10}^{\text{pre}}$ and $\text{RQ-Transformer}_{10}^{\text{coco}}$, respectively. And the images of $N_{\text{ref}} = 10$ would be better than those of $N_{\text{ref}} = 5$. The results indicate that the reject decoding could improve the synthesizing quality and generate realistic images. For example, given "a home with lots of wood darkly stained", in the left top part, the images of $\text{Our}_5^{\text{pre}}$ and $\text{Our}_{10}^{\text{pre}}$ include more vivid visual details of "home with lots of wood" than those of $\text{RQ-Transformer}_5^{\text{pre}}$ and $\text{RQ-Transformer}_{10}^{\text{pre}}$. Besides, $\text{Our}_5^{\text{coco}}$ retrieves the same image of "home" as that of $\text{RQ-Transformer}_{10}^{\text{coco}}$, and $\text{Our}_{10}^{\text{coco}}$ generates the better image than that of $\text{Our}_5^{\text{coco}}$. The results show that our reject decoding could generate high-quality images.

a) Synthesizing with Different N_e:

In Figure 7, the up two rows are for large-scale pre-trained models. The first row shows the 10 generated images for re-ranking. $\text{RQ-Transformer}_5^{\text{pre}}$ will re-rank the first 5 images to get the best image, and $\text{Our}_5^{\text{pre}}$ would search the sub-paths of the total 10 generated images when $N_b = 10$. $[1, 5] \rightarrow x$ denotes that the x^{th} image is the best image selected from the first 5 images as in the original decoding, and $\{3, 7\} \rightarrow x$ denotes that the best image x is selected from the set $\{3, 7\}$. In Figure 7, the quality of 10 generated images varies widely. Thus, re-ranking is an essential phase. The original model, $\text{RQ-Transformer}_5^{\text{pre}}$ and $\text{RQ-Transformer}_5^{\text{coco}}$, can only touch the first 5 images $[1, 5]$, and the remaining $[6, 10]$ is unreachable. However, our reject decoding can touch the full 10 paths. For example, given "large black and white panda bear walking around in an enclosure", $\text{RQ-Transformer}_5^{\text{pre}}$ selects the 4^{th} image from the first 5 images. When $N_e = 1$, $\text{Our}_5^{\text{pre}}$ selects the 9^{th} image, which is better than the 4^{th} image. When $N_e = 2$, $\text{Our}_5^{\text{pre}}$ generates the 9^{th} and 3^{th} images, $\{9, 3\}$, then the 9^{th} image is selected in the re-ranking phase. Thus, Figure 7 shows that the reject decoding can expand the searching space so as to increase the possibilities

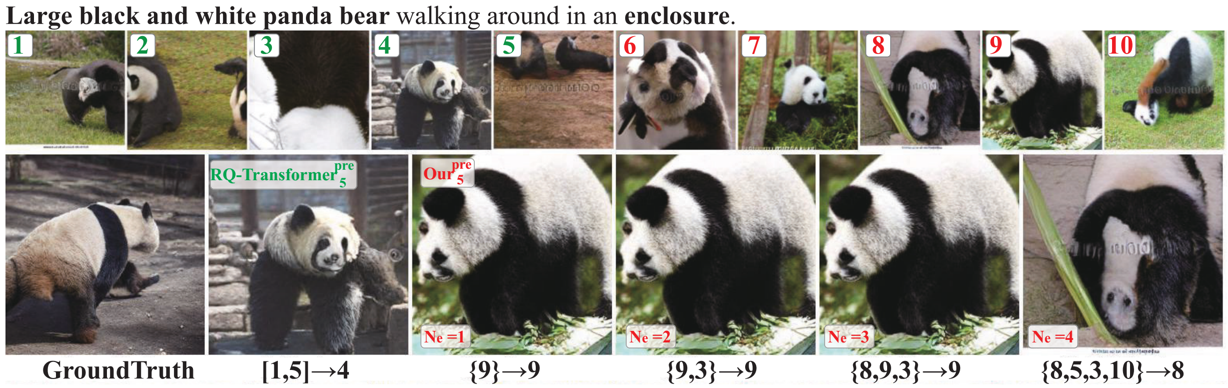

Figure 7: Synthesizing examples with different N_e.

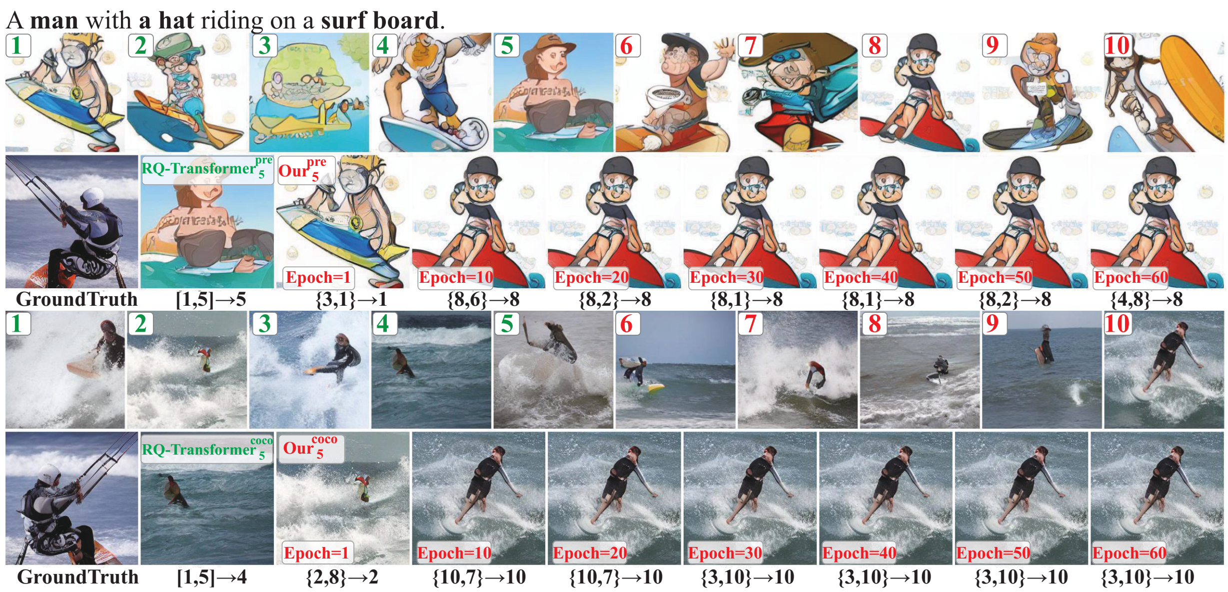

Figure 8: Synthesizing examples with different language-vision models.

of reaching the ground-truth.

b) Synthesizing with Different Language-Vision Models:

In Figure 8, an example consists of the results of large-scale pre-trained models at the up part and the results of the models trained with the MS-COCO dataset at the low part. The second row of an example includes images generated with multimodal vision models under different training epochs. Given "A man with a hat riding on a surf board", when Epoch=1, Our$_5^{pre}$ generates the 1^{th} and 3^{th} images. When Epoch > 10, Our$_5^{pre}$ selects the 8^{th} image as the best image, which is the best image with the vivid visual features "A man with a hat" and "surf board" among the 10 raw images. When Epoch=1, Our$_5^{coco}$ generates the 2^{th} and 8^{th} images. When Epoch > 10, Our$_5^{coco}$ selects the 10^{th} image as the best image, which is the best image among the 10 raw images. The results indicate that the better multimodal vision model would yield more vivid and higher-quality images.

Limitation and Discussion

Akin to the language model in machine translation, the multimodal vision models are important to guide the decoding process to search the large selecting space. Here, we only exploit the MS-COCO dataset to train the language-vision models, and a large-scale dataset would be beneficial to train the language-vision models and improve the final results. Similar to CLIP, a sophisticated language-vision method would improve the results. Besides, vision-only models may also guide the decoding to yield high-quality results, which will be a focus of future works.

Conclusion

We propose a reject decoding algorithm with tiny multimodal models to improve the decoding effectiveness and efficiency, which would skip the useless paths as early as possible and enlarge the searching space with little cost. We exploit the transformer-based model to build the tiny multimodal models. Then, we train the tiny models with the contrastive loss to evaluate the similarity between the textual description and the part of tokens for rejecting. The experiments show that the reject decoding could synthesize better images under a similar computing load and improve effectiveness and efficiency of decoding.

Acknowledgments

This work was supported in part by the Major Science and Technology Innovation 2030 "New Generation Artificial Intelligence" key project (No. 2021ZD0111700), Australian Research Council Project (DP-180103424), National Natural Science Foundation of China (U21A20487, 62206268), Guangdong Technology Project (No. 2016B010125003), Shenzhen Technology Project (JCYJ20220818101206014, KCXFZ20201221173411032, Y795001001, JCYJ20200109113416531), CAS Key Technology Talent Program.

References

Barratt, S.; and Sharma, R. 2018. A Note on the Inception Score. *arXiv preprint arXiv:1801.01973*.

Cheng, J.; Wu, F.; Tian, Y.; Wang, L.; and Tao, D. 2020. RiFeGAN: Rich Feature Generation for Text-to-Image Synthesis From Prior Knowledge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 10908–10917.

Ding, M.; Yang, Z.; Hong, W.; Zheng, W.; Zhou, C.; Yin, D.; Lin, J.; Zou, X.; Shao, Z.; Yang, H.; and Tang, J. 2021. Cogview: Mastering text-to-image generation via transformers. In *Proceedings of Advances in Neural Information Processing Systems, NeurIPS*, volume 34.

Esser, P.; Rombach, R.; Blattmann, A.; and Ommer, B. 2021. Imagebart: Bidirectional context with multinomial diffusion for autoregressive image synthesis. *Conference on Neural Information Processing Systems, NeurIPS*, 34: 3518–3532.

Esser, P.; Rombach, R.; and Ommer, B. 2021. Taming Transformers for High-Resolution Image Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 12873–12883.

Gu, S.; Chen, D.; Bao, J.; Wen, F.; Zhang, B.; Chen, D.; Yuan, L.; and Guo, B. 2022. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 10696–10706.

Hinz, T.; Heinrich, S.; and $S.$,Wermter. 2022. Semantic Object Accuracy for Generative Text-to-Image Synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3): 1552–1565.

Hinz, T.; Heinrich, S.; and Wermter, S. 2019. Generating Multiple Objects at Spatially Distinct Locations. In *Proceedings of International Conference on Learning Representations, ICLR*.

Huang, Y.; Liu, B.; Fu, J.; and Lu, Y. 2021a. A Picture is Worth a Thousand Words: A Unified System for Diverse Captions and Rich Images Generation. In *Proceedings of the 29th ACM International Conference on Multimedia, ACM MM*, 2792–2794.

Huang, Y.; Xue, H.; Liu, B.; and Lu, Y. 2021b. Unifying multimodal transformer for bi-directional image and text generation. In *Proceedings of the 29th ACM International Conference on Multimedia, ACM MM*, 1138–1147.

Kim, T.; Song, G.; Lee, S.; Kim, S.; Seo, Y.; Lee, S.; Kim, S. H.; Lee, H.; and Bae, K. 2022. L-Verse: Bidirectional Generation Between Image and Text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 16526–16536.

Lee, D.; Kim, C.; Kim, S.; Cho, M.; and Han, W.-S. 2022. Autoregressive Image Generation using Residual Quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 11523–11532.

Li, K.; Zhang, T.; and Malik, J. 2019. Diverse image synthesis from semantic layouts via conditional IMLE. In *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, 4220–4229.

Li, W.; Zhang, P.; Zhang, L.; Huang, Q.; He, X.; Lyu, S.; and Gao, J. 2019. Object-driven Text-to-Image Synthesis via Adversarial Training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 12174–12182.

Li, Y.; Cheng, Y.; Gan, Z.; Yu, L.; Wang, L.; and Liu, J. 2020. BachGAN: High-Resolution Image Synthesis from Salient Object Layout. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 8365–8374.

Li, Z.; Min, M. R.; Li, K.; and Xu, C. 2022. StyleT2I: Toward Compositional and High-Fidelity Text-to-Image Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 18197–18207.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *Proceedings of European Conference on Computer Vision, ECCV*, 740–755.

Nichol, A.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; McGrew, B.; Sutskever, I.; and Chen, M. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*.

Park, D. H.; Azadi, S.; Liu, X.; Darrell, T.; and Rohrbach, A. 2021. Benchmark for compositional text-to-image synthesis. In *Proceedings of Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Pavllo, D.; Lucchi, A.; and Hofmann, T. 2020. Controlling Style and Semantics in Weakly-Supervised Image Generation. In *Proceedings of European Conference on Computer Vision, ECCV*, 482–499.

Qiao, T.; Zhang, J.; Xu, D.; and Tao, D. 2019. MirrorGAN: Learning Text-to-image Generation by Redescription. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 1505–1514.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-Shot Text-to-Image Generation. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, volume 139, 8821–8831.

Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Schiele, B.; and Lee, H. 2016. Generative Adversarial Text to Image Synthesis. In *Proceedings of International Conference on Machine Learning, ICML*, 1681–1690.

Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E.; Ghasemipour, S. K. S.; Ayan, B. K.; Mahdavi, S. S.; Lopes, R. G.; et al. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *arXiv preprint arXiv:2205.11487*.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training gans. In *Proceedings of Advances in Neural Information Processing Systems, NeurIPS*, 2234–2242.

Sun, W.; and Wu, T. 2019. Image synthesis from reconfigurable layout and style. In *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, 10531–10540.

Susanto, R. H.; Chollampatt, S.; and Tan, L. 2020. Lexically constrained neural machine translation with levenshtein transformer. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics, ACL*, 3536–3543.

Sylvain, T.; Zhang, P.; Bengio, Y.; Hjelm, R. D.; and Sharma, S. 2020. Object-Centric Image Generation from Layouts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, AI for Content Creation Workshop*.

Tan, H.; Liu, X.; Liu, M.; Yin, B.; and Li, X. 2021. KT-GAN: Knowledge-Transfer Generative Adversarial Network for Text-to-Image Synthesis. *IEEE Transactions on Image Processing*, 30: 1275–1290.

Tang, Z.; Gu, S.; Bao, J.; Chen, D.; and Wen, F. 2022. Improved Vector Quantized Diffusion Models. *arXiv preprint arXiv:2205.16007*.

Tao, M.; Tang, H.; Wu, F.; Jing, X.-Y.; Bao, B.-K.; and Xu, C. 2022. DF-GAN: A Simple and Effective Baseline for Text-to-Image Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 16515–16525.

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

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems, NeurIPS*, 5998–6008.

Wu, F.; Cheng, J.; Wang, X.; Wang, L.; and Tao, D. 2022a. Image Hallucination From Attribute Pairs. *IEEE Transactions on Cybernetics*, 52(1): 568–581.

Wu, F.; Liu, L.; Hao, F.; He, F.; and Cheng, J. 2022b. Text-to-Image Synthesis Based on Object-Guided Joint-Decoding Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 18113–18122.

Xu, T.; Zhang, P.; Huang, Q.; Zhang, H.; Gan, Z.; Huang, X.; and He, X. 2018. AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 1316–1324.

Yu, J.; Xu, Y.; Koh, J. Y.; Luong, T.; Baid, G.; Wang, Z.; Vasudevan, V.; Ku, A.; Yang, Y.; Ayan, B. K.; et al. 2022a. Scaling Autoregressive Models for Content-Rich Text-to-Image Generation. *arXiv preprint arXiv:2206.10789*.

Yu, J.; Xu, Y.; Koh, J. Y.; Luong, T.; Baid, G.; Wang, Z.; Vasudevan, V.; Ku, A.; Yang, Y.; Ayan, B. K.; et al. 2022b. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*.

Yuan, M.; and Peng, Y. 2020. Bridge-GAN: Interpretable Representation Learning for Text-to-Image Synthesis. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(11): 4258–4268.

Zhang, H.; Xu, T.; Li, H.; Zhang, S.; Wang, X.; Huang, X.; and Metaxas, D. N. 2019. StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8): 1947–1962.

Zhang, H.; Yin, W.; Fang, Y.; Li, L.; Duan, B.; Wu, Z.; Sun, Y.; Tian, H.; Wu, H.; and Wang, H. 2021a. ERNIE-ViLG: Unified generative pre-training for bidirectional vision-language generation. *arXiv preprint arXiv:2112.15283*.

Zhang, Z.; Ma, J.; Zhou, C.; Men, R.; Li, Z.; Ding, M.; Tang, J.; Zhou, J.; and Yang, H. 2021b. UFC-BERT: Unifying Multi-Modal Controls for Conditional Image Synthesis. In *35th Conference on Neural Information Processing Systems, NeurIPS*, volume 32, 27196–27208.

Zhang, Z.; Xie, Y.; and Yang, L. 2018. Photographic Text-to-Image Synthesis with a Hierarchically-nested Adversarial Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 6199–6208.

Zhu, M.; Pan, P.; Chen, W.; and Yang, Y. 2019. DM-GAN: Dynamic Memory Generative Adversarial Networks for Text-To-Image Synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 5802–5810.