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

Fuxiang Wu1,2, Liu Liu3, Fusheng Hao1,2, Fengxiang He4, Lei Wang1,2, Jun Cheng1,2*

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.

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).

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
*J. Cheng is the corresponding author.

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
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 by selecting the best image. In phase (c), the inferior images with number of $N_{ref}−1$ 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 $N_{ref} = 5$ and $N_{ref} = 10$, 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 handling 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. 2021) proposed two-stage UFC-Diff that exploited the progressive non-autoregressive generation to improve the holistic consistency and support preserving operation. The UFC-Diff 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. 2021) introduced ERTNIE-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 autencoder and bidirectional auto-regressive transformers for the image-text bidirectional generating. Huang et al. (Huang et al. 2021) 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-
In this section, we propose a new decoding algorithm to improve the multi-modal language vision models to measure the alignment between the given textual description and the partial paths as early as possible. After that, we describe the decoding effectiveness and efficiency. However, for image synthesis via image tokens, a smaller decoding under some useful constraints (Susanto, Chollampatt, and Tan 2020) to select the k-best elements to better understand captions and synthesize high-quality photorealistic 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. 2021), 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 $\gamma$ is the encoder of the RQ-VAE (Lee et al. 2022). Let $\hat{C}$ be predicted image tokens from caption $T$, the transformer model predicts the current tokens $\hat{c}_j \in \hat{C}$, $j \in \{1, ..., |\hat{C}|\}$, based on the previous image tokens $\hat{c}_{1:j-1} \subset \hat{c}$ as follows,

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

where $\Theta$ is the parameter of the transformer. The function $\text{Multinomial}()$ samples a token from the multinomial probability distribution $p_{\Theta}$, 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 $N_{\text{ref}}$ paths, where each path consists of one image tokens $\hat{C}$ that can be transformed into the corresponding image $\hat{I} = \phi(\hat{C})$ by using the decoder $\phi$ of the RQ-VAE.

Since only one of $N_{\text{ref}}$ 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 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.

**Algorithm 1: Original Decoding in Transformer**

**Input:** The given caption $T$; The referent predicted size $N_{\text{ref}}$; The number of one image tokens $L$.

**Output:** A set of paths $\hat{G}$

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

**Algorithm 2: Reject Decoding in Transformer**

**Input:** The given caption $T$; The initial predicted size $N_0$; The end predicted size $N_e$; The reject threshold $\{\sigma_1, \sigma_2, ..., \sigma_K\}$; The size of predicted group $M$.

**Output:** A set of paths $\hat{G}$

1: $\hat{G} \leftarrow \{\{1\}, \{2\}, ..., \{N_0\}\}$
2: for $i \in \{1, K\}$ do
3: \hspace{1em} for $\hat{g} \in \hat{G}$ do
4: \hspace{2em} for $j \in \{1, M\}$ do
5: \hspace{3em} $\hat{c}_j \leftarrow \text{Multinomial}(p_{\Theta}(\hat{c}_j|\hat{g}, T))$
6: \hspace{2em} $\hat{g} \leftarrow \hat{g} \cup \{\hat{c}_j\}$
7: end for
8: end for
9: $S \leftarrow \{\}$
10: for $\hat{g} \in \hat{G}$ do \hspace{1em} the alignment between group and $T$
11: \hspace{2em} $s_g \leftarrow M_i(\hat{g}, T)$
12: \hspace{2em} $S \leftarrow S \cup \{\hat{g}, s_g\}$
13: end for
14: $R_e \leftarrow \max(|\hat{G}| \times \sigma_i, N_e)$
15: $\hat{G} \leftarrow \text{TopK}(S, R_e)$ \hspace{1em} $\bowtie$ reject $|\hat{G}| - R_e$ groups with lowest scores
16: end for
17: Return the set of predicted tokens $\hat{G}$
A train runs on a trail.

Figure 2: Diagrams of reserving count $R_c$ and the probability of reserving ground-truth under different $p^r$ of scorers $M_i$; 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 $N_{\text{ref}}$ 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,

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

(2)

where $N_1 = N_b$, and $N_{i+1} = \max([N_i \sigma_i], N_c)$. $N_b$, $N_c$ and $\sigma_i$ 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

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

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

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

(4)

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

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

(5)

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

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

(6)

Thus, the scale factor $f_{\text{scale}}$ will be a decreasing function with respect to $\xi$. In Eq. (4), given $\xi$, 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 $M_i$. Given a set of paths $\mathcal{G}$, we presume that $\mathcal{G}_s$ is the ground-truth element of $\mathcal{G}$, and the scorer $M_i$ can classify $\mathcal{G}_s$ from other elements with a probability $p$, i.e., $M_i(\mathcal{G}_s, T) \geq M_i(\mathcal{G}_k, T), k \in \{1, \ldots, N_{\mathcal{G}}\}$, and $N_{\mathcal{G}} = |\mathcal{G}|$. Thus, the probability of
Algorithm 3: Searching Reject Threshold

**Input:** The referent counter \( N_{\text{ref}} \), 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 \( \{N_i\}_{i=1}^K \).

1: \( N_i \leftarrow N_e, \forall i \in \{2, ..., K\} \);
2: \( N_1 \leftarrow N_b \);
3: for \( i \in \{2, ..., K\} \) do
4: \( C_{\text{res}} \leftarrow \max(N_{\text{ref}} - \sum_j N_j, 0) \)
5: \( N_i \leftarrow \min([N_{i-1} p_r], C_{\text{res}} + N_i) \)
6: end for
7: if \( \sum_j N_j \neq N_{\text{ref}}K \) then
8: Return \( \{N_i\}_{i=1}^K \), Fail
9: end if
10: Return \( \{N_i\}_{i=1}^K \), Successful

\( \mathcal{M}_i(\hat{g}_s, T) \geq \mathcal{M}_i(\hat{g}_e, T), \forall \hat{g}_e \in \hat{G} \) is \( p_{\phi}^{N_{\phi}^{-1}} \). In line 15 of Algorithm 2, we can split \( \hat{g} \) into two sets \( \hat{g}_0 = \text{Topk}(\hat{S}, R_c) \) and \( \hat{g}^1 = \hat{g} \setminus \hat{g}_0 \), where \( R_c \) is size of preserving the paths and function \( \text{Topk}(\hat{S}, R_c) \) is to reject the \( [\hat{g}] - R_c \) groups with the lowest scores. \( \hat{S} \) is score set of \( \hat{g} \). Thus, the probability of preserving the ground truth \( \hat{g}_s \) in \( \hat{g} \setminus \hat{g}_0 \) is \( P_{R_c} \), which is defined as follows,

\[
P_{(R_c=1)} = p_{\phi}^{N_{\phi}^{-1}},
\]

\[
P_{(R_c=j+1)} = (1 - P_{(R_c=j)} + P_{(R_c=j)}),
\]

where the first term of Eq. (8) is the probability of \( P(\hat{g}_0) = \hat{g}_s \). The probability \( P_{R_c} \) to 1 when \( R_c \rightarrow N_b \); and \( P_{(R_c=N_{\phi})} = 1 \). For examples, in Figure 2, the plots indicate that a larger \( p_{\phi}^{N_{\phi}^{-1}} \) would make \( P_{R_c} \) increasing faster and can exclude more trailing elements. When the partial path is short and \( p_{\phi}^{N_{\phi}^{-1}} \) 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 \( \hat{g} \), which could be dropped with little cost.

**Configure of Reject Counters** In Algorithm 3, given the reference counter \( N_{\text{ref}} \) 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 \( N_i/N_{\text{ref}} \).

**Multimodal Language-Vision Models**

In Algorithm 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 \( N_\omega \) is the dimension of word embedding and \( |T| \) is the size of caption \( T \). Thus, the representation of \( T \) is calculated as follows,

\[
f_M(\omega) = L_{\text{mean,1}}(L_{\text{Trans}}(L_{\text{NAT}}(\omega))),
\]

where \( L_{\text{NAT}} \) denotes a linear layer to translate the embedding \( \omega \) into the hidden features, followed by a layer normalization; \( L_{\text{Trans}} \) includes several transformer blocks, consisting of a multi-head self-attention layer, layer normalization, and a multi-layer perceptron; \( L_{\text{mean,1}} \) 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 \( C \in \mathbb{R}^{L \times 4} \) of an image, we employ the decoder \( \phi \) of the RQ-VAE to get the embedding \( \upsilon = \mathcal{E}_\phi(C) \in \mathbb{R}^{L \times 4 \times N_\upsilon} \), where \( N_\upsilon \) is the dimension of the visual embedding i.e.,

\[
f_M(\upsilon) = L_{\text{mean,1}}(L_{\text{NAT}}(L_{\text{Trans}}(L_{\text{NAT}}(L_{\text{mean,1}}(\upsilon))))).
\]

**Training with Contrastive Loss** Given the \( k \)-th caption \( T \) and residual quantization tokens \( C \) in a batch, their embeddings are,

\[
\omega_k = \text{Norm}(f_M(\mathcal{E}_{\text{GPT2}}(T))),
\]

\[
\upsilon_k = \text{Norm}(f_M(\mathcal{E}_\phi(C))),
\]

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

\[
-\sum_i \left\{ \frac{\exp(\hat{\omega}_i \cdot \hat{\upsilon}_i)}{\sum_k \exp(\hat{\omega}_k \cdot \hat{\upsilon}_k)} + \frac{\exp(\hat{\upsilon}_k \cdot \hat{\upsilon}_i)}{\sum_k \exp(\hat{\omega}_k \cdot \hat{\upsilon}_k)} \right\}.
\]

Besides, as shown in the Figure 3, we train and evaluate the alignment between part image tokens and the given caption. For model \( M_i = \{f_M, f_M\} \), we sample a subset of tokens \( C' \subset C \) 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 \( \{M_i\}_{i=1}^{8} \).

**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 denoted by the superscript “pre”, which is trained by CC-3M, CC-12M, and YFCC-subset.

**Evaluating Metrics**

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

\[\text{github.com/kakaobrain/rq-vae-transformer}\]

\[\text{github.com/google-research-datasets/conceptual-12m}\]

\[\text{github.com/openai/CLIP/blob/main/data/yfcc100m.md}\]
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 these 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 $RP_{\text{trans}}$. Since transformer models could extract high-quality features, similar to works (Park et al. 2021), we utilize the language-vision transformer, like $M_1$, to extract base features to compute the R-precision, denoted as $RP_{\text{trans}}$.

### 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$_{\text{pre}}^{5}$ encoding under $N_{\text{ref}} = 10$ and trained by CC-3M, CC-12M, and YFCC-subset, the IS of Our$_{10}$ increases 1.27. The FID of Our$_{10}$ largely decreases 2.11. The $RP_{\text{cnn}}$ and $RP_{\text{trans}}$ 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$_{\text{coco}}^{10}$, the IS of Our$_{10}^{\text{coco}}$ increases by 0.71. The FID of Our$_{10}^{\text{coco}}$ decreases 0.09. The $RP_{\text{cnn}}$ and $RP_{\text{trans}}$ 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, $RP_{\text{cnn}}$, and $RP_{\text{trans}}$ under Different $N_{\text{e}}$: In Figure 4, the results demonstrate the influence of $N_{\text{e}}$ with $N_{\text{b}} = 20$: the first figure shows that the number $N_{\text{e}}$ will be dropping faster with larger $N_{\text{e}}$. With increasing $N_{\text{e}}$, 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 $N_{\text{e}}$, the decoding may be hard to cover the high-quality path, because the number of inner paths $N_{\text{e}}$ is dropping faster with larger $N_{\text{e}}$. With increasing $N_{\text{e}}$, 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 $N_{\text{e}}$, the decoding may be hard to cover the high-quality path, because the number of inner paths $N_{\text{e}}$ is dropping faster. The results indicate that the reject threshold is important, and the $N_{\text{e}}$ prefers a relatively small number. However, IS, $RP_{\text{cnn}}$, and $RP_{\text{trans}}$ are better than those of the corresponding baseline.

b) IS, FID, $RP_{\text{cnn}}$, and $RP_{\text{trans}}$ 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”, $x \in \{8, 16, \ldots, 64\}$, $RP_{\text{trans}}$ 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 $RP_{\text{trans}}$ will increase with more training epochs. By using the vision models with higher $RP_{\text{trans}}$, the FID of our model will decrease.
and IS is on an ascending trend. By exploiting different vision models, the last figure shows that $R_{P_{\text{cnn}}}$ and $R_{P_{\text{trans}}}$ will increase with more training epochs of vision models (with higher $R_{P_{\text{trans}}}$), 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, $R_{P_{\text{cnn}}} = 68.06$, $R_{P_{\text{trans}}} = 67.89$, $FID = 8.63$, 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 Our$^{\text{pre}}_{5}$ and Our$^{\text{coco}}_{5}$ are better than those of RQ-Transformer$^{\text{pre}}_{5}$ and RQ-Transformer$^{\text{coco}}_{5}$, respectively. For the referent number $N_{\text{ref}} = 10$, images of Our$^{\text{pre}}_{10}$ and Our$^{\text{coco}}_{10}$ are also better than those of RQ-Transformer$^{\text{pre}}_{10}$ and RQ-Transformer$^{\text{coco}}_{10}$, 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 Our$^{\text{pre}}_{5}$ and Our$^{\text{coco}}_{10}$ include more vivid visual details of “home with lots of wood” than those of RQ-Transformer$^{\text{pre}}_{3}$ and RQ-Transformer$^{\text{pre}}_{10}$. Besides, Our$^{\text{coco}}_{5}$ retrieves the same image of “home” as that of RQ-Transformer$^{\text{coco}}_{10}$, and Our$^{\text{coco}}_{5}$ generates the better image than that of Our$^{\text{coco}}_{10}$. 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. RQ-Transformer$^{\text{pre}}_{5}$ will re-rank the first 5 images to get the best image, and Our$^{\text{pre}}_{5}$ would search the sub-paths of the total 10 generated images when $N_{e} = 10$. $[1, 5] \rightarrow x$ 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, RQ-Transformer$^{\text{pre}}_{5}$ and RQ-Transformer$^{\text{coco}}_{5}$, 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”, RQ-Transformer$^{\text{pre}}_{5}$ selects the $4^{th}$ image from the first 5 images. When $N_{e} = 1$, Our$^{\text{pre}}_{5}$ selects the $9^{th}$ image, which is better than the $4^{th}$ image. When $N_{e} = 2$, Our$^{\text{pre}}_{5}$ generates the $9^{th}$ and $3^{rd}$ 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.
**RQ-Transformer**

Large black and white panda bear walking around in an enclosure.

![GroundTruth](image1)

Figure 7: Synthesizing examples with different \(N_e\).

A man with a hat riding on a surf board.

![GroundTruth](image2)

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

---

**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 multi-
modal 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, KCFXZZ20201221173411032, Y795001001, JCYJ2020010913416531), CAS Key Technology Talent Program.

References


