Unlocking the Capabilities of Masked Generative Models for Image Synthesis via Self-Guidance

Jiwan Hur[1] **Dong-Jae Lee**[1] **Gyojin Han**[1] **Jaehyun Choi**[1]
Yunho Jeon[2†] **Junmo Kim**[1†]
[1]KAIST, South Korea [2]Hanbat National University, South Korea
{jiwan.hur, jhtwosun, hangj0820, chlwogus}@kaist.ac.kr
yhjeon@hanbat.ac.kr, junmo.kim@kaist.ac.kr

Code is available at: `https://github.com/JiwanHur/UnlockMGM`

Figure 1: **Comparison of sampled images using 18-step MaskGIT [4] without (top) and with the proposed self-guidance (bottom) on ImageNet 512×512 (left) and 256×256 (right) resolutions.** Each paired image is sampled using the same random seed and sampling hyperparameters. The proposed self-guidance effectively improves the capabilities of the masked generative models.

Abstract

Masked generative models (MGMs) have shown impressive generative ability while providing an order of magnitude efficient sampling steps compared to continuous diffusion models. However, MGMs still underperform in image synthesis compared to recent well-developed continuous diffusion models with similar size in terms of quality and diversity of generated samples. A key factor in the performance of continuous diffusion models stems from the guidance methods, which enhance the sample quality at the expense of diversity. In this paper, we extend these guidance methods to generalized guidance formulation for MGMs and propose a self-guidance sampling method, which leads to better generation quality. The proposed approach leverages an auxiliary task for semantic smoothing in vector-quantized token space, analogous to the Gaussian blur in continuous pixel space. Equipped with the parameter-efficient fine-tuning method and high-temperature sampling, MGMs with the proposed self-guidance achieve a superior quality-diversity trade-off, outperforming existing sampling methods in MGMs with more efficient training and sampling costs. Extensive experiments with the various sampling hyperparameters confirm the effectiveness of the proposed self-guidance.

†Corresponding authors: junmo.kim@kaist.ac.kr, yhjeon@hanbat.ac.kr

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Keywords

Image synthesis, discrete diffusion models, masked generative models, sampling guidance, parameter-efficient fine-tuning

1 Introduction

With the advent of generative adversarial networks (GANs) [14], generative models have attracted significant attention for their powerful ability to synthesize highly realistic images. However, due to the limited mode coverage and training instability, recently, likelihood-based models such as diffusion models [20] have been actively researched. Diffusion models have shown promising results for their diverse and high-quality samples, surpassing GANs on class conditional image synthesis [10].

A key factor in the success of diffusion models stems from the various guidance techniques, which enhance the fidelity of the generated image at the expense of diversity. The guidance is usually conducted throughout the sampling process of diffusion models, driving it toward enhancing specific information such as class [10, 19], text [44, 52], or details in the image [22]. From the practical perspective, however, diffusion models suffer from sampling inefficiency, requiring hundreds of sampling steps to generate high-quality images. Moreover, guidance techniques further decrease the sampling efficiency, as they typically require twice as many model inferences as the original process. For instance, ADM [10] with guidance requires $\sim$500 model inferences.

On the other side, masked generative models (MGMs) have shown superior trade-offs between sampling quality and speed compared to (continuous) diffusion models [4, 35, 36]. MGMs use an absorbing state (`[MASK]`) diffusion process [1] and aim to generate discrete tokens by predicting the masked region, similar to BERT [9]. Specifically, they use the Markov transition matrix to model the diffusion process and sample the tokens with the categorical distribution. Recently, vector-quantized (VQ) image token-based MGMs with (non-)autoregressive transformer [11, 4] have demonstrated an efficient sampling process, providing an order of magnitude fewer sampling steps (e.g., $\sim$18 steps) than continuous diffusion models [4].

While continuous diffusion models utilize guidance sampling to enhance generation quality, MGMs typically utilize sampling with low-temperature Gumbel noise in categorical distribution to enhance the quality [4, 35]. However, low-temperature sampling may impose *multi-modality problem* [15, 39, 60], where the non-autoregressive sampling process fails to generate plausible outputs due to the lack of sequential dependencies. As a result, the upper bound of the sample quality generated by MGMs was relatively limited. Several approaches have been suggested to improve the sampling quality of MGMs, such as discrete predictor-corrector-based methods [35, 36] that train a second transformer to discern the unrealistic tokens. However, despite the improved sampling, MGMs still underperform in terms of FID scores [18] compared to state-of-the-art continuous diffusion models such as LDM [44] on ImageNet benchmark [8], even when the model sizes are similar.

To overcome such limitations and enhance the sample quality of MGMs, we propose discrete self-guidance sampling (Fig. 1). Self-guidance [22] in continuous diffusion models improves the generation quality by enhancing the fine-grained detail of the sample by guiding the diffusion process with coarse-grained information within intermediate diffusion steps, similar to that of classifier-free guidance, which utilizes unconditional generation to enhance the quality of class conditional generation. In continuous space, the coarse-grained information can be easily obtained with spatial smoothing, like Gaussian blur. To apply self-guidance in MGMs, we first define the general guidance formulation for the discrete diffusion models and introduce self-guidance in MGMs. However, while it is simple to define the coarse-grained information in continuous space, the VQ token space [43, 11] in MGMs is absent of continuous semantic structure and cannot apply such a simple strategy, e.g., blur [2]. Furthermore, we cannot directly define coarse-grained, i.e., semantically smoothed outputs in the VQ token space. Therefore, we introduce an auxiliary task specifically designed for semantic smoothing of the VQ token space, which enables the network to selectively remove details such as local patterns while preserving overall information in VQ token spaces. Consequently, by guiding the MGMs with semantically smoothed information, we can enhance the quality of MGMs (Fig. 2) even with high temperatures, therefore achieving both high quality and diversity compared to previous MGMs. For efficient implementation of the discrete self-guidance, we introduce a plug-and-play module with parameter-efficient fine-tuning [49] to pre-trained MGMs, leveraging the generative prior in the MGMs while enabling efficient training with a few parameters and epochs.

Input image | Masked image (*m.r.*=90%) | One step prediction | Self-guidance w/ spatial smoothing (SAG) | **Self-guidance w/ semantic smoothing (Ours)**

Figure 2: Visualization of the effect of guidance using spatial smoothing (SAG) [22] and the proposed semantic smoothing. We tokenize the input image using VQGAN [11] encoder, mask the 90% of VQ tokens, and predict $\hat{\boldsymbol{x}}_{0,t}$ using MaskGIT [4]. With the proposed self-guidance leveraging semantic smoothing, generated sample quality is improved by enhancing fine-scale details.

Experimental results demonstrate that the proposed guidance effectively improves the sample quality with only 10 epochs of fine-tuning. Notably, combined with a high sampling temperature, the proposed guidance not only improves the quality of generated samples but also keeps diversity high, showing a superior quality-diversity trade-off compared to other improved sampling techniques for MGMs and other generative families with similar model sizes.

2 Background

2.1 Masked Generative Models

Discrete diffusion models [1] aim to generate categorical data $\boldsymbol{x}_0 \in \mathbb{R}^N$ with a length of N and K categories. The forward Markov process $q(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)$ gradually corrupts data $\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_T$ until the marginal distribution of $\boldsymbol{x}_T \sim p(\boldsymbol{x}_T)$ becomes stationary. Then starting from the $\boldsymbol{x}_T$, the learned reverse Markov process $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ gradually recovers the corrupted data to generate $\boldsymbol{x}_0$.

Masked Generative Models (MGMs) are a family of discrete diffusion models that use an *absorbing state diffusion process* [1], in which they gradually mask the input tokens by replacing them with a mask token (`[MASK]`) and learn to predict the masked region. Recently, vector-quantized (VQ) image token-based MGMs [4, 35, 36] have achieved a superior trade-off between sampling time and quality for image synthesis compared to Gaussian (continuous) diffusion models [10, 44] through the non-autoregressive parallel decoding process. MGMs are trained to predict the masked region similar to BERT [9] in natural language processing (NLP), but with various masking ratios $\gamma_t \in (0, 1]$, where γ is pre-defined mask scheduling function that masks $n = \lceil \gamma(t/T) \cdot N \rceil$ tokens from N total tokens. Without loss of generality, we assume the unmasked region of the mask $\boldsymbol{m}_t$ is 1, and the masked region is 0. Then, given external condition c (e.g., class or text) and masked input $\boldsymbol{x}_t = \boldsymbol{x}_0 \odot \boldsymbol{m}_t$, where the mask $\boldsymbol{m}_t$ is randomly sampled according to masking ratio γ_t, MGMs are trained to predict clean data $\boldsymbol{x}_0$ (or equivalently, to predict the masked regions) by the objective function

$$\mathcal{L}_{mask} = -\mathbb{E}_{\boldsymbol{x},t}\left[\sum_{\forall i \in [1,N], \boldsymbol{m}_i = 0} \log p_\theta(\boldsymbol{x}_0^i|\boldsymbol{x}_t, c)\right]. \tag{1}$$

Here, we denote the i-th token of $\boldsymbol{x}_0$ as $\boldsymbol{x}_0^i$.

After the training, to sample the image $\boldsymbol{x}_0$, MGMs typically use an iterative prediction-masking procedure starting from the blank canvas $\boldsymbol{x}_T$ (i.e., all tokens are masked). Given (partially) masked image token $\boldsymbol{x}_t$ sampled from the previous step, MGMs first predict all tokens $\hat{\boldsymbol{x}}_{0,t} \sim p_\theta(\hat{\boldsymbol{x}}_{0,t}|\boldsymbol{x}_t)$ simultaneously, where $\hat{\boldsymbol{x}}_{0,t}$ denotes the prediction of $\boldsymbol{x}_0$ at timestep t and we omit c for simplification. Then $\hat{\boldsymbol{x}}_{0,t}$ is masked according to masking ratio γ_{t-1} to obtain $\boldsymbol{x}_{t-1}$. MGMs usually sample $\boldsymbol{m}_{t-1} \sim p(\boldsymbol{m}_{t-1}|\boldsymbol{m}_t, \hat{\boldsymbol{x}}_{0,t})$ which is equivalent to sample $\boldsymbol{x}_{t-1}$ since $\boldsymbol{x}_{t-1} = \boldsymbol{m}_{t-1} \odot \hat{\boldsymbol{x}}_{0,t}$. Randomly sampling the $\boldsymbol{m}_{t-1}$ can be one choice; however, the prediction $\hat{\boldsymbol{x}}_{0,t}$ may have numerous errors, thus randomly selecting the mask can produce sub-optimal results by potentially masking relatively accurate tokens while leaving unrealistic tokens unmasked. To overcome this, various research adopts improved sampling methods such as utilizing the output confidence of the $\hat{\boldsymbol{x}}_{0,t}$ since confident tokens tend to be more accurate [4] or train an external corrector to classify the realistic tokens [35, 36].

However, these improved sampling may impose a problem, named *multi-modality problem* that is a well-known problem in non-autoregressive parallel sampling [15, 39, 60]. Given the input such as $\boldsymbol{x}_t$ and class condition c, the model can have multiple plausible outputs, which brings challenges to the non-autoregressive model as they generate each token independently. In an extreme case where the input token is all masked, and the model predicts output only using a given external condition such as class, each token can predict *easy* token more confidently and correctly, such as a background in every token. As a result, correctness-based sampling may result in images only filled with background images. To resolve this problem, various MGMs adopt additional randomness to sample $\boldsymbol{m}_t$, such as sampling with temperature [4, 35, 36]. Let the l_t measure the realism of sampled tokens $\hat{\boldsymbol{x}}_{0,t}$, such as the confidence scores of sampled tokens in MaskGIT [4]. Then MGMs sample $\boldsymbol{m}_{t-1}$ by selecting top-k elements of $\tilde{l}_t = l_t + \tau \cdot (t/T)\boldsymbol{n}$ according to γ_t, where $\boldsymbol{n}$ denotes the sampling noise such as i.i.d. Gumbel noise and τ is temperature scale that is annealed according to the timesteps. Generally, high-temperature sampling results in more diverse samples while degrading the sample quality.

2.2 Sampling Guidance

Iterative sampling processes of diffusion models are often guided by external networks or themselves. Therein, salient information-based guidance has been actively explored in continuous diffusion models [19, 22, 52] for high-quality image synthesis. Let h_t be salient information of $\boldsymbol{x}_t$ and $\bar{\boldsymbol{x}}_t$ be a perturbed sample that lacks h_t. h_t can be internal information within $\boldsymbol{x}_t$ or an external condition, or both. Lee et al. [22] proposed a general guidance technique that guides the sampling process toward enhancing the information h_t, and the equation for the sampling process is:

$$\tilde{\epsilon}(\bar{\boldsymbol{x}}_t, h_t) = \epsilon(\bar{\boldsymbol{x}}_t) + (1+s)(\epsilon(\bar{\boldsymbol{x}}_t, h_t) - \epsilon(\bar{\boldsymbol{x}}_t)), \tag{2}$$

where ϵ is a score function, $\tilde{\epsilon}$ is a guided score function of the continuous diffusion model, and s is a guidance scale. For instance, in the setting $h_t = c$ and $\bar{\boldsymbol{x}}_t = \boldsymbol{x}_t$, the Eq. (2) collapses to classifier-free guidance (CFG) [19], which guides the sampling toward the given class distribution. Lee et al. [22] propose using adversarial blurring, enhancing the fine-scale details of the sample. Generally, using a large guidance scale s enhances the quality of generated samples while reducing the diversity.

Recently, discrete CFG [52, 5] has been introduced to improve the correlation between the input class or text condition c and the images generated by discrete diffusion models as below equation:

$$\log p(\tilde{\boldsymbol{x}}_t) = \log p(\boldsymbol{x}_t) + (1+s)(\log p(\boldsymbol{x}_t|c) - \log p(\boldsymbol{x}_t)), \tag{3}$$

where $\tilde{\boldsymbol{x}}_t$ denotes the guided token.

Unlike CFG in the continuous domain, discrete CFG estimates the probability distribution $p(\boldsymbol{x}_t|c)$ directly. However, it requires a specific training strategy and paired labels such as class and text.

2.3 Parameter-Efficient Fine-Tuning

MGMs across various domains adopt transformer architecture due to their superior ability to handle context with bidirectional attention [9, 13, 4, 57, 63]. However, training a transformer from scratch requires significant computational resources due to the quadratic complexity of the attention mechanism. In recent years, parameter-efficient fine-tuning (PEFT) techniques have received significant attention, especially in light of the growing size and complexity of pre-trained models. PEFT adapts large pre-trained models to specific tasks or datasets by tuning a small portion of parameters [58, 55] or introducing task-specific parameters [45, 23, 49], effectively transferring knowledge without extensive retraining. Notably, by preserving most of the parameters, the fine-tuned model effectively preserves knowledge with few forgetting.

3 Methods

3.1 Generalized Information-Based Guidance for Discrete Diffusion Models

Similar to the general guidance in continuous diffusion models in Eq. (2), discrete CFG in Eq. (3) can be extended to the generalized information-based guidance from the optimization perspective. Given some salient information h_t, we aim to sample $\boldsymbol{x}_t$ which maximizes $p(\bar{\boldsymbol{x}}_t|h_t)$. Simultaneously, for the correlation between the information and sample, $p(h_t|\bar{\boldsymbol{x}}_t)$ also needs to be maximized as stated in the equation below which is from Tang et al. [52]:

$$\arg\max_{\bar{\boldsymbol{x}}_t}[\log p(\bar{\boldsymbol{x}}_t|h_t) + s\log p(h_t|\bar{\boldsymbol{x}}_t)]. \tag{4}$$

Using the Bayes' theorem and ignoring the prior probability term for salient information, the optimization goal can be represented as:

$$\arg\max_{\bar{\boldsymbol{x}}_t}[\log p(\bar{\boldsymbol{x}}_t) + (1+s)(\log p(\bar{\boldsymbol{x}}_t|h_t) - \log p(\bar{\boldsymbol{x}}_t))]. \tag{5}$$

Then, the Eq. (5) guides the sampling process toward enhancing the relevance of the sample and salient information h_t.

In the inference stage, various MGMs adopt to predict unmasked state $\hat{\boldsymbol{x}}_{0,t}$ rather than directly predicting $\boldsymbol{x}_{t-1}$. If we limit h_t to the internal information of $\boldsymbol{x}_t$ to make h_t removable from the $\boldsymbol{x}_t$ through an information bottleneck module $\mathcal{H}_\phi$; in other words, if $\bar{\boldsymbol{x}}_t = \mathcal{H}_\phi(\boldsymbol{x}_t)$ and $p(\boldsymbol{x}_t) = p(\bar{\boldsymbol{x}}_t, h_t)$ get satisfied, we can sample next state for the denoising step as below:

$$\log p_\theta(\tilde{\boldsymbol{x}}_{0,t}|\boldsymbol{x}_t) = \log p_\theta(\bar{\boldsymbol{x}}_{0,t}|\mathcal{H}_\phi(\boldsymbol{x}_t))) + (1+s)(\log p_\theta(\hat{\boldsymbol{x}}_{0,t}|\boldsymbol{x}_t) - \log p_\theta(\bar{\boldsymbol{x}}_{0,t}|\mathcal{H}_\phi(\boldsymbol{x}_t))), \tag{6}$$

when a MGM with parameter θ predict $\hat{\boldsymbol{x}}_{0,t}$ from $\boldsymbol{x}_t$ and $\bar{\boldsymbol{x}}_{0,t}$ from $\mathcal{H}_\phi(\boldsymbol{x}_t)$. This implies that by defining the information bottleneck module $\mathcal{H}_\phi$ that can selectively subtract salient information h_t from $\boldsymbol{x}_t$ in the discrete space, we can guide the sampling of MGMs in a direction that enhances h_t. Since we aim to improve the sample quality of MGMs by presenting a novel guidance method, it is necessary to define $\mathcal{H}_\phi$ that can remove information about the fine details of the samples. For continuous diffusion models, utilizing samples spatially smoothed by Gaussian blur for guidance has been helpful in improving sample quality by restricting fine-scale information [22]. This motivates us to investigate guidance with smoothed output for discrete domains, especially for VQ tokens.

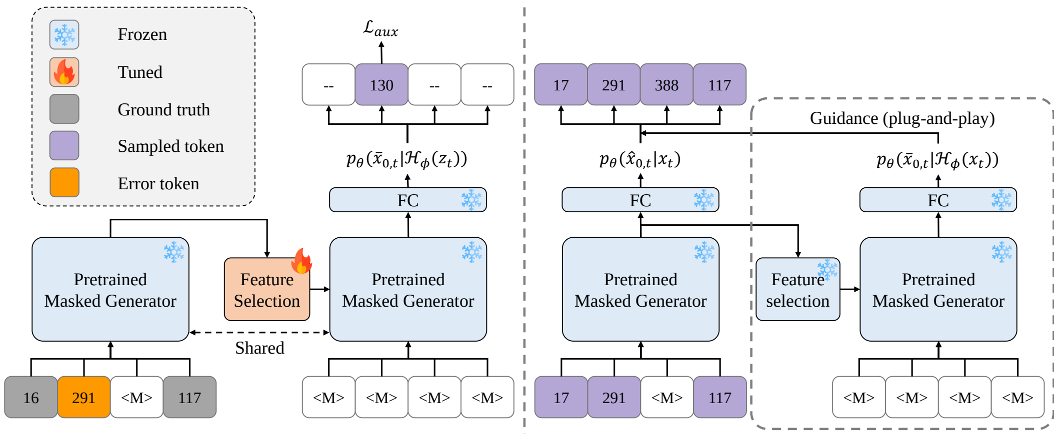
3.2 Auxiliary Task Learning for Semantic Smoothing on VQ Token space

We aim to apply smoothing, such as Gaussian blur, for VQ tokens, as discussed in the previous section, to implement guidance in the discrete domain. However, unlike natural images which have inherent, observable patterns and structures, the latent space of autoencoders often lacks such properties [2]. For instance, two successive tokens in VQ codebooks, such as the 11th and 12th tokens, are not semantically linked. As a result, applying Gaussian blur in the VQ token cannot produce meaningful representations. Nevertheless, we empirically found that applying Gaussian blur in the probability spaces, i.e. blurring the output logits of the generator $p_\theta(\hat{\boldsymbol{x}}_0|\boldsymbol{x}_t)$ similar to Lee et al. [22] can provide meaningful guidance. However, the improvement is marginal because Gaussian blur is not a suitable information bottleneck for subtracting fine details in VQ token space.

To overcome this, we introduce an auxiliary task designed to leverage semantic smoothing for VQ tokens, a process that selectively removes details such as local patterns while preserving overall information in VQ token space. Specifically, given the masked input $\boldsymbol{x}_t$, masked generator predicts $p_\theta(\hat{\boldsymbol{x}}_{0,t}|\boldsymbol{x}_t)$. We aim to train information bottleneck $\mathcal{H}_\phi$ to generate the semantically smoothed output $p_\theta(\hat{\boldsymbol{x}}_{0,t}|\mathcal{H}_\phi(\boldsymbol{x}_t))$. However, training $\mathcal{H}_\phi$ directly is challenging, as we cannot define semantically smoothed outputs. To naturally impose a model to generate semantically smoothed output, we leverage error token correction, originally introduced in non-autoregressive machine translation to mitigate the compounding decoding error in the iterative sampling process [24]. During the error token correction, input unmasked tokens are randomly replaced with error tokens, and the model learns to correct them. To be more specific, let $\boldsymbol{z}_t$ be a corrupted data where some tokens in $\boldsymbol{x}_t$ are replaced with error tokens with some probability p. Then, the objective function for the auxiliary task to update ϕ can be

$$\mathcal{L}_{aux} = -\mathbb{E}_{\boldsymbol{x},t}\left[\sum_{\forall i\in[1,N],\boldsymbol{m}_i=0} \log p_\theta(\boldsymbol{x}_0^i|\mathcal{H}_\phi(\boldsymbol{z}_t), c)\right]. \tag{7}$$

From the perspective of *Vicinal Risk Minimization* (VRM) [6, 59], a vicinity distribution p_ϕ minimizes the empirical risk for all data points $\boldsymbol{x}_t$ given a vicinity of the data $\boldsymbol{z}_t$. Given that randomly replaced



(a) Training feature selection module (b) Guided sampling

Figure 3: (a) Fine-tuning the feature selection module $\mathcal{H}_\phi$ (TOAST [49]). With the auxiliary objective in Eq. (7), $\mathcal{H}_\phi$ implicitly learns to smooth erroneous input z_t to address semantic outliers (Section 3.2). (b) During the sampling steps, self-guidance can be efficiently implemented by leveraging the feature map from the generative process. $\mathcal{H}_\phi$ performs semantic smoothing on the input $\boldsymbol{x}_t$, guiding the sampling process toward enhancing fine-scale details in the generated sample.

error tokens often act as semantic vicinities within the input data, to minimize the overall empirical risk, the model implicitly learns to smooth vicinities of $\boldsymbol{z}_t$. This involves leveraging coarse information from the surrounding context while minimizing the fine-scale details in the presence of unknown input errors[1]. However, training a network from scratch with Eq. (7) does not ensure that the outputs will resemble real images, potentially converging on trivial solutions that may be undesirable, in addition to being computationally expensive.

3.3 Efficient Implementation

To mitigate the aforementioned problem, we adopt a parameter-efficient fine-tuning (PEFT) method to utilize deep image priors in the pre-trained masked generator and to enhance the training efficiency of transfer learning. Among various PEFT methods, we adopt TOAST [49], which shows favorable performance in various visual and linguistic tasks under transformer architecture. With a frozen pre-trained backbone, TOAST selects task-relevant features from the output and feeds them into the model by adding them to the value matrix of self-attention. This top-down signal steers the attention to focus on the task-relevant features, effectively transferring the model to other tasks without changing parameters.

Besides its effectiveness in transfer learning, TOAST brings more practical strengths to our task. Before the discussion, it is important to note that masked generative models exhibit a strong training bias. Because the training data does not contain error tokens, the model predicts unmasked input as identical and unable to correct error tokens. Since the model is trained only to consider masked regions, we propose to use a blank canvas as an input (i.e., all tokens are masked $\boldsymbol{x}_T$), enabling the model to make corrections in response to error tokens. However, most PEFT methods lack a direct solution for incorporating information about $\boldsymbol{x}_t$ when the input is replaced with all masked tokens. On the other hand, we empirically found that simply replacing the input for the second stage of TOAST as $\boldsymbol{x}_T$ can mitigate this problem without a performance drop (Fig. 3 a).

Furthermore, the two-stage approach of TOAST can be efficiently implemented in the sampling process. The generator first sample $p_\theta(\hat{\boldsymbol{x}}_{0,t}|\boldsymbol{x}_t)$ from the input $\boldsymbol{x}_t$. The hidden state obtained from the generation stage can be recycled for the second stage to produce guidance logit $p_\theta(\bar{\boldsymbol{x}}_{0,t}|\mathcal{H}_\phi(\boldsymbol{x}_t))$

[1]For a simple example, a common approach to correcting unknown *numerical outliers* in a 1-dimensional signal, such as impulse noise in time series data, is smoothing the signal, like applying low-pass filters. Similarly, to correct the unknown error tokens, which are *semantic outliers* in our case, we expect that the model implicitly learns to smooth $\boldsymbol{z}_t$ to deal with unknown error tokens.

Table 1: Quantitative comparison of various generative models for class-conditional image generation on ImageNet 256×256 and 512×512 resolutions. "↓" or "↑" indicate lower or higher values are better. †: taken from MaskGIT [4], ‡: taken from VAR [53], ∗: taken from Token-Critic [35].

			ImageNet 256×256				ImageNet 512×512			
Model	Type	NFE	FID↓	IS↑	Prec↑	Rec↑	FID↓	IS↑	Prec↑	Rec↑
BigGAN-deep [3]	GANs	1	6.95	224.5	**0.89**	0.38	8.43	177.9	0.85	0.25
GigaGAN [26]	GANs	1	3.45	225.5	0.84	0.61	–	–	–	–
ADM [10]	Diff.	250	10.94	101.0	0.69	**0.63**	23.24	58.0	0.73	**0.60**
ADM (+ SAG) [22]	Diff.	500	9.41	104.7	0.70	0.62	–	–	–	–
CDM [21]	Diff.	250	4.88	158.7	–	–	–	–	–	–
LDM-4 [44]	Diff.	250	10.56	103.4	0.71	0.62	–	–	–	–
LDM-4 (+ CFG) [44]	Diff.	500	3.60	247.7	–	–	–	–	–	–
DiT-L/2‡ (+ CFG) [41]	Diff.	500	5.02	167.2	0.75	0.57	–	–	–	–
VQVAE-2† [43]	AR	5120	31.11	∼45	0.36	0.57	–	–	–	–
VQGAN† [11]	AR	∼1024	18.65	80.4	0.78	0.26	7.32	66.8	0.73	0.31
VQ-Diffusion [16]	Discrete.	100	11.89	–	–	–	–	–	–	–
ImprovedVQ. (+ CFG) [52]	Discrete.	200	4.83	–	–	–	–	–	–	–
MaskGIT∗ [4]	Mask.	18	6.56	203.6	0.79	0.48	8.48	167.1	0.78	0.46
Token-Critic [35]	Mask.	36	4.69	174.5	0.76	0.53	6.80	182.1	0.73	0.50
DPC-light [36]	Mask.	66	4.8	249.0	0.80	0.50	6.09	228.1	0.81	0.46
DPC-full [36]	Mask.	180	4.45	244.8	0.78	0.52	6.06	218.9	0.80	0.47
Ours (T=12)	Mask.	24	3.35	259.7	0.81	0.52	**5.38**	226.0	**0.88**	0.36
Ours (T=18)	Mask.	36	**3.22**	**263.9**	0.82	0.51	5.57	**233.2**	**0.88**	0.35

(Fig. 3 b). We note that different from the fine-tuning step where the error tokens $\boldsymbol{z}_t$ are used to train $\mathcal{H}_\phi$, the sampling process directly utilizes $\boldsymbol{x}_t$ to generate the semantically smoothed tokens $\mathcal{H}_\phi(\boldsymbol{x}_t)$.

4 Related Works

Generative Adversarial Networks (GANs). Trained by adversarial objective [14], GANs have shown impressive performance in image synthesis with only 1-step model inference [27, 28, 47, 3, 33]. However, despite their practical performance, the training instability and limited mode coverage caused by the adversarial objective [46, 62] is still a bottleneck for their broader applications.

Diffusion Models. Recent diffusion models in continuous space mostly utilize Gaussian noise to perturb the input data and learn to predict clean data from the noisy data [50]. After the advent of DDPM [20], diffusion models have rapidly grown with architecture improvements [38, 41], improved training strategies [29, 7, 25], improved samplings [51, 37], latent space models [44], and sampling guidance [10, 19, 22], outperforming various generative families such as GANs in various domains.

Masked Generative Models. With the recent success of transformers [54] and GPT [42] in NLP, generative transformers have also been adopted in image synthesis. To mimic the generative process of transformers for discrete embedded natural language, recent studies encode images into quantized visual tokens using the VQ-VAE encoder [43, 11] and apply the generation process in an autoregressive [32, 56] or non-autoregressive manner [4, 35, 36, 5]. In particular, non-autoregressive generation shows a better trade-off between generation quality and speed. Nevertheless, they suffer from *compounding decoding errors* [36], which means that small decoding errors in early generation steps can accumulate into large differences in later steps. To address this issue, Token-Critic [35] and DPC [36] use additional transformers to identify more realistic tokens. However, they require a training second transformer, which incurs a training cost similar to training a generator from scratch.

The non-autoregressive predict-mask sampling of MGMs can be regarded as a discrete diffusion process that uses *absorbing state diffusion process* [1, 36]. Similar to MGMs, VQ-Diffusion [16] proposed a mask-and-replace diffusion strategy to predict masked tokens. Improved VQ-Diffusion [52] adopt purity-based sampling with discrete CFG to further improve the sample quality.

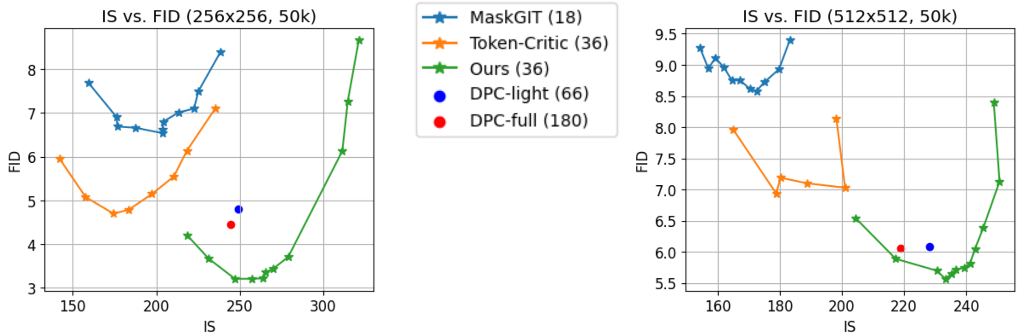


Figure 4: IS vs. FID curves of various sampling methods for MGMs on ImageNet 256×256 and 512×512. The curve positioned towards the bottom right indicates a better trade-off between sample quality and diversity. We plot the curve by varying the sampling temperature (τ), and the curves of MaskGIT [4] and Token-Critic [35] are taken from Token-Critic [35].

5 Experiments

Datasets, Baselines, and Metrics. We demonstrate the effectiveness of the proposed guidance for masked generative models on class conditional generation using the Imagenet benchmark [8] with 256×256 and 512×512 resolutions. For a baseline model, we use MaskGIT [4], which shows a state-of-the-art trade-off between quality and sampling speed on a class conditional generation of MGMs and has publicly available checkpoints for our target datasets. To evaluate the trade-off between sample fidelity and diversity, we measure Fréchet Inception Distance (FID) [18], Inception Score (IS) [46], Precision, and Recall [31] using the implementation provided by Dhariwal et al. [10]. To measure the computational cost, we report the number of function evaluations (NFE) required to sample an image. Note that total sampling timesteps (T) may differ from NFE due to guidance.

Implementation Details. We use a VQGAN tokenizer [11] provided by MaskGIT [4], which encodes images into 10-bit integers. Since the training code for MaskGIT is unavailable, we implement based on the open Pytorch reproduction [40]. We follow the previous work for error token correction [24] to prepare input with error tokens and randomly replace 30% of input tokens with error tokens. We utilized an NVIDIA RTX A6000 for fine-tuning and sampling. We used an exponential moving average (EMA) of fine-tuning weights with a decay of 0.9999 and `bf16` precision. The batch size was set to 256, and the additional parameters introduced by TOAST are approximately 20-25% of the model size. Notably, the fine-tuning was completed efficiently within 10 epochs. More detailed implementation of TOAST [49] is provided in the Appendix A. We use sampling step $T = 18$ and sampling temperature 25 for Imagenet 256×256 and 45 for Imagenet 512×512. For sampling step $T = 12$, we use temperatures 10 and 20, respectively for each resolution.

5.1 Comparison with Various Generative Models

Quantitative results. We compare the performance of the proposed method with various class conditional generative methods. (1) *GANs*: BigGAN [3] and GigaGAN [26]; (2) *continuous diffusion models* (Diff.): ADM [10], CDM [21], LDM [44], and DiT [41]; (3) *auto-regressive models* (AR): VQVAE-2 [43] and VQGAN [11]; (4) *discrete diffusion models* (Discrete.): VQ Diffusion [16], and Improved VQ Diffusion [52]; (5) *masked generative models* (Mask.): MaskGIT [4], Token-Critic [35], DPC [36]. As noted in previous literature [19, 36], sampling using a pre-trained classifier may impact the classifier-based metrics such as FID and IS. Thus, to see the base generative capacity of each method, we compare the models that do not use external pre-trained networks such as a classifier or upsampler during training or sampling. We also exclude large-scale models such as DiT-XL/2 [41] for a fair comparison. Table 1 presents the quantitative results of various generative models with guidance. All values are taken from the original paper unless otherwise noted. The proposed method achieves superior FID and IS despite the low computational cost for sampling. For instance, the proposed method achieves better FID and IS than DiT-L/2 with CFG on Imagenet 256×256 even though the proposed method requires an order of magnitude fewer sampling steps.

LDM (NFE=500, FID=3.60) MaskGIT (NFE=18, FID=6.56) Ours (NFE=36, FID=3.22)

Figure 5: Sampled images on ImageNet 256×256 class conditional generation using selected classes (`105`: Koala, `661`: model T, and `933`: Cheeseburger). left: LDM [44] + CFG (s=1.5, NFE=250×2), middle: MaskGIT (NFE=18), right: Ours (s=1.0, NFE= 18×2).

We comprehensively compare the proposed methods with various sampling strategies for MGMs in Fig. 4. We note that all MGMs use the same VQGAN tokenizer [11] provided in MaskGIT [4], the same baseline generator [4], and the same sampling timestep $T = 18$. Although the previous methods, such as Token-Critic [35] and DPC [36], require similar or more NFEs to sample images and more training resources to train an external corrector transformer, our simple guidance shows a better trade-off between sample quality and diversity.

Qualitative results. Fig. 5 presents the randomly sampled images using LDM [44] with CFG, MaskGIT, and MaskGIT with the proposed guidance. The proposed guidance sampling enhances details in the generated samples while maintaining high diversity. More sampled results with various classes are provided in the Fig. 1 and Appendix C.

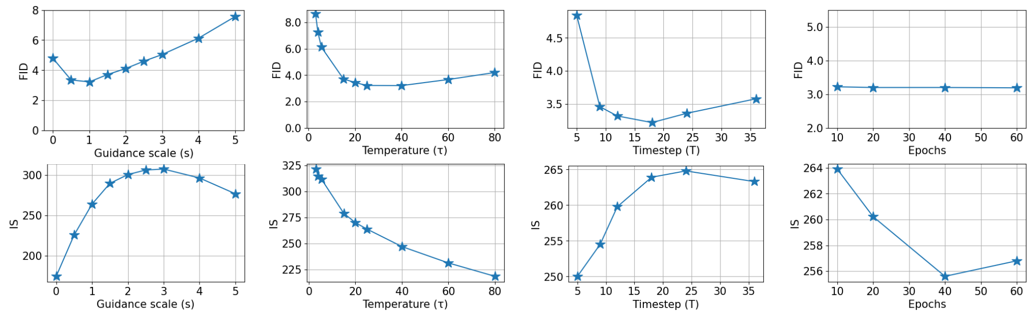
5.2 Ablation Studies and Analysis

In this section, we explore the effectiveness of the guidance on class conditional ImageNet generation in 256×256 scale across the various sampling hyperparameters. We vary each sampling parameter and otherwise use the default settings.

Table 2: Ablation results of various guidances on ImageNet 256×256 class conditional generation.

	FID↓	IS↑
Blur Guidance	8.32	231.2
SAG [22]	4.73	177.3
ft. w/ $\mathcal{L}_{mask}$	4.11	238.4
ft. w/ $\mathcal{L}_{aux}$ (Ours)	**3.22**	**263.9**

Effectiveness of Auxiliary Task. To demonstrate that fine-tuning with a proposed auxiliary task using the objective in Eq. (7) can effectively improve the generative capabilities of MGMs, we conduct ablation studies and report the FID and IS in Table 2. As noted in Section 3.2, applying Gaussian blur in the input VQ token does not properly smooth VQ tokens. However, we empirically found that applying Gaussian blur in the output logit can produce meaningful guidance. Therefore, we provide the quantitative comparison with guidance by applying Gaussian blur in the logit space (Blur Guidance). We further utilize self-attention value for adversarial masking following Lee et al. [22]. Blur guidance and SAG enhance the quality (IS) or diversity (FID) in some



(a) Guidance scale (b) Temperature (c) Timestep (d) Fine-tuning epochs

Figure 6: Exploring the sampling hyperparameters by varying (a) guidance scale, (b) sampling temperature, (c) sampling timesteps, and (d) fine-tuning epochs.

degree, but the improvement is marginal compared to ours. We further compare the results when the TOAST module is fine-tuned with the generative objective in Eq. (1) to verify the effectiveness of the auxiliary task (ft. w/ $\mathcal{L}_{mask}$). Since the TOAST architecture and blank input for the second stage naturally play the role of information bottleneck, the performance has increased. Nevertheless, fine-tuning with the proposed auxiliary loss in Eq. (7) demonstrates its effectiveness, showing superior performance in both metrics.

Varying the guidance scale (Fig. 6 a). In line with previous literature on sampling guidance [10, 19, 22], a high guidance scale improves the sample quality while sacrificing diversity. We found that the guidance scale 1.0 shows the best performance in terms of FID score and the best trade-offs. However, strong guidance ($s > 3$) does not ensure quality improvement, often leading to undesirably highlighted details or saturated colors (Appendix B), similar to the effect of large scale with CFG [19].

Varying the sampling temperature (Fig. 6 b). Compared to MaskGIT [4] limit their sampling temperature relatively low value ($\tau = 4.5$), we found that with the proposed guidance, the sample quality can be effectively preserved with a higher sampling temperature. As a result, with a high sampling temperature ($\tau = 25$), we achieve better quality and diversity compared to MaskGIT.

Varying the sampling steps T (Fig. 6 c). We observed that for higher T, the optimal sampling temperature also increases. For example, at $T = 5$, the best results are achieved with a sampling temperature of 4, while at $T = 18$, the optimal sampling temperature rises to 25. Thus, we plot the best FID and IS for each timestep using different temperatures. Whereas the optimal performance-efficiency trade-off of MaskGIT is observed around $T = 8$, ours shows such "sweet spot" around $T = 18$. Up to this point, both quality and diversity increase as T increases, consistently outperforming MaskGIT. Furthermore, with fewer NFEs ($T = 5$), the proposed method achieves an FID of 4.84 and an IS of 249.9, outperforming the MaskGIT samples with $T = 18$. This demonstrates that the proposed guidance technique provides more scalable and efficient sampling for MGMs.

Varying fine-tuning epochs (Fig. 6 d). We found that the proposed fine-tuning is efficiently trained within 10 epochs and that no further fine-tuning is required to achieve better results.

6 Conclusion and Future Work

In this paper, we define generalized guidance for discrete diffusion models, as a counterpart for guidance in continuous domain [22]. To generate guidance, we propose an auxiliary task to apply semantic smoothing in VQ tokens. Experimental results show that the proposed guidance effectively and efficiently improves the generative capabilities of MGMs on class conditional image generation.

Future work. Although the proposed guidance can improve the generative capabilities of MGMs on class conditional image generation, there is still room for improvement in various aspects: (1) experiments on large-scale text conditional generative models [5], (2) generalization to discrete diffusion models [16], and (3) generalization to various domains such as audio [63], and video [57].

Societal impacts. While our research does not directly touch ethical issues, the rapid growth of generative models raises considerations in AI ethics (e.g., potential misuse in creating deepfakes [34], generating antisocial content [12], or vulnerabilities to adversarial attacks [61, 17, 30]).

Acknowledgements

This work was partly supported by Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (RS-2024-00439020, Developing Sustainable, Real-Time Generative AI for Multimodal Interaction, SW Starlab), Artificial intelligence industrial convergence cluster development project funded by the Ministry of Science and ICT(MSIT, Korea)&Gwangju Metropolitan City, and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2023-00240379).

References

[1] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

[2] D. Berthelot, C. Raffel, A. Roy, and I. Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. *arXiv preprint arXiv:1807.07543*, 2018.

[3] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[4] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022.

[5] H. Chang, H. Zhang, J. Barber, A. Maschinot, J. Lezama, L. Jiang, M.-H. Yang, K. Murphy, W. T. Freeman, M. Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023.

[6] O. Chapelle, J. Weston, L. Bottou, and V. Vapnik. Vicinal risk minimization. *Advances in neural information processing systems*, 13, 2000.

[7] J. Choi, J. Lee, C. Shin, S. Kim, H. Kim, and S. Yoon. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11472–11481, 2022.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[10] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[11] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[12] R. Gandikota, J. Materzynska, J. Fiotto-Kaufman, and D. Bau. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2426–2436, 2023.

[13] M. Ghazvininejad, O. Levy, Y. Liu, and L. Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. *arXiv preprint arXiv:1904.09324*, 2019.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[15] J. Gu, J. Bradbury, C. Xiong, V. O. Li, and R. Socher. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*, 2017.

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

[17] G. Han, J. Choi, H. Lee, and J. Kim. Reinforcement learning-based black-box model inversion attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20504–20513, 2023.

[18] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[19] J. Ho and T. Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[20] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[21] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

[22] S. Hong, G. Lee, W. Jang, and S. Kim. Improving sample quality of diffusion models using self-attention guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7462–7471, 2023.

[23] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019.

[24] X. S. Huang, F. Perez, and M. Volkovs. Improving non-autoregressive translation models without distillation. In *International Conference on Learning Representations*, 2021.

[25] J. Hur, J. Choi, G. Han, D.-J. Lee, and J. Kim. Expanding expressiveness of diffusion models with limited data via self-distillation based fine-tuning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5028–5037, 2024.

[26] M. Kang, J.-Y. Zhu, R. Zhang, J. Park, E. Shechtman, S. Paris, and T. Park. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10124–10134, 2023.

[27] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[28] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila. Training generative adversarial networks with limited data. *Advances in neural information processing systems*, 33:12104–12114, 2020.

[29] D. Kingma, T. Salimans, B. Poole, and J. Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

[30] S. Koh, H. Shon, J. Lee, H. G. Hong, and J. Kim. Disposable transfer learning for selective source task unlearning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11752–11760, 2023.

[31] T. Kynkäänniemi, T. Karras, S. Laine, J. Lehtinen, and T. Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.

[32] D. Lee, C. Kim, S. Kim, M. Cho, and W.-S. Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11523–11532, 2022.

[33] D. Lee, J. Y. Lee, D. Kim, J. Choi, J. Yoo, and J. Kim. Fix the noise: Disentangling source feature for controllable domain translation. *arXiv preprint arXiv:2303.11545*, 2023.

[34] J. Lee, J. Hyung, S. Jeong, and J. Choo. Selfswapper: Self-supervised face swapping via shape agnostic masked autoencoder. *arXiv preprint arXiv:2402.07370*, 2024.

[35] J. Lezama, H. Chang, L. Jiang, and I. Essa. Improved masked image generation with token-critic. In *European Conference on Computer Vision*, pages 70–86. Springer, 2022.

[36] J. Lezama, T. Salimans, L. Jiang, H. Chang, J. Ho, and I. Essa. Discrete predictor-corrector diffusion models for image synthesis. In *The Eleventh International Conference on Learning Representations*, 2022.

[37] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

[38] A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

[39] M. Ott, M. Auli, D. Grangier, and M. Ranzato. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*, pages 3956–3965. PMLR, 2018.

[40] S. Patil, B. William, and P. von Platen. Amused: An open MUSE model. URL `https://github.com/huggingface/open-muse`.

[41] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[42] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[43] A. Razavi, A. Van den Oord, and O. Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

[44] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[45] S. Ryu. Low-rank adaptation for fast text-to-image diffusion fine-tuning, 2023.

[46] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[47] A. Sauer, K. Schwarz, and A. Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022.

[48] B. Shi, T. Darrell, and X. Wang. Top-down visual attention from analysis by synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2102–2112, 2023.

[49] B. Shi, S. Gai, T. Darrell, and X. Wang. Toast: Transfer learning via attention steering. *arXiv preprint arXiv:2305.15542*, 5(7):13, 2023.

[50] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[51] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[52] Z. Tang, S. Gu, J. Bao, D. Chen, and F. Wen. Improved vector quantized diffusion models. *arXiv preprint arXiv:2205.16007*, 2022.

[53] K. Tian, Y. Jiang, Z. Yuan, B. Peng, and L. Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*, 2024.

[54] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[55] E. Xie, L. Yao, H. Shi, Z. Liu, D. Zhou, Z. Liu, J. Li, and Z. Li. Difffit: Unlocking transferability of large diffusion models via simple parameter-efficient fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4230–4239, 2023.

[56] J. Yu, X. Li, J. Y. Koh, H. Zhang, R. Pang, J. Qin, A. Ku, Y. Xu, J. Baldridge, and Y. Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.

[57] L. Yu, Y. Cheng, K. Sohn, J. Lezama, H. Zhang, H. Chang, A. G. Hauptmann, M.-H. Yang, Y. Hao, I. Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10459–10469, 2023.

[58] E. B. Zaken, S. Ravfogel, and Y. Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, 2021.

[59] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[60] K. Zhang, R. Wang, X. Tan, J. Guo, Y. Ren, T. Qin, and T.-Y. Liu. A study of syntactic multi-modality in non-autoregressive machine translation. *arXiv preprint arXiv:2207.04206*, 2022.

[61] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 253–261, 2020.

[62] S. Zhao, H. Ren, A. Yuan, J. Song, N. Goodman, and S. Ermon. Bias and generalization in deep generative models: An empirical study. *Advances in Neural Information Processing Systems*, 31, 2018.

[63] A. Ziv, I. Gat, G. L. Lan, T. Remez, F. Kreuk, A. Défossez, J. Copet, G. Synnaeve, and Y. Adi. Masked audio generation using a single non-autoregressive transformer. *arXiv preprint arXiv:2401.04577*, 2024.

A Detailed Implementation of feature selection module (TOAST)

The architecture of the proposed sampling method consists of two parts: MaskGIT [4] and TOAST [49]. Since we have not changed the MaskGIT architecture for plug-and-play sampling guidance, we briefly explain the implementation details of TOAST below. The TOAST modules consist of three parts: token selection module, channel selection module, and linear feed-forward networks.

(i) The token selection module selects the task or class-relevant tokens by measuring the similarity with the learnable anchor vector ξ_c. We generate class conditional anchors ξ_c with simple class conditional MLPs.

(ii) The channel selection is applied with learnable linear transformation matrix $\mathbf{P}$. Then, the output of the token and channel selection module is calculated via $\boldsymbol{z}_i = \mathbf{P} \cdot sim(\boldsymbol{z}_i, \xi_c)$, where $\boldsymbol{z}_i$ denotes the i-th input token.

(iii) After the feature selection, the output is processed with L layer MLP layers, where L is equal to the number of Transformer's layers. The output of the l-th layer of MLP blocks is added to the value matrix of the attention block in $(L-l)$-th Transformer layer (top-down attention steering). Following the previous work [49], we add variational loss to regularize the top-down feedback path. A more detailed process and theoretical background can be found in previous works on top-down attention steering [48, 49].

B Effect of High Sample Temperature and Guidance Scale

We show the effect of high sampling temperature (τ) and high guidance scale in Fig. 7. The high sampling temperature (i.e., strong guidance) often leads to undesirably highlighted fine-scale details or saturated colors, similar to the large scale of CFG [19]. High temperatures often lead to the collapse of the overall structure of generated samples.

(a) s=1, τ=25 (b) s=5, τ=25 (c) s=1, τ=50

Figure 7: Sampled images on ImageNet 256×256 class conditional generation using (a) our default config, (b) large guidance scale ($s = 5$), and (c) high sampling temperature ($\tau = 50$).

C More Visual Results

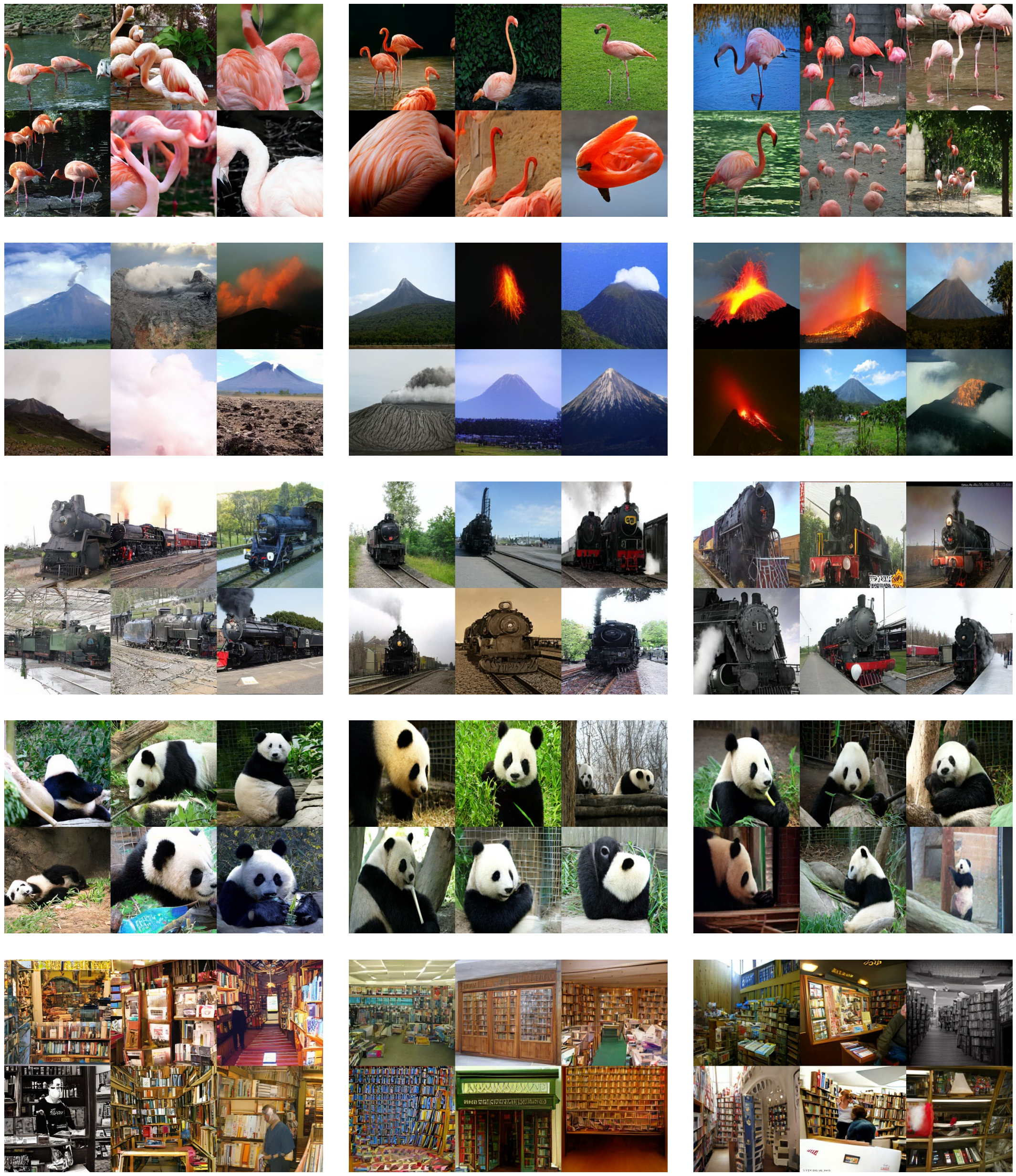


Figure 8: Sampled images on ImageNet 256×256 class conditional generation using selected classes (130: Flamingo, 980: Volcano, 820: Steam locomotive, 388: Giant panda, and 454: Bookshop). left: LDM [44] + CFG (s=1.5, NFE=250×2), middle: MaskGIT (NFE=18), right: Ours (s=1.0, NFE=18×2).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claim in the abstract and introduction is dealt with in the background, method, and experiment sections.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We deal with this in the conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide a correct proof for our theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide implementation details in the paper for reproducibility.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We do not include code for our experiments. Nevertheless, the dataset and baseline code for our implementation and evaluation are clearly denoted in the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide results by varying the hyperparameters and provide insights for selecting those values. We use open-source evaluation code, which is denoted in the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The sampling process and evaluation are statistical, but we do not report the error bars since with the sufficiently large number of samples (=50k), the impact of stochasticity is minimized.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the training environment, required training epochs, and required sampling steps in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We wrote a paper in compliance with the code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We denote the negative societal impacts that the proposed method may have.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not include such a large pre-trained model in the paper. We will release only a small module for fine-tuning, which does not have a high risk of misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use open source implementation for our baseline and denote in the paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We have no results related to the human evaluations.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We have no results related to the human evaluations.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.