Masked Diffusion Transformer is a Strong Image Synthesizer

Shanghua Gao[1,2] * Pan Zhou[2] † Ming-Ming Cheng[1] † Shuicheng Yan[2]
[1]Nankai University [2]Sea AI Lab
{shanghuagao,shuicheng.yan}@gmail.com zhoupan@sea.com cmm@nankai.edu.cn

Abstract

Despite its success in image synthesis, we observe that diffusion probabilistic models (DPMs) often lack contextual reasoning ability to learn the relations among object parts in an image, leading to a slow learning process. To solve this issue, we propose a Masked Diffusion Transformer (MDT) that introduces a mask latent modeling scheme to explicitly enhance the DPMs' ability to contextual relation learning among object semantic parts in an image. During training, MDT operates in the latent space to mask certain tokens. Then, an asymmetric masking diffusion transformer is designed to predict masked tokens from unmasked ones while maintaining the diffusion generation process. Our MDT can reconstruct the full information of an image from its incomplete contextual input, thus enabling it to learn the associated relations among image tokens. Experimental results show that MDT achieves superior image synthesis performance, e.g., a new SOTA FID score in the ImageNet data set, and has about 3× faster learning speed than the previous SOTA DiT. The source code is released at https://github.com/sail-sg/MDT.

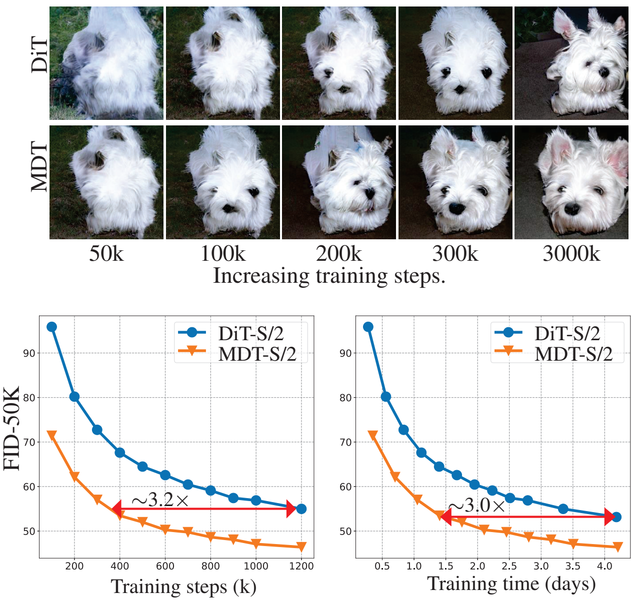

Figure 1. Top: Visual examples of MDT/DiT [35]. Down: learning progress comparison between DiT and MDT w.r.t. training steps/time on 8×A100 GPUs. MDT has about 3× faster learning speed than DiT while achieving superior FID scores.

1. Introduction

Diffusion probabilistic models (DPMs) [11, 38] have been at the forefront of recent advances in image-level generative models, often surpassing the previously state-of-the-art (SOTA) generative adversarial networks (GANs) [4, 17, 37, 54]. Additionally, DPMs have demonstrated their success in numerous other applications, including text-to-image generation [38] and speech generation [24]. DPMs adopt a time-inverted Stochastic Differential Equation (SDE) to gradually map a Gaussian noise into a sample by multiple time steps, with each step corresponding to a network evaluation. In practice, generating a sample is time-consuming due to the thousands of time steps required for the SDE to converge. To address this issue, various generation sampling strategies [21, 32, 41] have been proposed to accelerate the inference speed. Nevertheless, improving the training speed of DPMs is less explored but highly desired. Training of DPMs also unavoidably requires a large number of time steps to ensure the convergence of SDEs, making it very computationally expensive, especially in this era where large-scale models [11, 35] and data [9, 14, 44] are often used to improve generation performance.

In this work, we first observe that DPMs often struggle to learn the associated relations among object parts in an image. This leads to its slow learning process during training. Specifically, as illustrated in Fig. 1, the classical DPM, DDPM [21] with DiT [35] as the backbone, has learned the shape of a dog at the 50k-th training step, then learns its one eye and mouth until at the 200k-th step while missing another eye. Also, the relative position of two ears is not very accurate, even at the 300k-th step. This learning process reveals that DPMs fail to learn the associated relations

*This work was done while S. Gao was a research intern at Sea AI Lab.
†Pan Zhou and Ming-Ming Cheng are joint corresponding authors.

among semantic parts and independently learn each semantic part. The reason behind this phenomenon is that DPMs maximize the log probability of real data by minimizing the per-pixel prediction loss, which ignores the associated relations among object parts in an image, thus resulting in their slow learning progress.

Inspired by the above observation, we propose an effective Masked Diffusion Transformer (MDT) to improve the training efficiency of DPMs. MDT proposes a mask latent modeling scheme designed for transformer-based DPMs to explicitly enhance contextual learning ability and improve the associated relation learning among semantic parts in an image. Specifically, following [35, 38], MDT operates the diffusion process in the latent space to save computational costs. It masks certain image tokens and designs an asymmetric diffusion transformer structure to predict masked tokens from unmasked ones in a diffusion generation manner. To this end, the asymmetric structure contains an encoder, a side-interpolater, and a decoder. The encoder and decoder modify the transformer block in DiT [35] by inserting global and local token position information to help predict masked tokens. The encoder only processes unmasked tokens during training handling all tokens during inference, as there are no masks. So to ensure the decoder always processes all tokens for training prediction or inference generation, a side-interpolater implemented by a small network aims to predict masked tokens from encoder output during training, and it is removed during inference.

With the masking latent modeling scheme, our MDT can reconstruct the full information of an image from its contextual incomplete input, learning the associated relations among semantic parts in an image. As shown in Fig. 1, MDT typically generates two eyes of the dog at almost the same training steps, indicating that it correctly learns the associated semantics of an image by utilizing the mask latent modeling scheme. In contrast, DiT [35] cannot easily synthesize a dog with the correct semantic part relations. This comparison shows MDT's superior relation modeling and faster learning ability over DiT. Experimental results demonstrate that MDT achieves superior performance on the image synthesis task, and set the new SOTA on class-conditional image synthesis on the ImageNet dataset, as shown in Fig. 2 and Tab. 2. MDT also enjoys about 3× faster learning progress during training than the SOTA DPMs, namely DiT, as demonstrated by Fig. 1 and Tab. 1. We hope our work can inspire more work on speeding up the diffusion training process with unified representation learning.

The main contributions are summarised as follows:

- By introducing an effective mask latent modeling scheme, we proposed a masked diffusion transformer method, which, for the first time, explicitly enhances the contextual learning ability of DPMs.

Figure 2. Visualization of images generated by the MDT-XL/2.

- Experiments show that our method better synthesis images while using much less training time than SOTA.

2. Related works

2.1. Diffusion Probabilistic models

Diffusion probabilistic model (DPM) [11, 21], also known as score-based model [48, 49], is a competitive image synthesis approach. DPMs begin by using an evolving Stochastic Differential Equation (SDE) to gradually add Gaussian noise into real data, transforming a complex data distribution into a Gaussian distribution. Then, it adopts a time-inverted SDE to map a Gaussian noise gradually into a sample by multiple steps. At each sampling time step, a network is utilized to generate the sample along the gradient of the log probability, also known as the score function [50]. The iterative nature of diffusion models can result in high training and inference costs. Efficient sampling strategies [21, 23, 32, 41, 46], latent space diffusion [38, 51], and multi-resolution cascaded generation [22] have been proposed to reduce the inference cost. Additionally, some training schemes [2, 12] are introduced to improve the diffusion model training, *e.g.*, approximate maximum likelihood training [28, 34, 47], and training loss weighting [26, 27]. We identify the lack of contextual modeling ability in diffusion models. To address this, we propose the mask latent modeling scheme to enhance the contextual representation of diffusion models, which is orthogonal to existing diffusion training schemes.

2.2. Networks for Diffusion Models

The UNet-like [39] network, enhanced by spatial self-attention [42, 52] and group normalization [55] is firstly used for diffusion models [21]. Several design improvements, *e.g.*, adding more attention heads [18], BigGAN [4] residual block, and adaptive group normalization, are proposed in [11] to further enhance the generation ability of the UNet.

Recently, due to the broad applicability of transformer networks, several works have attempted to utilize the vision transformer (ViT) structure for diffusion models [1, 35, 56]. GenViT [56] demonstrates that ViT is capable of image generation but has inferior performance compared to UNet. U-ViT [1] improves ViT by adding long-skip connections and convolutional layers, achieving competitive performance with that of UNet. DiT [35] verifies the scaling ability of ViT on large model sizes and feature resolutions. Our MDT is orthogonal to these diffusion networks as it focuses on contextual representation learning. Moreover, the position-aware designs in MDT reveal that the mask latent modeling scheme benefits from a stronger diffusion network. We will explore further to release the potential of these networks with MDT.

2.3. Mask Modeling

Mask modeling has been proven to be effective in both recognition learning [8, 10, 15, 19] and generative modeling [7, 36]. In the natural language processing (NLP) field, mask modeling was first introduced to enable representation pretraining [10, 36] and language generation [5, 25]. Subsequently, it also proved feasible for vision recognition [3] and generation [7, 16, 57] tasks. In vision recognition, pre-training schemes that utilize mask modeling enable good representation quality [58], scalability [19] and faster convergence [15]. In generative modeling, following the bi-directional generative modeling in NLP, MaskGIT [7] and MUSE [6] use the masked generative transformer to predict randomly masked image tokens for image generation. Similarly, VQ-Diffusion [16] presents a mask-replace diffusion strategy to generate images. In contrast, our MDT aims to enhance the contextual representation of the denoising diffusion transformer [35] with mask latent modeling. This preserves the detail refinement ability of denoising diffusion models by maintaining the diffusion process during inference. To ensure that the mask latent modeling in MDT focuses on representation learning instead of reconstruction, we propose an asymmetrical structure in mask modeling training. As an extra benefit, it enables lower training costs than masked generative models because it skips the masked patches in training instead of replacing masked input patches with a mask token.

3. Masked Diffusion Transformer

Revisit Diffusion Probabilistic Model. For diffusion probabilistic models [11, 45], such as DDPM [21] and DDIM [46], training involves a forward noising process and a reverse denoising process. In the forward noising process, Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is gradually added to the real sample x_0 via a discrete SDE formulation [46]. If the time step t is large, x_t would be a Gaussian noise. Similarly, the reverse denoising process is a discrete SDE that gradually maps a Gaussian noise into a sample. At each time step, given x_t, it predicts the next reverse step $p_\theta(x_{t-1}|x_t)$ via a network. Following [34, 35], the network is trained by optimizing the variational lower-bound L_{vlb} of the log-likelihood $p_\theta(x_0)$ [45]. During inference, one can sample a Gaussian noise and then gradually reverses to a sample x_0. Same as [34, 35], we train the diffusion model conditioned with class label c, *i.e.* $p_\theta(x_{t-1}|x_t, c)$. By default, we use class-conditioned image generation in our experiments.

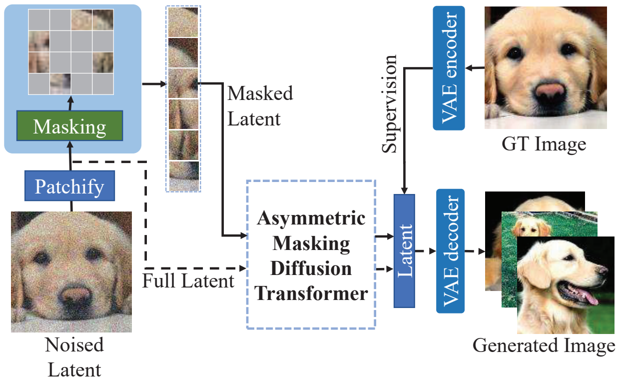

Figure 3. The overall framework of Masked Diffusion Transformer (MDT). Solid/dotted lines indicate each time step's training/inference process. Masking and side-interpolater are only used during training and are removed during inference.

3.1. Overview

As shown in Fig. 1, DPMs with DiT backbone exhibit slow training convergence due to the slowly learning of the associated relations among semantic parts in an image. To relieve this issue, we propose Masked Diffusion Transformer (MDT), which introduces a mask latent modeling scheme to enhance contextual learning ability explicitly. To this end, as depicted in Fig. 3, MDT consists of 1) a latent masking operation to mask the input image in the latent space, and 2) an asymmetric masking diffusion transformer structure that performs vanilla diffusion process as DPMs, but with masked input. To reduce computational costs, MDT follows LatentDiffusion [38] to perform generative learning in the latent space instead of raw pixel space.

In the training phase, MDT first encodes an image into a latent space with a pre-trained VAE encoder [38]. Then, MDT adds Gaussian noise into the image latent embedding. The latent masking operation in MDT then patchifies the resulting noisy latent embedding into a sequence of tokens, and masks certain tokens. The remaining unmasked tokens are fed into the asymmetric masking diffusion transformer, as shown in Fig. 4(a), which contains an encoder, a side-interpolater, and a decoder to predict the masked tokens from the unmasked ones. During inference, MDT replaces the side-interpolater with additional position embedding. MDT takes the latent embedding of a Gaussian noise as input to generate the denoised latent embedding, which is then passed to a pre-trained VAE decoder [38] for image generation.

The above masking latent modeling scheme in the training phase forces the diffusion model to reconstruct the full information of an image from its contextual incomplete input. Thus, the model is encouraged to learn the relations among image latent tokens, particularly the associated relations among semantic parts in an image. For example, as illustrated in Fig. 3, the model should first understand the correct associated relations among small image parts (tokens) of the dog image. Then, it should generate the masked "eye" tokens using other unmasked tokens as contextual information. Furthermore, Fig. 1 shows that MDT often learns to generate the associated semantics of an image at nearly the same pace, such as the generation of the dog's two eyes (two ears) at almost the same training step. While DiT [35] (DDPM with transformer backbone) learns to generate one eye (one ear) initially and then learns to generate another eye (ear) after roughly 100k training steps. This demonstrates the superior learning ability of MDT over DiT in terms of the associated relation learning of image semantics.

In the following parts, we will introduce the two key components of MDT, 1) a latent masking operation, and 2) an asymmetric masking diffusion transformer structure.

3.2. Latent Masking

Following the Latent diffusion model (LDM) [38], MDT performs generation learning in the latent space instead of raw pixel space to reduce computational costs. In the following, we briefly recall LDM and then introduce our latent masking operation on the latent input.

Latent diffusion model (LDM). LDM employs a pre-trained VAE encoder $\mathbf{E}$ to encode an image $v \in \mathbb{R}^{3\times H\times W}$ to a latent embedding $z = \mathbf{E}(v) \in \mathbb{R}^{c\times h\times w}$. It gradually adds noise to z in the forward process and then denoises to predict z in the reverse process. Finally, LDM uses a pre-trained VAE decoder $\mathbf{D}$ to decode z into a high-resolution image $v = \mathbf{D}(z)$. Both the VAE encoder and decoder are fixed during training and inference. Since h and w are smaller than H and W, performing the diffusion process in the low-resolution latent space is more efficient than pixel space.

Latent masking operation. We first add Gaussian noise to an image's latent embedding z during training. Then following [35], we divide the noisy embedding z into a sequence of $p \times p$-sized tokens and concatenate them to a matrix $u \in \mathbb{R}^{d\times N}$, where d is the channel number and N is the number of tokens. Next, we randomly mask tokens with a ratio ρ and concatenate the remaining tokens as $\hat{u} \in \mathbb{R}^{d\times \hat{N}}$, where $\hat{N} = \rho N$. Accordingly, we can create a binary mask $M \in \mathbb{R}^N$ in which one (zero) denotes the masked (unmasked) tokens. Finally, we feed the tokens $\hat{u}$

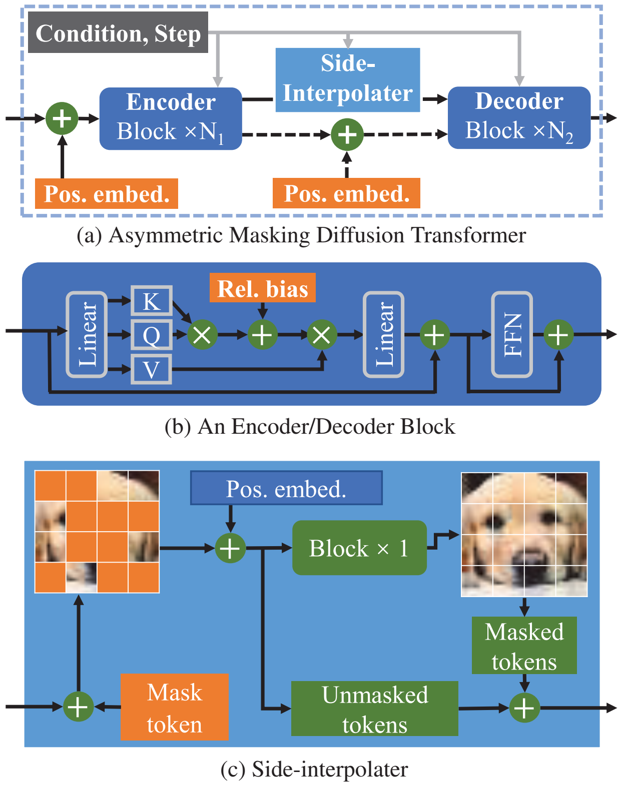

Figure 4. The asymmetric masking diffusion transformer structure in MDT. We modify the DiT [35] by adding a side-interpolater, local relative positional bias, and learnable global position embeddings. The conditional scheme is omitted for simplicity.

into our diffusion model for processing. We only use tokens $\hat{u}$ for two reasons. 1) The model should focus on learning semantics instead of predicting the masked tokens. As shown in Sec. 4.3, it achieves better performance than replacing the masked tokens with a learnable mask token and then processing all tokens like [3, 6, 7]; 2) It saves the training cost compared to processing all N tokens.

3.3. Asymmetric Masking Diffusion Transformer

We introduce our asymmetric masking diffusion transformer for joint training of mask latent modeling and diffusion process. As shown in Fig. 4(a), it consists of three components: an encoder, a side-interpolater, and a decoder, each of which is described in detail below.

Position-aware encoder and decoder. In MDT, predicting the masked latent tokens from the unmasked tokens requires the position relations of all tokens. To enhance the position information in the model, we propose a positional-aware encoder and decoder that facilitate the learning of the masked latent tokens. Specifically, the encoder and decoder tailor the standard DiT block via adding two types of token position information containing N_1 and N_2 blocks, as illustrated in Fig. 4(a).

Firstly, the encoder adds the conventional learnable global position embedding into the noisy latent embedding input. Similarly, the decoder introduces the learnable position embedding into its input but with different training and inference phase approaches. The side-interpolater already uses the learnable global position embedding introduced below during training. During inference, since the side interpolater is discarded (see below), the decoder explicitly adds the position embedding into its input to enhance positional information.

Secondly, as depicted in Fig. 4(b), the encoder and decoder add a local relative positional bias [30] to each head in each block when computing the attention score of the self-attention [53]:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + B_r\right)V,$$

where Q, K, and V, respectively denote the query, key, and value in the self-attention module, d_k is the dimension of the key, and $B_r \in \mathbb{R}^{N\times N}$ is the relative positional bias. B_r is selected by the relative positional difference between the i-th position and other positions, which is updated during training. The local relative positional bias helps to capture the relative relations among tokens, facilitating the masking of latent modeling.

The encoder takes the unmasked noisy latent embedding provided by our latent masking operation and feeds its output into the side-interpolater/decoder during training/inference. For the decoder, its input is the output of the side-interpolater for training or the combination of the encoder output and the learnable position embedding for inference. During training, the encoder and decoder, respectively, handle unmasked tokens and full tokens. Thus, we name our model the "asymmetric" model.

Side-interpolater. As shown in Fig. 3, during training, for efficiency and better performance, the encoder only processes the unmasked tokens $\hat{u}$. While in the inference phase, the encoder handles all tokens u due to the lack of masks. This means that there is a big difference in the encoder output (*i.e.*, decoder input) during training and inference, at least in terms of token number. To ensure the decoder always processes all tokens for training prediction or inference generation, a side-interpolater implemented by a small network aims to predict masked tokens from encoder output during training and would be removed during inference.

In the training phase, the encoder processes the unmasked tokens to obtain its output token embedding $\hat{q} \in \mathbb{R}^{d\times\hat{N}}$. Then, as shown in Fig. 4(c), the side-interpolater first fills the masked positions, indicated by the mask M defined in Sec. 3.2, by a shared learnable mask token, and also adds a learnable positional embedding to obtain an embedding $q \in \mathbb{R}^{d\times N}$. Next, we use a basic encoder block to process q to predict an interpolated embedding $\hat{k}$. The $\hat{k}$ tokens denote the predicted tokens. Finally, we use a masked shortcut connection to combine prediction $\hat{k}$ and q as $k = M\cdot q + (1-M)\cdot\hat{k}$. In summary, for masked tokens, we use the prediction by side-interpolater. For unmasked tokens, we still adopt the corresponding tokens in q. This can 1) boost the consistency between training and inference phases, 2) eliminates the mask-reconstruction process in the decoder.

Since there are no masks during inference, the side-interpolater is replaced by a position embedding operation, which adds the learnable position embeddings of the side-interpolater, learned during training. This ensures the decoder always processes all tokens and uses the same learnable position embeddings for training prediction or inference generation, thus having better image generation performance.

3.4. Training

During training, we feed both full latent embedding u and the masked latent embedding $\hat{u}$ to the diffusion model. We observe that only using masked latent embedding makes the model focus too much on masked region reconstruction, while ignoring the diffusion training. Full/masked latent inputs are independently sent to the network, and the training objectives optimize the variational lower-bound, as in [34, 35]. Due to the asymmetrical masking structure, the extra costs for using masked latent embedding are small. This is also demonstrated by Fig. 1, which shows that MDT still achieves about 3× faster learning progress than previous SOTA DiT in total training hours.

4. Experiments

4.1. Implementation

We give the implementation details of MDT, including model architecture, training details, and evaluation metrics.

Model architecture. We follow DiT [35] to set the total block number (*i.e.*, $N_1 + N_2$), token number, and channel numbers of the diffusion transformer of MDT. As DiT reveals stronger synthesis performance when using a smaller patch size, we also use a patch size p=2 by default, denoted by MDT-/2. Moreover, We also follow DiT's parameters to design MDT for getting its small-, base-, and xlarge-sized model, denoted by MDT-S/B/XL. Same as LatentDiffusion [38] and DiT, MDT adopts the fixed VAE[1] provided by the Stable Diffusion to encode/decode the image/latent tokens by default. The VAE encoder has a downsampling ratio of $1/8$, and a feature channel dimension of 4, *e.g.*, an image

[1]The model is downloaded in https://huggingface.co/stabilityai/sd-vae-ft-mse

Method	Image Res.	Training Steps (k)	FID-50K↓
DiT-S/2	256×256	400	68.40
MDT-S/2	256×256	300	57.01
MDT-S/2	256×256	400	53.46
MDT-S/2	256×256	2000	44.14
MDT-S/2	256×256	3500	**41.37**
DiT-B/2	256×256	400	43.47
MDT-B/2	256×256	400	34.33
MDT-B/2	256×256	3500	**20.45**
DiT-XL/2	256×256	400	19.47
DiT-XL/2	256×256	2352	10.67
DiT-XL/2	256×256	7000	9.62
MDT-XL/2	256×256	400	16.42
MDT-XL/2	256×256	**1300**	**9.60**
MDT-XL/2	256×256	3500	**6.65**

Table 1. Comparison between DiT [35] and MDT under different model sizes and training steps on ImageNet 256×256. DiT results are obtained from DiT reported results.

of size 256×256×3 is encoded into a latent embedding of size 32×32×4.

Training details. Following [35], all models are trained by AdamW [31] optimizer of 3e-4 learning rate, 256 batch size, and without weight decay on ImageNet [9] with an image resolution of 256×256. We set the mask ratio as 0.3 and $N_2 = 2$. Following the training settings in DiT, we set the maximum step in training to 1000 and use the linear variance schedule with a range from 10^{-4} to 2×10^{-2}. Other settings are also aligned with DiT.

Evaluation. We evaluate models with commonly used metrics, *i.e*. Fre'chet Inception Distance (FID) [20], sFID [33], Inception Score (IS) [40], Precision and Recall [29]. The FID is used as the major metric as it measures both diversity and fidelity. sFID improves upon FID by evaluating at the spatial level. As a complement, IS and Precision are used for measuring fidelity, and Recall is used to measure diversity. For fair comparisons, we follow [35] to use the TensorFlow evaluation suite from ADM [11] and report FID-50K with 250 DDPM sampling steps. Unless specified otherwise, we report the FID scores without the classifier-free guidance [23].

4.2. Comparison Results

Performance comparison. Tab. 1 compares our MDT with the SOTA DiT under different model sizes. It is evident that MDT achieves higher FID scores for all model scales with fewer training costs. The parameters and inference cost of MDTs are similar to DiT, since the extra modules in MDT are negligible as introduced in Sec. 3.1. For small models, MDT-S/2 trained with 300k steps outperforms the DiT-S/2 trained with 400k steps by a large margin on FID (57.01 vs. 68.40). More importantly, MDT-S/2 trained with 2000k steps achieves similar performance with a larger model DiT-B/2 trained with a similar computational budget. For the largest model, MDT-XL/2 trained with 1300k steps outperforms DiT-XL/2 trained with 7000k steps on FID (9.60 vs. 9.62), achieving about 5× faster training progress.

Method	Cost(Iter×BS)	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
DCTrans. [33]	-	36.51	-	-	0.36	0.67
VQVAE-2 [37]	-	31.11	-	-	0.36	0.57
VQGAN [13]	-	15.78	78.3	-	-	-
BigGAN-deep [4]	-	6.95	7.36	171.4	0.87	0.28
StyleGAN [43]	-	2.30	4.02	265.12	0.78	0.53
Impr. DDPM [34]	-	12.26	-	-	0.70	0.62
MaskGIT [7]	1387k×256	6.18	-	182.1	0.80	0.51
CDM [22]	-	4.88	-	158.71	-	-
ADM [11]	1980k×256	10.94	6.02	100.98	0.69	0.63
LDM-8 [38]	4800k×64	15.51	-	79.03	0.65	0.63
LDM-4	178k×1200	10.56	-	103.49	0.71	0.62
DiT-XL/2 [35]	7000k×256	9.62	6.85	121.50	0.67	0.67
MDT	2500k×256	7.41	4.95	121.22	0.72	0.64
MDT	3500k×256	6.46	**4.92**	131.70	**0.72**	0.63
MDT	6500k×256	**6.23**	5.23	**143.02**	0.71	0.65
ADM-G [11]	1980k×256	4.59	5.25	186.70	0.82	0.52
ADM-G, U	1980k×256	3.94	6.14	215.84	0.83	0.53
LDM-8-G [38]	4800k×64	7.76	-	209.52	0.84	0.35
LDM-4-G	178k×1200	3.60	-	247.67	0.87	0.48
U-ViT-G [1]	300k×1024	3.40	-	-	-	-
DiT-XL/2-G [35]	7000k×256	2.27	4.60	278.24	0.83	0.57
MDT-G	2500k×256	2.15	4.52	249.27	0.82	0.58
MDT-G	3500k×256	2.02	**4.46**	263.77	0.82	0.60
MDT-G	6500k×256	**1.79**	4.57	**283.01**	0.81	**0.61**

Table 2. Comparison with existing methods on class-conditional image generation with the ImageNet 256×256 dataset. -G denotes the results with classifier-free guidance [23]. Results of MDT-XL/2 model are given for comparison. Compared results are obtained from their papers.

We also compare the class-conditional image generation performance of MDT with existing methods in Tab. 2. To make fair comparisons with DiT, we also use the EMA weights of VAE decoder in this table. Under class-conditional settings, MDT with half training iterations outperforms DiT by a large margin, *e.g*., 6.83 vs 9.62 in FID. Following previous works [1, 11, 35, 38], we utilize an improved classifier-free guidance [23] with a power-cosine weight scaling to trade off between precision and recall during class-conditional sampling. MDT achieves superior performance over previous SOTA DiT and other methods with the FID score of 1.81, *setting a new SOTA for class-conditional image generation*. Similar to DiT, we never

Mask Ratio	FID↓	sFID↓	IS↑	Precision↑	Recall↑
0.1	51.60	10.23	26.65	0.44	0.60
0.2	51.44	10.09	26.75	0.44	0.58
0.3	**50.26**	**10.08**	**27.61**	**0.45**	0.60
0.4	50.88	10.21	27.44	0.45	0.60
0.5	51.57	9.92	27.14	0.44	0.60
0.6	53.20	10.36	26.55	0.44	0.61
0.7	52.90	10.03	26.51	0.44	0.61
0.8	53.73	10.15	25.55	0.43	**0.61**

Table 3. Effect of different masking ratios. MDT-S/2 trained with 600k iterations.

Decoder pos.	FID↓	sFID↓	IS↑	Precision↑	Recall↑
Last0	51.05	9.97	27.31	0.44	0.60
Last1	50.96	**9.90**	**27.63**	0.45	0.60
Last2	**50.26**	10.08	27.61	**0.45**	**0.60**
Last4	51.67	10.12	26.91	0.45	0.60
Last6	52.64	10.36	26.46	0.44	0.60

Table 4. Effect of position of side-interpolater. MDT-S/2 models contain 12 blocks and are trained with 600k iterations.

observe the model has saturated FID scores when continuing training.

Convergence speed. Fig. 1 compares the performance of the DiT/S-2 baseline and MDT/S-2 under different training steps and training time on 8×A100 GPUs. Because of the stronger contextual learning ability, MDT achieves better performance with faster generation learning speed. MDT enjoys about 3× faster learning speed in terms of both training steps and training time. For example, MDT-S/2 trained with about 33 hours (400k steps) achieves superior performance than DiT-S/2 trained with about 100 hours (1500k steps). This reveals that contextual learning is vital for faster generation learning of diffusion models.

4.3. Ablation

In this part, we conduct ablation to verify the designs in MDT. We report the results of MDT-S/2 model and use FID-50k as the evaluation metric unless otherwise stated.

Masking ratio. The masking ratio determines the number of input patches that can be processed during training. We give the comparison of using different masking ratios in Tab. 3. The best masking ratio for MDT-S/2 is 30%, which is quite different from the masking ratio used for recognition models, e.g. 75% masking ratio in MAE [19]. We assume that the image generation requires learning more details from more patches for high-quality synthesis, while recognition models only need the most essential patches to infer semantics.

Side-interpolater position. To meet the high-quality image generation requirements of the diffusion model, the side-interpolater is placed in the middle of the network instead of the end of the network in recognition models [3, 19]. Tab. 4, presents the comparison of placing the side-interpolater at different positions of the MDT-S model with 12 blocks. The results show that placing the side-interpolater before the last two blocks achieves the best FID score, whereas placing it at the end of the network like recognition models impairs the performance. Placing the side-interpolater at the early stages of the network also harm the performance, indicating the mask latent modeling is beneficial to most stages in the diffusion models.

Asymmetric vs. Symmetric architecture in masking. Unlike the masked generation works [6, 7], *e.g.*, MaskGIT, that utilize the masking scheme to generate images, MDT focuses on improving diffusion models with contextual learning ability via the masking latent modeling. Therefore, we use an asymmetric architecture to only process the unmasked tokens in the diffusion model encoder. We compare the asymmetric architecture in MDT and the symmetrical architecture [7] that processes full input with masked tokens replaced by a learnable mask token. As shown in Tab. 5a, the asymmetric architecture in MDT has an FID of 50.26, outperforming the FID of 51.56 achieved by the symmetric architecture. The asymmetric architecture further reduces the training cost and allows the diffusion model to focus on learning contextual information instead of reconstructing masked tokens.

Effect of side-interpolater. The side-interpolater in MDT predicts the masked tokens, allowing the diffusion model to learn more semantics and maintain consistency in decoder inputs during training and inference. We compare the performance with/without the side-interpolater in Tab. 5b, and observe a gain of 1.34 in FID when using the side-interpolater, proving its effectiveness.

Masked shortcut in side-interpolater. The masked shortcut ensures that the side-interpolater only predicts the masked tokens from unmasked ones. Tab. 5c shows that using the masked shortcut enhances the FID from 50.91 to 50.26, indicating that restricting side-interpolater to only predict masked tokens helps the diffusion model achieve stronger performance.

Full and masked latent tokens. In MDT, both the full and masked latent embeddings are fed into the diffusion model during training. In comparison, we give the results trained by only using full/masked latent embeddings as shown in Tab. 5d, where the computational cost is aligned for

Asymmetric stru.	FID-50k↓
×	51.56
✓	**50.26**

(a) Effect of asymmetric masking structure.

Side-interpolater	FID-50k↓
×	51.60
✓	**50.26**

(b) Effect of side-interpolater.

Masked shortcut	FID-50k↓
×	50.91
✓	**50.26**

(c) Effect of masked shortcut.

Latent type	FID-50k↓
Full+Masked	**50.26**
Full	52.30
Masked	76.63

(d) Using full/masked latent under aligned cost.

Sup. parts	FID-50k↓
All	**50.26**
Masked	58.35

(e) Supervision on token parts.

Number	FID-50k↓
1	**50.26**
2	51.77
3	51.96

(f) Number of blocks in side-interpolator.

IS Pos. embed.	FID-50k↓
×	51.58
✓	**50.26**

(g) Effect of positional embeddings in SI.

Learnable pos.	FID-50k↓
×	50.80
✓	**50.26**

(h) Effect of learnable positional embeddings.

Relative pos. bias	FID-50k↓
×	53.56
✓	**50.26**

(i) Effect of relative positional bias.

Table 5. Ablation study on MDT-S/2. Models are trained for 600k iterations.

fair comparisons. Trained with both full and masked latent leads to clear gain over two competitors. While using only the masked latent embeddings results in slow convergence, which we attribute to the training/inference inconsistency as the inference in MDT is a diffusion process instead of the masked reconstruction process.

Loss on all tokens. By default, we calculate the loss on both masked and unmasked latent embeddings. In comparison, mask modeling for recognition models commonly calculates loss on masked tokens [3, 19]. Tab. 5e shows that calculating the loss on all tokens is much better than on masked tokens. We assume that this is because generative models require stronger consistency among patches than recognition models do, since details are vital for high-quality image synthesis.

Block number in side-interpolater. We compare the performance of different numbers of blocks in the side-interpolater in Tab. 5f. The default setting of 1 block achieves the best performance, and the FID worsens with an increase in block number. This result is consistent with our motivation that side-interpolater should not learn too much information other than interpolating the masked representations.

Positional-aware enhancement. To further release the potential of mask latent modeling, we enhance the DiT baseline with stronger positional awareness ability, *i.e.* learnable positional embeddings and the relative positional bias in basic blocks. Tab. 5g shows the positional embeddings in side-interpolater improves the FID from 51.58 to 50.26, indicating the positional embedding is vital for the side-interpolater. Also, enables the training of positional embeddings brings the gain in FID as revealed in Tab. 5h. In Tab. 5i, the relative positional bias in the basic blocks significantly improves the FID from 53.56 to 50.26, showing the relative positional modeling ability is essential for diffusion models to obtain the contextual representation ability and generate high-quality images. Therefore, the positional awareness ability in diffusion model structure is required to accompany the masked latent modeling, playing a key role in improving performance.

5. Conclusion

This work proposes a masked diffusion transformer to enhance the contextual representation and improve the relation learning among image semantics for DPMs. We introduce an effective mask latent modeling scheme into DPMs and also accordingly designs an asymmetric masking diffusion transformer structure. Experiments show that our masked diffusion transformer enjoys higher performance on image synthesis and largely improves the learning progress during training, achieving the new SOTA for image synthesis on the ImageNet dataset. We hope our initial exploration of contextual learning for generative modeling can facilitate more work on unifying the representation learning for recognition and generative models.

Acknowledgement.

This research was supported by the NSFC (NO. 62225604) and the Fundamental Research Funds for the Central Universities (Nankai University, 070-63233089). The Supercomputing Center of Nankai University supports computation.

References

[1] Fan Bao, Chongxuan Li, Yue Cao, and Jun Zhu. All are worth words: a vit backbone for score-based diffusion models. *arXiv preprint arXiv:2209.12152*, 2022. 3, 6

[2] Fan Bao, Chongxuan Li, Jiacheng Sun, Jun Zhu, and Bo Zhang. Estimating the optimal covariance with imperfect mean in diffusion probabilistic models. In *Int. Mach. Learn.*, 2022. 2

[3] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *Int. Conf. Learn. Represent.*, 2022. 3, 4, 7, 8

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *Int. Conf. Learn. Represent.*, 2019. 1, 2, 6

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Adv. Neural Inform. Process. Syst.*, 33:1877–1901, 2020. 3

[6] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023. 3, 4, 7

[7] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11315–11325, 2022. 3, 4, 6, 7

[8] Gong Cheng, Ruimin Li, Chunbo Lang, and Junwei Han. Task-wise attention guided part complementary learning for few-shot image classification. *Science China Information Sciences*, 64:1–14, 2021. 3

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009. 1, 6

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL*, 2019. 3

[11] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Adv. Neural Inform. Process. Syst.*, 34:8780–8794, 2021. 1, 2, 3, 6

[12] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped langevin diffusion. In *Int. Conf. Learn. Represent.*, 2022. 2

[13] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12873–12883, 2021. 6

[14] Shanghua Gao, Zhong-Yu Li, Ming-Hsuan Yang, Ming-Ming Cheng, Junwei Han, and Philip Torr. Large-scale unsupervised semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022. 1

[15] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Towards sustainable self-supervised learning. *arXiv preprint arXiv:2210.11016*, 2022. 3

[16] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10696–10706, 2022. 3

[17] Yuchao Gu, Xintao Wang, Yixiao Ge, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Rethinking the objectives of vector-quantized tokenizers for image synthesis. *arXiv preprint arXiv:2212.03185*, 2022. 1

[18] Meng-Hao Guo, Tian-Xing Xu, Jiang-Jiang Liu, Zheng-Ning Liu, Peng-Tao Jiang, Tai-Jiang Mu, Song-Hai Zhang, Ralph R. Martin, Ming-Ming Cheng, and Shi-Min Hu. Attention mechanisms in computer vision: A survey. *Computational Visual Media*, 8(3):331–368, 2022. 2

[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 16000–16009, 2022. 3, 7, 8

[20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Adv. Neural Inform. Process. Syst.*, 30, 2017. 6

[21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Adv. Neural Inform. Process. Syst.*, 33:6840–6851, 2020. 1, 2, 3

[22] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23(47):1–33, 2022. 2, 6

[23] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *NeurIPS Workshop*, 2021. 2, 6

[24] Myeonghun Jeong, Hyeongju Kim, Sung Jun Cheon, Byoung Jin Choi, and Nam Soo Kim. Diff-tts: A denoising diffusion model for text-to-speech. In *INTERSPEECH*, 2021. 1

[25] Ge-Peng Ji, Mingchen Zhuge, Dehong Gao, Deng-Ping Fan, Christos Sakaridis, and Luc Van Gool. Masked vision-language transformer in fashion. *Machine Intelligence Research*, 20(3):421–434, 2023. 3

[26] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Adv. Neural Inform. Process. Syst.*, 2022. 2

[27] Dongjun Kim, Seungjae Shin, Kyungwoo Song, Wanmo Kang, and Il-Chul Moon. Soft truncation: A universal training technique of score-based diffusion model for high precision score estimation. In *Int. Mach. Learn.*, 2022. 2

[28] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Adv. Neural Inform. Process. Syst.*, 34:21696–21707, 2021. 2

[29] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Adv. Neural Inform. Process. Syst.*, 32, 2019. 6

[30] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Int. Conf. Comput. Vis.*, pages 10012–10022, 2021. 5

[31] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *Int. Conf. Learn. Represent.*, 2019. 6

[32] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Adv. Neural Inform. Process. Syst.*, 2022. 1, 2

[33] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. *Int. Mach. Learn.*, 2021. 6

[34] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Int. Mach. Learn.*, pages 8162–8171. PMLR, 2021. 2, 3, 5, 6

[35] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 1, 2, 3, 4, 5, 6

[36] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training, 2018. 3

[37] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Adv. Neural Inform. Process. Syst.*, 32, 2019. 1, 6

[38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10684–10695, 2022. 1, 2, 3, 4, 5, 6

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Int. Conf. Med Image Comput Comput Assist Interv.*, pages 234–241. Springer, 2015. 2

[40] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Adv. Neural Inform. Process. Syst.*, 29, 2016. 6

[41] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *Int. Conf. Learn. Represent.*, 2022. 1, 2

[42] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. In *Int. Conf. Learn. Represent.*, 2017. 2

[43] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022. 6

[44] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Adv. Neural Inform. Process. Syst.*, 2022. 1

[45] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Int. Mach. Learn.*, pages 2256–2265. PMLR, 2015. 3

[46] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Int. Conf. Learn. Represent.*, 2021. 2, 3

[47] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Adv. Neural Inform. Process. Syst.*, 34:1415–1428, 2021. 2

[48] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Adv. Neural Inform. Process. Syst.*, 32, 2019. 2

[49] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Adv. Neural Inform. Process. Syst.*, 33:12438–12448, 2020. 2

[50] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *Int. Conf. Learn. Represent.*, 2021. 2

[51] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. In *Adv. Neural Inform. Process. Syst.*, 2021. 2

[52] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Adv. Neural Inform. Process. Syst.*, 29, 2016. 2

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Adv. Neural Inform. Process. Syst.*, 30, 2017. 5

[54] Yan Wu, Jeff Donahue, David Balduzzi, Karen Simonyan, and Timothy Lillicrap. Logan: Latent optimisation for generative adversarial networks. *Int. Conf. Learn. Represent.*, 2020. 1

[55] Yuxin Wu and Kaiming He. Group normalization. In *Eur. Conf. Comput. Vis.*, pages 3–19, 2018. 2

[56] Xiulong Yang, Sheng-Min Shih, Yinlin Fu, Xiaoting Zhao, and Shihao Ji. Your vit is secretly a hybrid discriminative-generative diffusion model. *arXiv preprint arXiv:2208.07791*, 2022. 3

[57] Zhu Zhang, Jianxin Ma, Chang Zhou, Rui Men, Zhikang Li, Ming Ding, Jie Tang, Jingren Zhou, and Hongxia Yang. M6-ufc: Unifying multi-modal controls for conditional image synthesis. *Adv. Neural Inform. Process. Syst.*, 2021. 3

[58] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *Int. Conf. Learn. Represent.*, 2022. 3