LMD: Faster Image Reconstruction with Latent Masking Diffusion

Zhiyuan Ma[1], **Zhihuan Yu**[2], **Jianjun Li**[2*], **Bowen Zhou**[1*]

[1]Department of Electronic Engineering, Tsinghua University
[2]School of Computer Science and Technology, Huazhong University of Science and Technology
mzyth@tsinghua.edu.cn

Abstract

As a class of fruitful approaches, diffusion probabilistic models (DPMs) have shown excellent advantages in high-resolution image reconstruction. On the other hand, masked autoencoders (MAEs), as popular self-supervised vision learners, have demonstrated simpler and more effective image reconstruction and transfer capabilities on downstream tasks. However, they all require extremely high training costs, either due to inherent high temporal-dependence (i.e., excessively long diffusion steps) or due to artificially low spatial-dependence (i.e., human-formulated high mask ratio, such as 0.75). To the end, this paper presents LMD, a simple but faster image reconstruction framework with **L**atent **M**asking **D**iffusion. First, we propose to project and reconstruct images in latent space through a pre-trained variational autoencoder, which is theoretically more efficient than in the pixel-based space. Then, we combine the advantages of MAEs and DPMs to design a progressive masking diffusion model, which gradually increases the masking proportion by three different schedulers and reconstructs the latent features from simple to difficult, without sequentially performing denoising diffusion as in DPMs or using fixed high masking ratio as in MAEs, so as to alleviate the high training time-consumption predicament. Our approach allows for learning high-capacity models and accelerate their training (by 3× or more) and barely reduces the original accuracy. Inference speed in downstream tasks also significantly outperforms the previous approaches.

	Pixel-Level (w/o Latent)	Spatial-Dependency (w/o Mask)	Temporal-Dependency (w/o Diffusion)	Training Time -Consuming
DPMs	Yes (↑)	Strong (↓)	Strong (↑)	High (↑)
MAEs	Yes (↑)	Weak (↑)	Weak (↓)	High (↑)
LMD (Ours)	No (↓)	Strong (↓)	Weak (↓)	Low (↓)

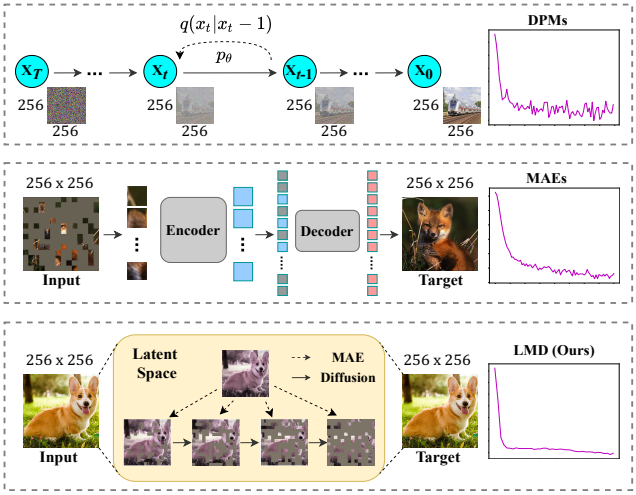


Figure 1: An example to illustrate the comparisons between our proposed LMD versus DPMs and MAEs on training time-consumption.

1 Introduction

Image reconstruction is one of the most challenging and computationally expensive tasks in computer vision. Recent years, diffusion probabilistic models (DPMs) (Ho, Jain, and Abbeel 2020) based on Markov probability prediction have gained prominence in various generative models (Rombach et al. 2022; Saharia et al. 2022; Ma et al. 2023a), due to their excellent performance in the diversity and high-fidelity of image synthesis. Subsequently, a large number of works based on DPMs have been proposed and great progresses have been achieved in terms of sampling procedure (Liu et al. 2022b), conditional guidance (Nichol et al. 2021), likelihood maximization (Kim et al. 2022) and generalization ability (Gu et al. 2022). However, almost all existing DPMs face an inherent time-consuming dilemma (e.g., *150-1000 V100 days in ADM* (Dhariwal and Nichol 2021)), because the Markov diffusion process in DPMs requires a very large number of diffusion steps (e.g., *thousands*) and sampling time, which can be optimized but is basically inevitable for effective synthesis (Cao et al. 2022; Croitoru et al. 2022). Meanwhile, masked image modeling, as a simpler and more efficient self-supervised learning technique, has been introduced to the CV fields to develop a widespread Vision Transformer (ViT) (Dosovitskiy et al. 2020), which is a non-autoregressive milestone model and has recently sparked the interest of many researchers (Zhang et al. 2022; Ma et al. 2022a). With the great success of ViT, a series of masked autoencoders (MAEs) have been proposed (He et al. 2022; Wei et al. 2022; Li et al. 2022a). These MAEs are widely viewed as efficient visual learners, and can fully utilize the paral-

*Corresponding authors.

lel computing capability of GPU to learn multi-head visual features and reconstruct images well at pixel-level. Compared with DPMs, MAEs seem to be more acceptable, due to their higher inference efficiency, wider generalization performance (Ma et al. 2023b) and lower theoretical threshold. However, they also seem to be somewhat simple and crude by mandatory setting the mask proportion as a fixed number (e.g., 75%), which makes the training of MAEs less elegant as compared to the progressive ways in DPMs. A large number of experiments (Huang et al. 2022; Liu et al. 2022a; Chen et al. 2022) also prove that the training of MAEs is very time-consuming (e.g., *112 V100 days for MAE* (He et al. 2022)), which is even close to that of DPMs.

To sum up, DPMs and MAEs have made significant contributions in their respective technical routes for image reconstruction. However, they both share a fatal problem, *high training time-consumption*. Through an in-depth analysis of these two categories of models, we attribute this problem to two factors: (1) *Pixel-level modeling*. **Firstly**, they almost all model image features at the pixel-level, which naturally requires a lot of time to encode (or add-noise) and decode (or de-noise) the image features in the entire pixel space. Taking the 256×256 image input in Figure 1 as an example, for DPMs, each diffusion step requires Gaussian sampling and noise prediction in the 256×256 pixel space, which greatly reduces its training efficiency. For MAEs, the pixel size of its input image determines the number of patches (i.e., *the input length of the ViT*) when the same patch size is maintained. Therefore, for both DPMs and MAEs, modeling at pixel-level is one of the significant causes of high time-consumption. (2) *Spatial-temporal dependency*. **Secondly**, their training efficiency is implicitly limited by their inherent or artificially spatial-temporal dependency. As illustrated in the table in Figure 1, for DPMs, each Markov diffusion step is calculated based on the previous step, so the training time theoretically is positively correlated with its temporal-dependency, that is, stronger time dependence (i.e., *longer diffusion steps*) will lead to higher training time-consumption. On the contrary, the training time seems to be negatively correlated with the spatial-dependency, that is, in each training iteration, the de-noising optimization process from $\mathbf{X}_t$ to $\mathbf{X}_{t-1}$ is obviously easier and the time-consumption is shorter than that directly from $\mathbf{X}_t$ to $\mathbf{X}_0$. Therefore, for MAEs, each iteration from 75% masked $\mathbf{X}_t$ to unmasked $\mathbf{X}_0$ will naturally require higher time-consumption for training.

Driven by the above two aspects, we propose a simple but well-considered latent masking diffusion framework for faster image reconstruction. Specifically, to address the first factor, inspired by the success of Stable Diffusion, we employ a pre-trained variational autoencoder to compress the input image from the original pixel space to a latent space with smaller scale for latent destruction and reconstruction. But different from Stable Diffusion, we then follow the MAEs' architecture to split the latent feature map into small patches to compose a patch sequence with length l as the readout. Based on the latent patch sequence, we further address the second factor by designing a progressive masking diffusion strategy, which gradually increases the masking proportion by mask schedulers and restores the latent features from simple to difficult. Among them, three different schedulers are adopted to avoid temporal-dependency for faster parallel computing and enhance spatial-dependency to help model reconstruct efficiently, so as to ultimately reduce the total training time-consumption.

Experiments on two representative datasets ImageNet-1K and Lsun-Bedrooms demonstrate the effectiveness of the proposed LMD model, showing that it achieves competitive performance against previous DPMs or MAEs models, but with significantly lower mean training time-consumption. The inference speed of LMD in image reconstruction also significantly outperforms the previous approaches. Moreover, LMD can be well generalized to a variety of downstream tasks, due to its flexible architecture.

2 Related Work

Diffusion Probabilistic Models (DPMs). Recent years has witnessed the remarkable success of DPMs, due to its impressive generative capabilities. After surpassing GAN on image synthesis (Dhariwal and Nichol 2021), diffusion models have shown a promising algorithm and emerged as the new state-of-the-art generative (Yang et al. 2022) and editing (Ma, Jia, and Zhou 2023) models. As a pioneering work, **DDPM** (Ho, Jain, and Abbeel 2020) still suffers from slow sampling procedure and sub-optimal log-likelihood estimations. To this end, **DDIM** (Song, Meng, and Ermon 2020) proposes a more efficient sampling procedure to accelerate the forward process and has been widely used in subsequent DPMs. Later, a brand new **ADM** (Dhariwal and Nichol 2021) model emerges and leads the trend of guided diffusion models (Liu et al. 2021a; Nichol et al. 2021) After that, a series of large DPMs (Saharia et al. 2022; Rombach et al. 2022; Yu et al. 2022; Ramesh et al. 2022) with billions of parameters have been proposed and have attracted the attention of a large number of researchers.

Masked Auto-Encoders (MAEs) Different from DPMs' route, as a series of efficient self-supervised visual learners, MAEs (Li et al. 2021; Zhou et al. 2021; Ma et al. 2022b; Zhang et al. 2022; Ma et al. 2022c) commit to pre-train a generalized representation models by *mask-then-predict* pixels. Among them, **MAE** (He et al. 2022) is one of the most representative models, which proposes an asymmetric encoder-decoder architecture to feed those visible patches (about 25%) into encoder and reconstructs the image by predicting the remaining 75% patches. Subsequently, **SimMIM** proposes a simpler framework without the special designs (e.g., *block-wise masking* and *tokenization* via discrete VAE) to perform masked image modeling for addressing the data-hungry issue faced by large-scale model training. Moreover, to train the hierarchical models faster and reduce the GPU memory consumption, **GreenMIM** (Huang et al. 2022) designs an optimal grouping strategy based on dynamic programming and couplings it with sparse convolution into MAEs, which enjoys a training-speed advantage in hierarchical ViT training, such as Swin Transformer (Liu et al. 2021b) and Twins Transformer (Chu et al. 2021).

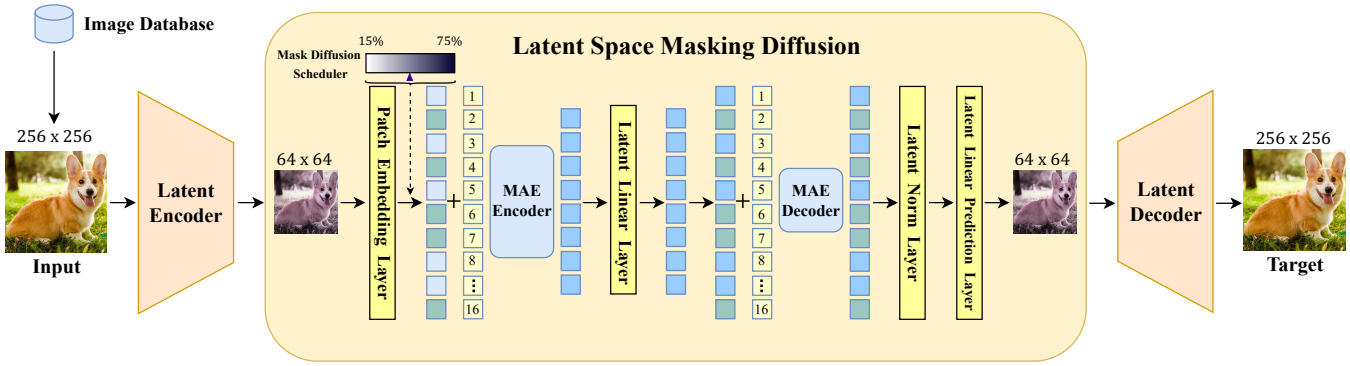


Figure 2: The Proposed Framework.

Compress-based Models. Compress-based image reconstruction models (Ramesh et al. 2021; Vahdat, Kreis, and Kautz 2021; Kim et al. 2022) aim to compress image into a smaller latent space for training acceleration, which is another line of research relevant to our work. **VQVAE** (Van Den Oord, Vinyals et al. 2017) proposes a simple yet powerful generative model to learn latent discrete representations by introducing vector quantization operation into VAEs and has shown powerful image compression capabilities. Based on the technique, **VQGAN** (Esser, Rombach, and Ommer 2021) further models images as a composition of perceptually rich image constituents and introduces adversarial training for better image reconstruction. Recently, **Stable Diffusion** (Rombach et al. 2022) has become one of the most sought-after diffusion models among researchers, due to its excellent text-to-image synthesis performance. Though achieving remarkable progress, these compress-based models are basically built on top of the DPMs and therefore still suffer from the inherently strong temporal-dependence of the Markov diffusion, e.g., Stable Diffusion is still very computationally expensive if trained from scratch.

Motivated by these works, our **LMD** model focuses on unifying the **L**atent space project technique, the **M**ask self-supervised technique and the **D**iffusion generative idea, then respectively leverages these techniques to reduce the dimension of the input image, avoid temporal-dependence for parallel computing, and enhance spatial-dependence for faster training, which eventually reduces the total training time-consumption for faster image reconstruction.

3 Methodology

To lower the computational demands and training time-consumption towards high-resolution image reconstruction, we propose a novel latent masking diffusion (LMD) framework, which integrates progressive mask self-supervised strategies into an encoder-decoder framework, as depicted in Figure 2. LMD mainly contains two steps: 1) Perceptual Latent Space Projection (Sec. 3.1) and 2) Latent Space Masking Diffusion (Sec. 3.2). The former aims to pretrain a VAE-based latent space projector to compress input images into a perceptually high-dimensional space for acceleration, while the latter aims to conduct latent masking diffusion procedure for more efficient image reconstruction.

We observe that the mask autoencoders provide high GPU parallel computing efficiency due to using ViT as the visual learner, but there are still two problems: (i) Almost all existing MAEs are modeled at the pixel-level. Though the computing cost can be reduced by assigning a higher pixel proportion to each patch, it will greatly damage the global-perceptibility of semantic elements. (ii) The current MAEs follow a high-proportion masking strategy for training, which will greatly impact the spatial-dependence among pixels and make the training unstable and the time-consumption longer. A better way may be to gradually increase the mask proportion to make full use of the spatial-dependency for acceleration. We now introduce the details of the two steps in LMD.

3.1 Perceptual Latent Space Projection

The latent space projection step is performed to compress the input images into a perceptual high-dimensional space by leveraging a pretrained latent space projector (LSP) based on previous work (Esser, Rombach, and Ommer 2021; Rombach et al. 2022). The LSP consists of an encoder $\mathcal{E}$, a decoder $\mathcal{G}$, a discriminator $\mathcal{D}$, and a learnable latent codebook $\mathcal{Z}$. Given an input image $x \in \mathbb{R}^{H\times W\times 3}$, LSP first compress the image x into a latent variable $\hat{z}$ by encoder $\mathcal{E}$, i.e., $\hat{z} = \mathcal{E}(x)$ and $\hat{z} \in \mathbb{R}^{h\times w\times d}$, where h and w respectively denote scaled height and width (scaled factor $f = H/h = W/w$), and d is the dimensionality of the compressed latent variable. After going through the step described in Sec. 3.2, the latent variable $\hat{z}$ is updated and finally reconstructed into $\hat{x}$ by decoder $\mathcal{G}$. Formally,

$$\hat{x} = \mathcal{G}_\theta(\text{LSMD}_\phi(\mathcal{E}_\theta(x))), \tag{1}$$

where LSMD$(\cdot)$ represents subsequent latent space masking diffusion step, ϕ denotes the parameters of LSMD, and θ denotes the parameters of LSP that are first pretrained and then frozen to use in the LSMD stage.

Vector Quantization As illustrated in Figure 3, vector quantization in the pre-training stage of LSP aims to map the aforementioned compressed latent variable $\hat{z}$ into a perceptual latent variable z_q. A learnable codebook $\mathcal{Z} = \{z_k\}_{k=1}^{K} \subset \mathbb{R}^d$, which has been pre-trained by (Van

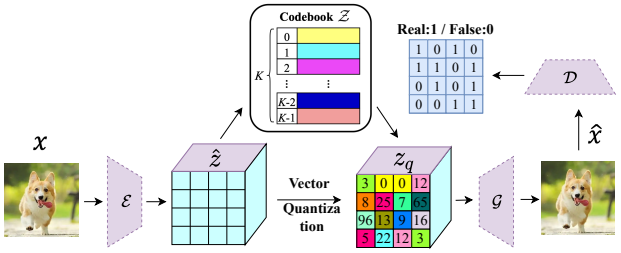


Figure 3: The Latent Space Projector.

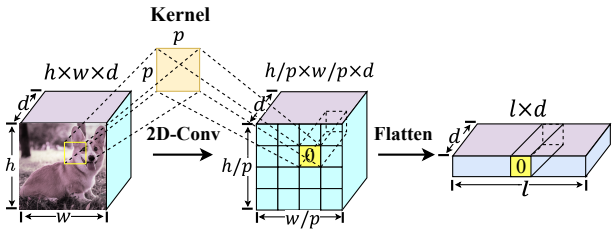


Figure 4: The Patch Embedding Layer.

Den Oord, Vinyals et al. 2017), is introduced to help LSP learn the perceptual latent feature of the image constituents.

Specifically, the codebook can be viewed as a discrete latent space of size K that is leveraged to express the relatively complete and perceptual semantic elements of the original image constituents (e.g., dog's eyes or tongue), and can be retrieved by $\hat{z}$ to obtain a latent variable z_q using element-wise vector quantization function $\mathbf{q}(\cdot)$, i.e., $z_q = \mathbf{q}(\hat{z})$, which is the key to compressing an image with little loss of accuracy. More precisely, the vector quantization function $\mathbf{q}(\cdot)$ aims to obtain a perceptual discrete representation z_q of the input image by using $\hat{z}$ to perform nearest neighbour look-up over latent codebook $\mathcal{Z}$, as follows,

$$\mathbf{q}(\hat{z}) := \left(\arg\min_{z_k \in \mathcal{Z}} ||\hat{z}_{ij} - z_k|| \right) \in \mathbb{R}^{h \times w \times d}. \quad (2)$$

In sum, the above forward process in the LSP stage can be formally described as,

$$\hat{x} = \mathcal{G}_\theta(\mathbf{q}_\theta(\mathcal{E}_\theta(x))). \quad (3)$$

Note the operations in Formula (2) are non-differentiable, so the above forward process cannot be directly optimized due to the gradient of the $\mathbf{q}(\cdot)$ cannot be backpropagated. Following (Esser, Rombach, and Ommer 2021), we adopt a straight-through gradient estimator to copy the gradients from the decoder to the encoder for end-to-end training via the loss function $\mathcal{L}_{\text{VQ}}$:

$$\begin{aligned}\mathcal{L}_{\text{VQ}}(\mathcal{E}, \mathcal{G}, \mathcal{Z}) = ||x - \hat{x}||^2 + \quad & ||\text{sg}[\mathcal{E}(x)] - z_q||_2^2 \\ & + \beta||\text{sg}[z_q] - \mathcal{E}(x)]||_2^2,\end{aligned} \quad (4)$$

where the first term is reconstruction loss between x and $\hat{x}$, the middle term is vector quantization loss between encoded vector $\hat{z}$ and retrieved vector z_q, here sg[$\cdot$] stands for stop-gradient operator and is used to solely update the z_q part of the codebook. The last term is the commitment loss designed to ensure the encoder commits to an embedding and its output does not grow. To keep consistent with (Van Den Oord, Vinyals et al. 2017), the commit factor β is set to 0.25.

Adversarial training To make the LSP more robust, adversarial training is also introduced into our work. As mentioned above, the decoder $\mathcal{G}$ is leveraged to reconstruct the latent variable z_q into $\hat{x}$. After that, a patch-based discriminator is introduced to accept the real image patch from x or the reconstructed image patch from $\hat{x}$ and give out a $(1, 0)$ judgment, which is trained via an adversarial loss $\mathcal{L}_{\text{ADV}}$:

$$\mathcal{L}_{\text{ADV}}(\{\mathcal{E}, \mathcal{Z}, \mathcal{G}\}, \mathcal{D}) = -[log(1 - \mathcal{D}(\hat{x})) + \log \mathcal{D}(x)). \quad (5)$$

The total objective for finding the optimal latent projector is:

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{VQ}}(\mathcal{E}, \mathcal{G}, \mathcal{Z}) + \gamma \mathcal{L}_{\text{ADV}}(\{\mathcal{E}, \mathcal{Z}, \mathcal{G}\}, \mathcal{D}), \quad (6)$$

where γ is an adaptive weight hyperparameter. With the patch-level adversarial training, LSP can be well generalized to the subsequent LSMD step, which is also trained on the same patch-level.

3.2 Latent Space Masking Diffusion

The latent space masking diffusion step (i.e., LSMD$_\phi(\cdot)$) follows an encoder-decoder architecture and is designed to achieve progressive masking diffusion by the following three components.

Latent Encoder To learn the deeper semantics of the compressed latent variables $\hat{z}$ of the image x, we sequentially perform a pipeline, which includes patch embedding layer, mask scheduling layer, spatial position embedding layer and MAE encoder blocks, for training.

The patch embedding layer aims to further embed the compressed latent variable $\hat{z} \in \mathbb{R}^{h \times w \times d}$ into patched-based latent variable $\hat{z}_p \in \mathbb{R}^{l \times d}$ (l is the number of the patches, and the size of each patch is $p \times p$ pixels), as illustrated in Figure 4. Specifically, we first feed the compressed latent variable $\hat{z}$ into a 2D-convolution layer to perform the convolution operation, where the kernel size and stride are both set to p, and then obtain the final patch-based representation $\hat{z}_p$ through a flatten operation. Note that $\hat{z}_p$ is a latent vector with length l, which is treated as the representation of a visual sequence and served as input to the subsequent visual Transformer.

The mask scheduling layer is designed to produce an increasing mask-ratio sequence by using a masking diffusion scheduler, which will be detailed in Sec. 3.2. Based such a mask-ratio sequence, LMD will progressively increase the mask proportion of patches in $\hat{z}_p$ with the increase of the training step, and finally only read out the unmasked patches for encoding.

Since the self-attention mechanism in Transformer is not sensitive to position, and the 1D position-features of tokens in NLP are ineffective for 2D data of image, we propose to use a spatial position embedding layer to learn the 2D-features and integrate them into latent variable $\hat{z}_p$ for obtaining better spatial-aware latent vector representations. Specifically, we first convert the index of each patch in $\hat{z}_p$ (see Figure 2) into their 2D coordinates (c_x, c_y) by dividing the index number into h/p (or w/p) to get the quotient as c_x and the

remainder as c_y. Here c_x and c_y both are integers and record spatial position of the current patch. Then, we respectively embed them into a $\sin$ - $\cos$ space via SPE$(\cdot)$,

$$\text{SPE}(i,j) = \begin{cases} \sin(\frac{i}{10000^{\frac{2i}{d}}}), & \text{if } i \text{ is even} \\ \cos(\frac{i}{10000^{\frac{2i-2}{d}}}), & \text{otherwise} \end{cases} \tag{7}$$

to obtain a 2D position vector $(\hat{z}_x, \hat{z}_y)$ corresponding to latent variable $\hat{z}_p$, where $i \in (1, l)$ and $j \in (1, d/2)$ respectively denotes patch and dimension index. Note here $\hat{z}_x \in \mathbb{R}^{\frac{d}{2}}$ and $\hat{z}_y \in \mathbb{R}^{\frac{d}{2}}$. Finally, we can obtain the updated spatial-aware vector $\hat{z}_p \in \mathbb{R}^d$, by first concatenating $\hat{z}_x$ and $\hat{z}_y$ and then adding it to previous $\hat{z}_p$.

Finally, a battery of MAE Encoder Blocks are adopted to learn attentive representations for better mining the deeper sematics of compressed images. Similar to ViT-base, we adopt 12 layers of Transformer as MAE encoder blocks, and define each Transformer block by a block function $f_{\text{block}}(\cdot)$ as,

$$\hat{z}_{\text{out}}^{(\ell)} = f_{\text{block}}^{(\ell)}(\hat{z}_{\text{out}}^{(\ell-1)}) \tag{8}$$

$$\hat{z}_{\text{out}}^{(0)} = \hat{z}_p \tag{9}$$

where ℓ is layer index, and $\hat{z}_{\text{out}}^{(12)}$ is the output of the last layer. Note that in $\hat{z}_p$, only unmasked patches are fed into the above MAE encoder blocks for image reconstruction.

Latent Decoder As opposed to the above encoder, the decoder is designed to reconstruct the compressed latent variable $\hat{z} \in \mathbb{R}^{h \times w \times d}$ by using the output $\hat{z}_{\text{out}}$ of the encoder as input. The pipeline of the decoder consists of a latent linear layer, a spatial position recall layer, MAE decoder blocks and latent normalization and prediction layers, and is also executed sequentially. Wherein, the latent linear layer and asymmetric MAE decoder blocks are adopted to given extra consideration for trade off between discriminant and generative tasks, the spatial position recall layer keeps the same operation as in the encoder but with negative SPE$(\cdot)$, and then the normalization and prediction layer is used for latent image reconstruction.

1) Trade off between discriminant tasks and generative tasks. In the original MAE (He et al. 2022), the asymmetric structure is adopted by setting up a heavier encoder (e.g., *12 blocks*) and a lighter decoder (e.g., *8 blocks*), which is more suitable for discriminant tasks. We consider to take a trade off between discriminant tasks and generative tasks, and additionally propose to use a lighter encoder (e.g., *8 blocks*) and a heavier decoder (e.g., *12 blocks*) for generative tasks such as image synthesis. Moreover, a latent linear layer is also preferentially placed at the beginning of the decoder for deeper decoder embedding (e.g., embed *512-dim* to *1024-dim* for decoding).

2) Latent Image Reconstruction. Different from the previous MAEs (He et al. 2022; Xie et al. 2022b), the latent image reconstruction (LIR) target of LMD is established in a latent space. The last layer of the decoder is a latent linear prediction layer whose number of output channels equals the dimension d of $\hat{z}$, which is used to predict the reconstructed latent image $\hat{z}_{\text{rec}}$. LMD adopts the mean squared error (MSE) between the reconstructed latent variable $\hat{z}_{\text{rec}}$ and original latent variable $\hat{z}$ of the compressed image as training target,

$$\mathcal{L}_{\text{LIR},\phi} = ||\hat{z} - \hat{z}_{\text{rec}}||_2^2, \tag{10}$$

here ϕ denote the parameters of the whole LSMD step. Similar to MAE, the loss is only computed on masked patches.

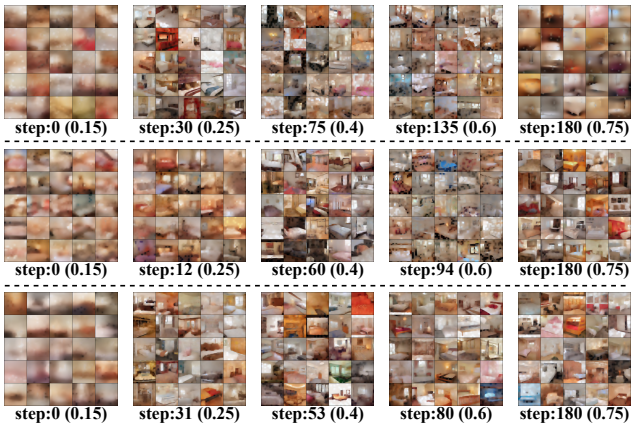


Figure 5: Mask diffusion examples under three different scheduling schemes: Uniform, Piecewise, and Cosine (from top to bottom). Note the 5×5 images from the LSUN-bedrooms with a maximum of 180k training steps in latent space.

Masking Diffusion Scheduler The masking diffusion scheduler aims to produce an increasing mask-ratio sequence for dynamically fitting the model training, so as to optimize the overall training time-consumption. Motivated by the lower temporal-dependence of MAEs and higher spatial-dependence of DPMs, we stand on the shoulder of mask self-supervise and diffusion generative techniques to propose the masking diffusion strategy, which is achieved through the following three scheduling schemes[1].

1) Uniform Scheduling. This uniform scheduling is a pre-explored scheme, which follows the assumption that the difficulty of model training decreases uniformly with the increase of the number of training steps. In the training stage, we randomly sampled 5×5 latent images for evaluation, as shown in Figure 5. From Figure 5, we can observe that when the mask ratio grows to 0.4, the masking speed of the uniform scheduler exceeds the reconstructing speed of the model. Moreover, it can be noticed that when the mask ratio reaches 0.75, this phenomenon is further aggravated.

2) Piecewise Scheduling. The previous scheme shows that the model capability is not uniformly improved with the growth of the mask ratio. To the end, we provide the piecewise scheduling scheme. Specifically, for the first 1/6 training steps, we increase the mask ratio linearly from 0.15 and 0.4, for the next 1/6 training steps, we maintain the mask

[1]In all schemes, the mask ratio of randomly masking is preset in [0.15-0.75], which follows the discrete optimal lower bound in BERT (Kenton and Toutanova 2019) and the continuous optimal upper bound in MAE (He et al. 2022).

ImageNet-1K (IN1K)							
Method	**Backbone**	**Image Size**	**Patch Size**	**Mask Ratio**	**MIT ↓**	**MLT ↓**	**MLI ↓**
MAE (2022)	ViT-B (12/8 blocks)	224 x 224	16 x 16	0.75	2.62	17.11	6.53
SimMIM (2022b)	Swin-B (2/2/18/2 blocks)	192 x 192	32 x 32	0.6	3.45	20.01	5.80
GreenMIM (2022)	Swin-B (2/2/18/2 blocks)	224 x 224	4 x 4	0.75	2.23	11.93	5.35
UM-MAE (2022b)	ViT-B (12/8 blocks)	256 x 256	16 x 16	0.25	**2.15**	13.80	6.42
LMD-PS (Ours)	ViT-B (12/8 blocks)	224 x 224	16 x 16	0.15 / 0.4 / 0.75	2.78	8.37	3.01
LMD-CS (Ours)	ViT-B (12/8 blocks)	224 x 224	16 x 16	cosine-based values	2.61	**6.92**	**2.65**

Table 1: The pre-training mean time-consumption versus MAEs methods on ImageNet-1K dataset.

Method	**Mask Ratio**	**MAT@1↓**	**MAT@5↓**
MAE (2022)	0.75	0.374	0.253
LMD-PS (Ours)	0.15 / 0.4 / 0.75	0.158	0.137
LMD-CS (Ours)	cosine values	**0.135**	**0.102**

Table 2: The fine-tuning valid mean time-consumption for accuracy improvement on ImageNet-1K dataset.

LSUN-Bedrooms 256 x 256				
Method	**Backbone**	**MIT ↓**	**MLT ↓**	**MLI ↓**
DDPM (2020)	U-Net	5.45	39.89	7.32
iDDPM (2021)	U-Net	5.32	24.74	4.65
DDIM (2020)	U-Net	4.46	29.53	6.62
LDM (2022)	U-Net	2.86	15.84	5.54
LMD (Ours)	ViT-B	**2.45**	**7.06**	**2.88**

Table 3: The training mean time-consumption versus DPMs methods on LSUN-Bedrooms dataset.

ratio at 0.4, and for the remaining 1/3 steps, we continue to linearly increase until reaches 0.75, and maintain such a ratio till the end of the training.

3) Cosine Scheduling. From Figure 5, we can observe that under piecewise scheduling, the model still performs poor in both the early stages (e.g., at step $12e^3$) and the later stages (e.g., at step $94e^3$). We guess the reasons why the model requires more training steps in the early and later stages are respectively due to poor data fitting and high mask ratio. To address this issue, we further propose a new cosine scheduling scheme. Specifically, considering that the cosine function can approximate the scheduling process well, we adopt the cosine function to compute a sequence of mask ratios. As can been seen from Figure 5, such a scheduling scheme achieves the best results.

4 Experiments

4.1 Experimental Setup

Datasets and Metrics. Following (He et al. 2022; Ho, Jain, and Abbeel 2020), we pre-train our model on ImageNet-1K (IN1K) (Deng et al. 2009) and LSUN-Bedrooms (Yu et al. 2015) respectively. Three main metrics: mean iteration time (MIT), mean loss-decrease time (MLT) and mean loss-decrease iterations (MLI) are adopted to evaluate the valid training time-consumption of the models, which will be detailed in the Appendix. On the downstream classification task, we evaluate our model on the IN1K dataset (with 1000 object categories) and adopt mean accuracy-increase time (MAT) including MAT@1 and MAT@5 as the main evaluation metrics, which can be used to test the valid time-consumption with mean accuracy improvement. Moreover, FID, CLIP-score and LPIPS metrics are used to evaluate the generative performance.

Baselines. For more holistic comparisons, we compare LMD with the two categories of baseline models: 1) DPMs, including DDPM (Ho, Jain, and Abbeel 2020), iDDPM (Nichol and Dhariwal 2021), DDIM (Song, Meng, and Ermon 2020) and LDM (Rombach et al. 2022); 2) MAEs, including MAE (He et al. 2022), SimMIM (Xie et al. 2022b), GreenMIM (Xie et al. 2022b) and UM-MAE (Li et al. 2022b). Note all baselines use the *base* model.

Implementation Details. LMD adopts 20-layers ViT as the backbone, of which 8 encoder blocks and 12 decoder blocks for generative training, and 12 encoder blocks and 8 decoder blocks for discriminant training. The mask ratio of the mask scheduler is set in $[0.15, 0.75]$. The scaling factor f is set as 8. The base learning rate is set as $1.5e^{-4}$, and the weight decay is set as 0.05. We use the Adan (Xie et al. 2022a) optimizer to optimize the model.

4.2 Overall Performance

As shown in Table 1, compared with MAEs methods, LMD achieves the best performance on MLT and MLI metrics. As for the MIT metric, although UM-MAE achieves the optimal mean iteration time, but with a very low mask ratio of 0.25. In fact, the MLT metric can better reflect the efficiency of the model for valid loss decreases, and we are nearly twice as fast as UM-MAE on this metric. From Table 2, we can observe from this fine-tuning results that LMD is much faster than MAE, and the actual speed up is about $3\times$ under the same accuracy contribution, which further proves LMD's effectiveness. Moreover, Table 3 shows the comparison between our model and DPMs-based methods in terms of training time. From Table 3, it can be clearly seen that LMD greatly accelerate the generative training process, which proves that the LMD model is substantially more efficient in training as compared to the DPMs-based methods, because it can take full advantage of the parallel computing of GPU by utilizing ViT as the backbone. Further, Table 4 shows that our LMD has achieved more competitive results in generative performance than SD-v1.4 or the recent Muse$_{base}$ with the best FID, CLIP-score and LPIPS results.

	FID ↓	CLIP-score ↑	LPIPS ↓
SD-v1.4 (2022)	17.01	0.24	0.45
Muse$_{base}$ (2023)	6.8	0.25	0.33
LMD-CS (Ours)	**6.2**	**0.26**	**0.27**

Table 4: Generative evaluation on CC (Sharma et al. 2018).

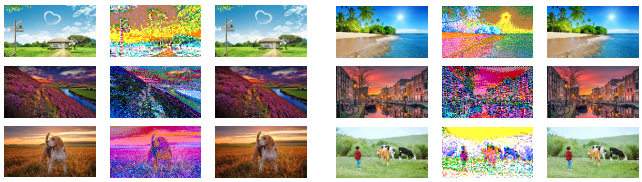


Figure 6: Case studies to test the impact of latent projector.

4.3 Ablation Studies

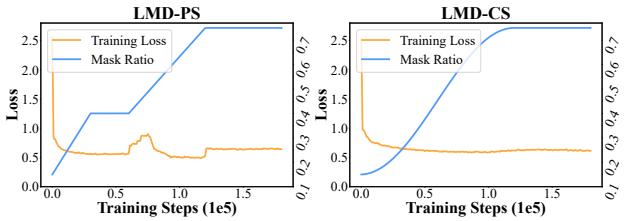
In this part, we perform ablation experiments to evaluate the impact of each setting in our LMD on training time-consumption. We focus on three crucial settings, as shown in Table 5. Specifically, #1 indicates the complete LMD model; #2 w/o L (LSP) denotes that we remove the latent space projector and directly perform the masking diffusion strategy on the pixel space; #3 w/o M (MAE) denotes that we remove the MAE encoder-decoder blocks and replace with a U-Net (Ronneberger, Fischer, and Brox 2015) model; #4 w/o D (MDS) denotes that we remove the masking diffusion scheduler and adopt a fixed 0.75 mask ratio for reconstruction. From Table 5, we can observe that removing each component will result in a time-consumption increasing, which proves the effectiveness of all the settings employed by LMD. In particular, w/o MAE caused a mean delay of 2.70, while w/o MDS only caused a mean delay of 1.27, indicating that low temporal-dependence directly promotes the reduction of the training time-consumption, while high spatial-dependence is a secondary optimization factor, which indirectly reduces the training time-consumption. In contrast, w/o LSP only caused an average delay of 0.33. We suspect that this is due to the adoption of ViT to offset latent space acceleration, but it may also have a significant impact on reducing GPU memory consumption.

4.4 Further Analysis

The impact of latent space projector. To better illustrate the effectiveness of our model in latent space reconstruction, we sample and visualize the latent image $\hat{z}$ (middle of the three columns in Figure 6) compressed by the latent space projector and ultimately reconstructed images (right of the three columns in Figure 6). It can be observed that our latent space projector based on VQ-GAN almost achieves lossless image compression, so the accuracy of latent space image reconstruction can be guaranteed. Moreover, when the latent space projector fails on partial images (loss of accuracy due to unseen feature distributions), our model can ensure better generalization by adding an explicit reconstruction loss $||x - \hat{x}||_2^2$ to efficiently fine-tune the ViT blocks, without re-training the latent space projector.

#	Setting	Overall↓ mean	MIT train	MLT train	MLI train	MAT@1 dev	MAT@5 dev
1	**LMD**	**2.48**	**2.61**	**6.92**	**2.65**	**0.135**	**0.102**
2	w/o L	2.81	3.18	7.92	2.49	0.285	0.198
3	w/o M	5.18	2.95	16.52	5.60	0.425	0.385
4	w/o D	3.75	2.62	11.40	4.35	0.225	0.143

Table 5: Ablation study on IN1K dataset.



(a) LMD-PS. (b) LMD-CS.

Figure 7: The loss curves from LMD-PS and LMD-CS.

Exploration of mask scheduling schemes. To more clearly illustrate the impact of different schedulers on loss decline and show why cosine scheduling is more effective, we respectively visualize the loss decline curves from LMD-PS and LMD-CS, as shown in Figure 7. From Figure 7 (a), we can see that there is an obvious fluctuation when the mask ratio exceeds 0.4, which indicates that the learning efficiency of the model at this time is slightly slower than that of the mask scheduler. This can be solved by two different ways: (1) adding more training steps at 0.4 mask ratio; (2) adopting a more gentle scheduling scheme (such as cosine scheduling) within the range of 0 to 0.4. The former requires more training time to make up for the learning efficiency of the model (similar to MAEs with fixed mask ratio). In contrast, the cosine-based scheduling scheme shown in Figure 7 (b) is more gentle in the decline of losses, which proves its potential advantage in reducing training time-consumption.

5 Conclusion

In this paper, we propose LMD, a latent masking diffusion framework for faster image synthesis. Specifically, we first employ a pre-trained latent space projector to compress the input image into a latent space with smaller scale to obtain their latent feature map, and then split the latent feature map into a patch sequence for masked self-supervised training. Unlike conventional using fixed mask ratio to reconstruction, we propose to gradually increase the masking ratio by mask schedulers and reconstruct the images by a progressive diffusion mode. By unifying the latent space project technique, mask self-supervised technique and diffusion generative scheme, LMD can reduce the total training time-consumption. Experiments on the representative ImageNet-1K and LSUN-Bedrooms datasets demonstrate the effectiveness of LMD, and illustrates its high-efficiency in training both generative and discriminant tasks.

Acknowledgments

This work was supported by the Project funded by China Postdoctoral Science Foundation (No.2023M741950), and the Nationally Funded Postdoctoral Researcher Program (GZB20230347).

References

Cao, H.; Tan, C.; Gao, Z.; Chen, G.; Heng, P.-A.; and Li, S. Z. 2022. A survey on generative diffusion model. *arXiv preprint arXiv:2209.02646*.

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

Chen, J.; Hu, M.; Li, B.; and Elhoseiny, M. 2022. Efficient Self-supervised Vision Pretraining with Local Masked Reconstruction. *arXiv preprint arXiv:2206.00790*.

Chu, X.; Tian, Z.; Wang, Y.; Zhang, B.; Ren, H.; Wei, X.; Xia, H.; and Shen, C. 2021. Twins: Revisiting the design of spatial attention in vision transformers. *Advances in NeurIPS*, 34: 9355–9366.

Croitoru, F.-A.; Hondru, V.; Ionescu, R. T.; and Shah, M. 2022. Diffusion models in vision: A survey. *arXiv preprint arXiv:2209.04747*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*, 248–255.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *Advances in NeurIPS*, 34: 8780–8794.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

Esser, P.; Rombach, R.; and Ommer, B. 2021. Taming transformers for high-resolution image synthesis. In *CVPR*, 12873–12883.

Gu, S.; Chen, D.; Bao, J.; Wen, F.; Zhang, B.; Chen, D.; Yuan, L.; and Guo, B. 2022. Vector quantized diffusion model for text-to-image synthesis. In *CVPR*, 10696–10706.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *CVPR*, 16000–16009.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in NeurIPS*, 33: 6840–6851.

Huang, L.; You, S.; Zheng, M.; Wang, F.; Qian, C.; and Yamasaki, T. 2022. Green hierarchical vision transformer for masked image modeling. *Advances in Neural Information Processing Systems*, 35: 19997–20010.

Kenton, J. D. M.-W. C.; and Toutanova, L. K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 4171–4186.

Kim, D.; Na, B.; Kwon, S. J.; Lee, D.; Kang, W.; and Moon, I.-C. 2022. Maximum Likelihood Training of Implicit Nonlinear Diffusion Models. *arXiv preprint arXiv:2205.13699*.

Li, G.; Zheng, H.; Liu, D.; Wang, C.; Su, B.; and Zheng, C. 2022a. Semmae: Semantic-guided masking for learning masked autoencoders. *arXiv preprint arXiv:2206.10207*.

Li, X.; Wang, W.; Yang, L.; and Yang, J. 2022b. Uniform Masking: Enabling MAE Pre-training for Pyramid-based Vision Transformers with Locality. *arXiv preprint arXiv:2205.10063*.

Li, Z.; Chen, Z.; Yang, F.; Li, W.; Zhu, Y.; Zhao, C.; Deng, R.; Wu, L.; Zhao, R.; Tang, M.; et al. 2021. Mst: Masked self-supervised transformer for visual representation. *Advances in NeurIPS*, 34: 13165–13176.

Liu, J.; Huang, X.; Liu, Y.; and Li, H. 2022a. MixMIM: Mixed and Masked Image Modeling for Efficient Visual Representation Learning. *arXiv preprint arXiv:2205.13137*.

Liu, L.; Ren, Y.; Lin, Z.; and Zhao, Z. 2022b. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*.

Liu, X.; Park, D. H.; Azadi, S.; Zhang, G.; Chopikyan, A.; Hu, Y.; Shi, H.; Rohrbach, A.; and Darrell, T. 2021a. More control for free! image synthesis with semantic diffusion guidance. *arXiv preprint arXiv:2112.05744*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021b. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 10012–10022.

Ma, Y.; He, Y.; Cun, X.; Wang, X.; Shan, Y.; Li, X.; and Chen, Q. 2023a. Follow Your Pose: Pose-Guided Text-to-Video Generation using Pose-Free Videos. *arXiv preprint arXiv:2304.01186*.

Ma, Y.; Yang, T.; Shan, Y.; and Li, X. 2022a. Simvtp: Simple video text pre-training with masked autoencoders. *arXiv preprint arXiv:2212.03490*.

Ma, Z.; Jia, G.; and Zhou, B. 2023. AdapEdit: Spatio-Temporal Guided Adaptive Editing Algorithm for Text-Based Continuity-Sensitive Image Editing. *arXiv preprint arXiv:2312.08019*.

Ma, Z.; Li, J.; Li, G.; and Cheng, Y. 2022b. UniTranSeR: A unified transformer semantic representation framework for multimodal task-oriented dialog system. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 103–114.

Ma, Z.; Li, J.; Li, G.; and Huang, K. 2022c. Cmal: A novel cross-modal associative learning framework for vision-language pre-training. In *Proceedings of the 30th ACM International Conference on Multimedia*, 4515–4524.

Ma, Z.; Yu, Z.; Li, J.; and Li, G. 2023b. HybridPrompt: bridging language models and human priors in prompt tuning for visual question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 13371–13379.

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

Nichol, A. Q.; and Dhariwal, P. 2021. Improved denoising diffusion probabilistic models. In *ICML*, 8162–8171.

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

Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-shot text-to-image generation. In *ICML*, 8821–8831.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR*, 10684–10695.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241.

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

Sharma, P.; Ding, N.; Goodman, S.; and Soricut, R. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2556–2565.

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

Vahdat, A.; Kreis, K.; and Kautz, J. 2021. Score-based generative modeling in latent space. *Advances in NeurIPS*, 34: 11287–11302.

Van Den Oord, A.; Vinyals, O.; et al. 2017. Neural discrete representation learning. *Advances in NeurIPS*, 30.

Wei, C.; Fan, H.; Xie, S.; Wu, C.-Y.; Yuille, A.; and Feichtenhofer, C. 2022. Masked feature prediction for self-supervised visual pre-training. In *CVPR*, 14668–14678.

Xie, X.; Zhou, P.; Li, H.; Lin, Z.; and Yan, S. 2022a. Adan: Adaptive Nesterov Momentum Algorithm for Faster Optimizing Deep Models. *arXiv preprint arXiv:2208.06677*.

Xie, Z.; Zhang, Z.; Cao, Y.; Lin, Y.; Bao, J.; Yao, Z.; Dai, Q.; and Hu, H. 2022b. Simmim: A simple framework for masked image modeling. In *CVPR*, 9653–9663.

Yang, L.; Zhang, Z.; Song, Y.; Hong, S.; Xu, R.; Zhao, Y.; Shao, Y.; Zhang, W.; Cui, B.; and Yang, M.-H. 2022. Diffusion models: A comprehensive survey of methods and applications. *arXiv preprint arXiv:2209.00796*.

Yu, F.; Seff, A.; Zhang, Y.; Song, S.; Funkhouser, T.; and Xiao, J. 2015. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*.

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

Zhang, X.; Tian, Y.; Huang, W.; Ye, Q.; Dai, Q.; Xie, L.; and Tian, Q. 2022. HiViT: Hierarchical Vision Transformer Meets Masked Image Modeling. *arXiv preprint arXiv:2205.14949*.

Zhou, J.; Wei, C.; Wang, H.; Shen, W.; Xie, C.; Yuille, A.; and Kong, T. 2021. Image BERT Pre-training with Online Tokenizer. In *ICLR*.