TimeStep Master: Asymmetrical Mixture of Timestep LoRA Experts for Versatile and Efficient Diffusion Models in Vision

Shaobin Zhuang [*1†] **Yiwei Guo** [*2] **Yanbo Ding** [*2] **Kunchang Li** [2] **Xinyuan Chen** [‡3] **Yaohui Wang** [‡3]
Fangyikang Wang [4] **Ying Zhang** [5] **Chen Li** [5] **Yali Wang** [‡2 3]

Abstract

Diffusion models have driven the advancement of vision generation over the past years. However, it is often difficult to apply these large models in downstream tasks, due to massive fine-tuning cost. Recently, Low-Rank Adaptation (LoRA) has been applied for efficient tuning of diffusion models. Unfortunately, the capabilities of LoRA-tuned diffusion models are limited, since the same LoRA is used for different timesteps of the diffusion process. To tackle this problem, we introduce a general and concise TimeStep Master (TSM) paradigm with two key fine-tuning stages. In the fostering stage (1-stage), we apply different LoRAs to fine-tune the diffusion model at different timestep intervals. This results in different TimeStep LoRA experts that can effectively capture different noise levels. In the assembling stage (2-stage), we design a novel asymmetrical mixture of TimeStep LoRA experts, via core-context collaboration of experts at multi-scale intervals. For each timestep, we leverage TimeStep LoRA expert within the smallest interval as the core expert without gating, and use experts within the bigger intervals as the context experts with time-dependent gating. Consequently, our TSM can effectively model the noise level via the expert in the finest interval, and adaptively integrate contexts from the experts of other scales, boosting the versatility of diffusion models. To show the effectiveness of our TSM paradigm, we conduct extensive experiments on three typical and popular LoRA-related tasks of diffusion models, including domain adaptation, post-pretraining, and model distillation. Our TSM achieves the state-of-the-art results on all these tasks, throughout various model structures (UNet, DiT and MM-DiT) and visual data modalities (Image and Video), showing its remarkable generalization capacity.

1. Introduction

Diffusion models have shown remarkable success in vision generation (Rombach et al., 2022b; Podell et al., 2023; Singer et al., 2022; Ho et al., 2022a; Chen et al., 2024d; Esser et al., 2024b; Chen et al., 2024a). Especially with the guidance of scaling law, they demonstrate the great power in generating images and videos from user prompts (Esser et al., 2024b; Liu et al., 2024a;c; Bao et al., 2024) owing to billions of model parameters. However, it is often difficult to deploy these diffusion models efficiently in various downstream tasks, since fine-tuning such huge models is resource-consuming. To fill this gap, Low-Rank Adaptation (LoRA) (Hu et al., 2021), initially developed in NLP (Chowdhary & Chowdhary, 2020), has been applied to diffusion models for rapid adaptation and efficient visual generation (Luo et al., 2023a; Li et al., 2024b; Peng et al., 2024; Yin et al., 2024b).

However, we observe that the generative capability of LoRA-tuned diffusion models is limited. For illustration, we take the well-known PixArt-α (Chen et al., 2024d) as an example, which is pre-trained on SAM-LLaVA-Captions10M (Chen et al., 2024d) for image generation. As shown in Fig. 1 (b) and (c), we perform LoRA on two typical fine-tuning settings. On one hand, we fine-tune this model with LoRA on new image data (*e.g.*, T2I-CompBench (Huang et al., 2023)). In this setting of downstream adaptation, the LoRA-tuned model makes similar errors as the pre-trained model, *i.e.*, they both fail to fit the target data distribution. On the other hand, we fine-tune this model with LoRA on the pretraining image data. In this setting of post-pretraining, LoRA-tuned model results in prompt misalignment, which deteriorates the generative capacity of the pre-trained model. Based on these observations, there is a natural question: *why does*

[*]Equal contribution , [‡]Equal corresponding ,[†]Work done as intern at WeChat, Tencent Inc. [1]Shanghai Jiao Tong University. [2]Shenzhen Key Laboratory of Computer Vision and Pattern Recognition, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. [3]Shanghai Artificial intelligence Laboratory. [4]Zhejiang University. [5]WeChat, Tencent Inc.. Correspondence to: Yali Wang <yl.wang@siat.ac.cn>, Xinyuan Chen <chenxinyuan@pjlab.org.cn>, Yaohui Wang <wangyaohui@pjlab.org.cn>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

Figure 1. **Motivation Visualization.** (a) During generation process, the hidden states in the same block of pre-trained PixArt-α changes significantly with timestep. (b) The pre-trained model and LoRA-tuned model incorrectly generate green bench and red vase, while TSM corrects these errors. (c) LoRA-tuned model generates degraded images, while TSM benefits visual quality and text alignment.

such deterioration appear in LoRA-tuned diffusion models? We generated 30K prompts from the COCO2014 (Lin et al., 2014) validation set and observed hidden states during the inference process. As shown in Fig. 1 (a), when Diffusion models process input at different timesteps, the performance of even the same block varies greatly (Balaji et al., 2022; Xue et al., 2024; Hang et al., 2023; Go et al., 2023). We then hypothesis that it is the low-rank characteristic of LoRA that makes it difficult to learn complex representations at different timesteps. In the vanilla LoRA setting, only ONE LoRA is applied for fine-tuning diffusion models at DIFFERENT timesteps. Thus, in the downstream adaptation case, it fails to fit the new target data just like the pre-trained model. In the post-pretraining case, such an inconsistent manner would reduce the capability of diffusion models to tackle different noise levels, especially with very limited parameters in LoRA (more evidence provided in Tab. 1 and 2).

To alleviate this problem, we propose a general and concise TimeStep Master (TSM) paradigm, with a novel asymmetrical mixture of TimeStep LoRA experts. Specifically, our TSM contains two distinct stages of fostering and assembling TimeStep LoRA experts, boosting the versatility and efficiency of tuning diffusion models in vision. In the fostering stage, we divide the training procedure into several timestep intervals. For different intervals, we introduce different LoRA modules for fine-tuning the diffusion model, leading to different TimeStep LoRA experts. This can effectively enhance the diffusion model to fit the data distribution under different noise levels. In the assembling stage, we combine the TimeStep LoRA experts of multi-scale intervals to further boost performance. Specifically, we introduce a novel asymmetrical mixture of TimeStep LoRA experts, for core-context expert collaboration. For each timestep, we leverage TimeStep LoRA expert within the smallest interval as the core expert without gating, and use experts within the bigger intervals of other scales as the context experts with time-dependent gating. In this case, our TSM can effectively learn the noise level via the expert in the finest interval, as well as adaptively integrate contexts from the experts of other scales, boosting the versatility and generalization capacity of diffusion model.

To show the effectiveness of our TSM paradigm, we conduct extensive experiments on three typical and popular LoRA-related tasks of diffusion models, including domain adaptation, post-pretraining, and model distillation. Our TSM achieves the state-of-the-art results on all these tasks, throughout various model structures (UNet (Ronneberger et al., 2015), DiT (Peebles & Xie, 2023), MM-DiT (Esser et al., 2024a)) and visual data modalities (Image, Video), showing its remarkable generalization capacity. For the above three tasks, TSM achieves the best performance on T2I-CompBench, efficiently improves model performance after post-pretraining using only public datasets, and our 4-step model reaches the FID of 9.90 on COCO2014 with a very low resource consumption of 3.7 A100 days.

2. Related Work

Diffusion models for visual synthesis. Recently, diffusion models (DMs) have swept across the realm of visual generation and have become the new state-of-the-art generative models for text-to-image (Podell et al., 2023; Nichol et al., 2021; Li et al., 2023; Saharia et al., 2022; Chen et al., 2024d;b;c; Xue et al., 2024) and text-to-video (Ho et al., 2022b; Blattmann et al., 2023; Khachatryan et al., 2023; Luo et al., 2023b; Wang et al., 2023; Singer et al., 2022; Chen et al., 2023a; Zhuang et al., 2024). Stable Diffusion 1.5 (SD1.5) (Rombach et al., 2022b) operates in the latent space and can generate high-resolution images. The PixArt series (Chen et al., 2024d;b;c) provide more accessibility in high-quality image generation by introducing efficient training and inference strategies. SD3 (Esser et al., 2024b) demonstrates even more astonishing generation results with the MM-DiT architecture and scaled-up parameters. VideoCrafter2 (VC2) (Chen et al., 2024a) discovers the spatial-temporal relationships of the video diffusion model and further proposes an effective training paradigm for high-

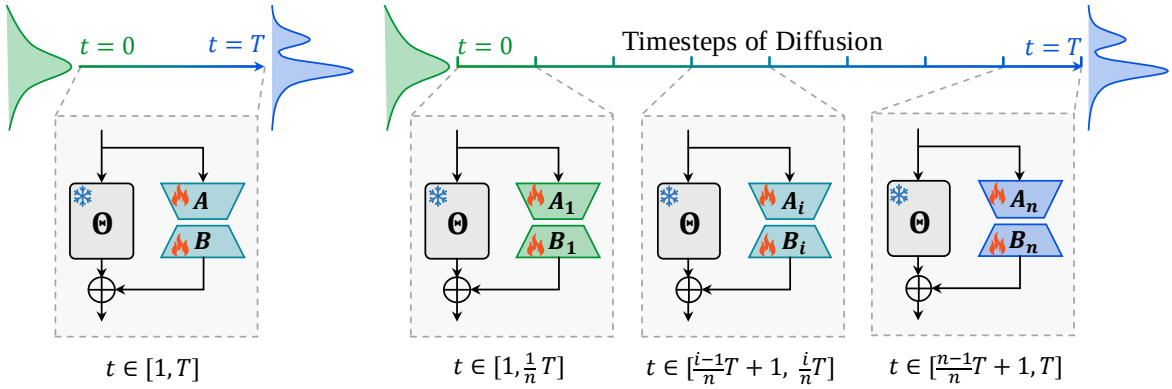

Figure 2. **Fostering Stage: TimeStep LoRA Expert Construction.** We divide all T timesteps into n intervals and fine-tune the diffusion model with individual LoRA module for each interval.

quality video generation. MoE based methods (Park et al., 2024; Lee et al., 2023; Balaji et al., 2022; Zhang et al., 2024; Fei et al., 2024; Sun et al., 2024; Fang et al., 2024) try to use multiple experts to better learn complex data distributions. However, the increasing number of parameters of the DMs also makes it difficult to directly transfer its powerful capabilities to other domains.

Efficient tuning of diffusion models. To reduce the cost of full fine-tuning DMs in downstream tasks and retain generalization ability, prompt tuning is a common and efficient fine-tuning method. DMP (Ham et al., 2024) adapts DMs to different timesteps by simultaneously training gating modules and prompt tokens, but performance still lags behind that of the LoRA-based (Hu et al., 2021) fine-tuning method. LoRA (Hu et al., 2021) is widely applied in DMs to efficiently train low-rank matrices (Zhang et al., 2023; Ye et al., 2023; Xie et al., 2023; Mou et al., 2024; Lin et al., 2024a; Xing et al., 2024; Ran et al., 2024; Gu et al., 2024; Lyu et al., 2024; Huang et al., 2023; Go et al., 2022; Choi et al., 2024). GORS (Huang et al., 2023) applies LoRA to fine-tune the DMs to the target domain. DMD (Yin et al., 2024b) supports the use of LoRA in model distillation for fast inference. ControlNeXt (Peng et al., 2024) employs LoRA for efficient and enhanced controllable generation. T2V-Turbo (Li et al., 2024b) injects LoRA into video diffusion model (Chen et al., 2024a) and optimizes with mixed rewards, achieving inference acceleration and quality improvement. DeMe (Ma et al., 2024) makes DMs better fit the data distribution by weighting the parameters of LoRA trained for different timesteps, but simple weighting cannot fully develop the knowledge of multiple LoRA. However, as discussed earlier, the generation capabilities of LoRA-tuned DMs are limited. We tackle this with our TSM, which assigns TimeStep LoRA experts to learn the distribution within diverse noise levels, and assemble these experts for further information aggregation. Using TSM, the generative performance of pre-trained diffusion models is significantly enhanced at a low fine-tuning cost.

3. Method

In this section, we introduce our TimeStep Master (TSM) paradigm in detail. First, we briefly review the diffusion model and LoRA as preliminaries. Then, we explain two key fine-tuning stages in TSM, *i.e.*, expert fostering and assembling, in order to build an asymmetrical mixture of TimeStep LoRA experts for efficient and versatile enhancement of the diffusion model.

Diffusion Model. The diffusion model is designed to learn a data distribution by gradually denoising a normally-distributed variable (Song et al., 2021; Ho et al., 2020). It has been widely used for image/video generation (Rombach et al., 2022b; Podell et al., 2023; Singer et al., 2022; Ho et al., 2022a; Chen et al., 2024d; Zhuang et al., 2024; Chen et al., 2024e). In the forward diffusion process, one should add Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ on the input x_0, in order to generate the noisy input x_t at each timestep, $x_t = \sqrt{\overline{\alpha}_t} x_0 + \sqrt{1-\overline{\alpha}_t}\epsilon$, where $t = 1, 2, \cdots, T$, and T is the total number of timesteps in the forward process. $\overline{\alpha}_t$ is a parameter related to t. When t approaches T, $\overline{\alpha}_t$ approaches 0. The training goal is to minimize the loss for denoising,

$$\mathcal{L} = \mathbf{E}_{x_0,c,\epsilon,t}\left[\|\epsilon - \epsilon_\Theta(x_t, t, c)\|_2^2\right], \quad t \in [1, T], \tag{1}$$

where ϵ_Θ is the output of neural network with model parameters Θ, and c indicates the additional condition, *e.g.*, text input. To achieve superior performance, the diffusion model is often designed with a large number of network parameters that are pre-trained on large-scale web data. Apparently, it is computationally expensive to fine-tune such a big model for specific downstream tasks.

Low-Rank Adaptation (LoRA). To alleviate the above difficulty, LoRA (Hu et al., 2021) has been recently applied for rapid fine-tuning diffusion models on target data (Ruiz et al., 2023; Huang et al., 2023). Specifically, LoRA introduces

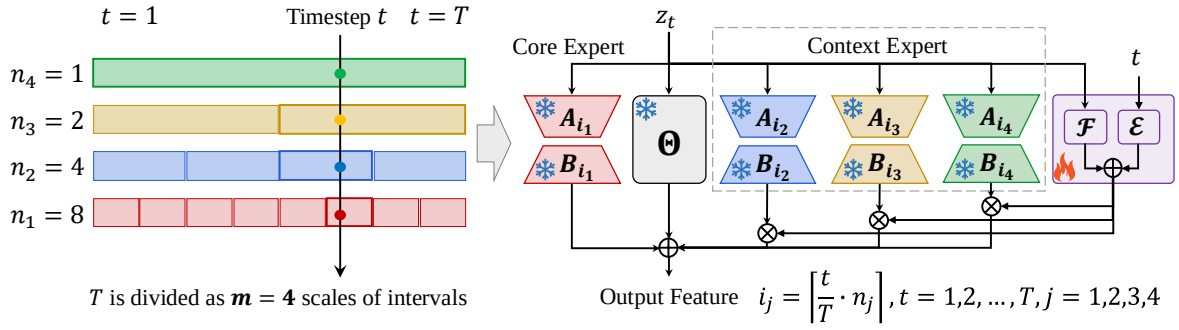

Figure 3. **Assembling Stage: Asymmetrical Mixture of TimeStep LoRA Experts.** We divide T into 4 intervals, namely n_1=8, n_2=4, n_3=2, n_4=1. The TimeStep LoRA expert within the smallest-scale interval plays the core role to model the noise level of t with fine granularity. The core expert (red) is without gating; the context experts (blue, yellow and green) are with gating. The router is timestep-dependent, which adaptively weights the importance of context experts at t.

low-rank decomposition of an extra matrix,

$$\Theta + \Delta\Theta = \Theta + BA, \tag{2}$$

where $\Theta \in R^{d\times k}$ is the pretrained parameter matrix of diffusion model. $\Delta\Theta \in R^{d\times k}$ is the extra parameter matrix that is decomposed as the multiplication of two low-rank matrices $A \in R^{r\times k}$ and $B \in R^{d\times r}$, where $r \ll d, k$. To achieve parameter-efficient fine-tuning, one can simply freeze the pre-trained parameter Θ, while only learning the low-rank matrices A and B on target data for computation cost reduction. However, the generation capabilities of these vanilla LoRA-tuned diffusion models are limited. The main reason is that, diffusion model exhibits different processing modes for the noisy inputs at different timesteps (Balaji et al., 2022; Hang et al., 2023). Alternatively, LoRA applies the same low-rank matrices A and B for different timesteps. Such inconsistency would reduce the capacity of diffusion model to tackle different noise levels, especially with a very limited number of learnable parameters in A and B. To address this problem, we propose a TimeStep Master (TSM) paradigm with two important stages as follows.

3.1. Fostering Stage: TimeStep LoRA Expert Construction

To learn different modes of the noisy inputs, we propose to introduce different LoRAs for different timesteps. Specifically, we uniformly divide the timesteps of T into n intervals. For the i-th interval, we introduce an individual LoRA,

$$\Theta + \Delta\Theta_i = \Theta + B_i A_i \tag{3}$$

where $A_i \in R^{r\times k}$ and $B_i \in R^{d\times r}$ refer to low-rank matrices in the i-th interval. We optimize A_i and B_i by fine-tuning the diffusion model on the noisy inputs within the i-th interval,

$$\mathcal{L} = \mathbf{E}_{x_0,c,\epsilon,t}\left[\|\epsilon - \epsilon_{\Theta,A_i,B_i}(x_t,t,c)\|_2^2\right], \quad t \in [\frac{i-1}{n}\cdot T + 1,\ \frac{i}{n}\cdot T]. \tag{4}$$

We dub the fine-tuned diffusion model as a TimeStep LoRA expert at interval i. Hence, we can obtain n TimeStep LoRA experts for n intervals of timesteps. During inference, we first sample x_T from Gaussian noise $x_T \sim \mathcal{N}(0, I)$, and then use these TimeStep LoRA experts to iteratively denoise x_T, i.e., when the timestep t iterates to one certain interval, we use the corresponding TimeStep LoRA expert of this interval to estimate the noise of x_t, where $t = T, ..., 1$.

It is worth mentioning that, there are two extreme cases with $n = 1$ and $n = T$. When $n = 1$, it refers to the vanilla LoRA setting that is limited to capture different noise levels at different timesteps. When $n = T$, it refers to the setting where there is a LoRA expert for each timestep. Apparently, this setting makes no sense since the noise levels are similar among the adjacent timesteps. Hence, it is unnecessary to equip a LoRA for each timestep. Especially T is often large in the diffusion model, such an extreme setting introduces too many LoRA parameters to learn. Consequently, we propose to divide T in different numbers of intervals, *i.e.*, $n = n_1, n_2, \cdots, n_m$. In this case, for each timestep t, there are m TimeStep LoRA experts. In the following, we introduce a novel asymmetrical mixture of these TimeStep LoRA experts, which can effectively and adaptively make them collaborate to further boost diffusion models via multi-scale noise modeling.

3.2. Assembling Stage: Asymmetrical Mixture of TimeStep LoRA Experts

Via the multi-scale design of interval division above, one can obtain m TimeStep LoRA experts for each timestep t. Hence, the next question is how to assemble their power to model the noise level of this step. Naively, one can leverage the standard Mixture of Experts (MoE) (Riquelme et al., 2021; Chen et al., 2023b) without distinguishing the role of experts. But this is not the case for TimeStep LoRA experts. Apparently, for each timestep, the TimeStep LoRA expert within the smallest interval plays the core role in modeling the noise level of this step with fine granularity. When

Method	Color↑	Shape↑	Texture↑	Spatial↑	Non-Spatial↑	Complex↑
SD1.4 (Rombach et al., 2022b)	37.65	35.76	41.56	12.46	30.79	30.80
SD1.5 (Rombach et al., 2022b)	36.97	36.27	41.25	11.04	31.05	30.79
SD2 (Rombach et al., 2022b)	50.65	42.21	49.22	13.42	31.27	33.86
SD2 + Composable (Liu et al., 2022)	40.63	32.99	36.45	8.00	29.80	28.98
SD2 + Structured (Yu et al., 2023)	49.90	42.18	49.00	13.86	31.11	33.55
SD2 + Attn Exct (Wang et al., 2024)	64.00	45.17	59.63	14.55	31.09	34.01
SD2 + GORS unbaised (Huang et al., 2023)	64.14	45.46	60.25	17.25	31.58	34.70
SDXL (Podell et al., 2023)	58.79	46.87	52.99	21.33	31.19	32.37
PixArt-α (Chen et al., 2024d)	41.70	37.96	45.27	19.89	30.74	33.43
PixArt-α-ft (Chen et al., 2024d)	66.90	49.27	64.77	20.64	**31.97**	34.33
DALLE3 (Betker et al., 2023)	77.85	62.05	70.36	28.65	30.03	37.73
SD3 (Esser et al., 2024b)	80.33	58.49	74.27	26.44	31.43	38.62
SD1.5 + Vanilla LoRA (Hu et al., 2021)	51.70	44.76	52.68	15.45	31.69	32.83
SD2 + Vanilla LoRA (Hu et al., 2021)	66.03	47.85	62.87	18.15	31.93	33.28
PixArt-α + Vanilla LoRA (Hu et al., 2021)	46.53	43.75	53.37	23.08	30.97	34.75
SD3 + Vanilla LoRA (Hu et al., 2021)	82.41	62.32	77.27	31.87	31.72	38.41
SD1.5 + TSM (Ours)	57.12	46.65	58.16	18.80	31.83	32.94
SD2 + TSM (Ours)	75.93	53.34	67.44	18.34	31.47	34.20
PixArt-α + TSM (Ours)	54.66	44.47	57.12	25.41	31.05	34.85
SD3 + TSM (Ours)	**83.45**	**63.16**	**78.18**	**34.50**	31.81	**38.71**

Table 1. **Domain Adaptation on T2I-CompBench.** Our TSM exhibits the strongest performance in all categories of T2I-Compbench.

Image Modality Method	Color↑	Shape↑	Texture↑	Spatial↑	Non-Spatial↑	Complex↑
PixArt-α (Chen et al., 2024d)	41.70	37.96	45.27	19.89	30.74	**33.43**
+ LoRA (Hu et al., 2021)	43.47 ↑ 1.77	34.74 ↓ 3.22	41.57 ↓ 3.70	15.37 ↓ 4.52	30.74	30.43 ↓ 3.00
+ TSM (Ours)	**48.86** ↑ 7.16	**37.97** ↑ 0.01	**47.31** ↑ 2.04	**21.55** ↑ 1.66	**31.13** ↑ 0.39	32.96 ↓ 0.47
Video Modality Method	**IS↑**	**Action↑**	**Amplitude↑**	**BLIP-BLEU↑**	**Color↑**	**Count↑**
VC2 (Chen et al., 2024a)	16.76	77.76	44.0	23.02	46.74	53.77
+ LoRA (Hu et al., 2021)	15.06 ↓ 1.70	73.85 ↓ 3.91	46.0 ↑ 2.0	21.89 ↓ 1.13	41.30 ↓ 5.44	27.89 ↓ 25.88
+ TSM (Ours)	**18.08** ↑ 1.32	**80.77** ↑ 3.01	**54.0** ↑ 10.0	**24.26** ↑ 1.24	**60.87** ↑ 14.13	**60.38** ↑ 6.61

Table 2. **Image/Video Modality Post-Pretraining on T2I-CompBench/EvalCrafter.** TSM performs much better than vanilla LoRA.

the interval is bigger, the granularity of noise modeling is getting bigger, *i.e.*, the TimeStep LoRA experts within bigger intervals are getting more insensitive to noise levels.

Based on this analysis, we introduce a novel and concise asymmetrical mixture of TimeStep LoRA experts for core-context expert collaboration. Specifically, for each timestep t, we leverage TimeStep LoRA expert within the smallest interval as the core expert without gating, and use the rest $(m-1)$ experts as the context ones with gating,

$$\begin{aligned}&\Theta + \Delta\Theta_{i_1} + \mathcal{G}(z_t, t) \odot [\Delta\Theta_{i_2}, ..., \Delta\Theta_{i_m}] \\ &= \Theta + B_{i_1}A_{i_1} + \sum\nolimits_{j=2}^{m} \mathcal{G}_j \odot B_{i_j}A_{i_j},\end{aligned} \tag{5}$$

Note that, we design the router of gating $\mathcal{G}(z_t, t) \in R^{m-1}$ to be timestep dependent, in order to adaptively weight contexts of the rest $(m-1)$ experts according to the timestep. Specifically, we make $\mathcal{G}(z_t, t)$ as a transformation of the timestep t and the input feature $z_t \in R^{k\times l}$ of this step. For simplicity, we design it as the sum over a FC layer of z_t and an embedding layer of t,

$$\mathcal{G}(z_t, t) = [\mathcal{G}_2, ..., \mathcal{G}_m] = \mathcal{F}(z_t) + \mathcal{E}(t), \tag{6}$$

where the embedding layer refers to a learnable matrix with a size of $T \times (m-1)$, and $\mathcal{E}(t)$ means that we extract the parameters in the t-th row as the embedding of timestep t. Finally, we minimize the diffusion loss function over this asymmetrical mixture of TimeStep LoRA experts,

$$\begin{aligned}\mathcal{L} = \mathbf{E}_{x_0,c,\epsilon,t}\left[\|\epsilon - \epsilon_{\Theta,\ \{A_{i_j},B_{i_j}\}_{j=1}^{m},\ \mathcal{G}}(x_t, t, c)\|_2^2\right], \\ i_j = \lceil \frac{t}{T} \cdot n_j \rceil,\end{aligned} \tag{7}$$

where the timestep t simultaneously belongs to intervals of m scales, *i.e.*, $t = 1, 2, \cdots, T$, and $j = 1, ..., m$. Note that, the TimeStep LoRA experts have been trained in the fostering stage. Hence, we freeze them and only learn the parameters of router $\mathcal{G}(z_t, t)$ in the assembling stage. Via such a distinct paradigm, our TSM can further boost diffusion to master noise modeling via TimeStep expert collaboration, as well as inherit the efficiency of LoRA for rapid adaption.

4. Experiments

We apply Timestep Master (TSM) to three typical fine-tuning tasks of diffusion model in visual generation: **domain adaptation**, **post-pretraining**, and **model distillation**. Extensive results demonstrate TSM achieves the state-of-the-art performance on all these tasks, throughout different model structures and modalities. We also make detailed ablation and visualization to show its effectiveness.

4.1. Domain Adaptation

Problem Definition and Dataset. Domain adaptation (Farahani et al., 2021) refers to the task of adapting a model trained on a source domain to perform well on a different

Family	Method	Resolution↑	N_{params}↓	Training Cost↓	FID↓
Unaccelerated Diffusion	DALL-E (Ramesh et al., 2021)	256	12.0B	2048 V100 × 3.4M steps	27.5
	DALL-E 2 (Ramesh et al., 2022)	256	6.5B	41667 A100 days	10.39
	Make-A-Scene (Gafni et al., 2022)	256	4.0B	-	11.84
	GLIDE (Nichol et al., 2021)	256	5.0B	-	12.24
	LDM (Rombach et al., 2022b)	256	1.45B	-	12.63
	Imagen (Saharia et al., 2022)	256	7.9B	4755 TPUv4 days	7.27
	eDiff-I (Balaji et al., 2022)	256	9.1B	256 A100 × 600K steps	6.95
	SD1.5 (50 step, cfg=3, ODE)	512	860M	6250 A100 days	8.59
	SD1.5 (200 step, cfg=2, SDE)	512	860M	6250 A100 days	7.21
Accelerated Diffusion	DPM++ (Lu et al., 2022)	512	-	-	22.36
	UniPC (4 step) (Zhao et al., 2024)	512	-	-	19.57
	LCM-LoRA (4 step) (Luo et al., 2023a)	512	67M	1.3 A100 days	23.62
	InstaFlow-0.9B (Liu et al., 2023)	512	0.9B	199 A100 days	13.10
	SwiftBrush (Nguyen & Tran, 2024)	512	860M	4.1 A100 days	16.67
	HiPA (Zhang & Hooi, 2023)	512	3.3M	3.8 A100 days	13.91
	UFOGen (Xu et al., 2024b)	512	860M	-	12.78
	SLAM (4 step) (Xu et al., 2024a)	512	860M	6 A100 days	10.06
	DMD (Yin et al., 2024b)	512	860M	108 A100 days	11.49
	DMD2 (Yin et al., 2024a)	512	860M	70 A100 days	8.35
	DMD2 + LoRA (Hu et al., 2021)	512	67M	3.6 A100 days	14.58
	DMD2 + TSM (Ours)	512	68M	3.7 A100 days	**9.90**

Table 3. **Model Distillation on 30K prompts from COCO2014.** Our TSM achieves comparable FID while lowering the training cost significantly. Rows marked in gray demonstrate the superiority of our TSM over the vanilla LoRA based on DMD2.

but related target domain. The goal is to fit the target domain distribution while preserving the strong generalization ability of the pre-trained model. We conduct domain adaptation experiments on T2I-CompBench (Huang et al., 2023), an open-world text-to-image generation benchmark which contains six domains. Each domain includes domain-specific training and testing prompts (700:300) and employs specialized models to evaluate generated test images and we convert all scores into percentile for ease of reading.

Implementation Details. Following (Huang et al., 2023), we generate 90 distinct 512x512 resolution images per training prompt for adaptation. We conduct both vanilla LoRA (Hu et al., 2021) and TSM experiments based on the pre-trained models of SD1.5 (Rombach et al., 2022a), SD2 (Rombach et al., 2022a), PixArt-α (Chen et al., 2024d) and Stable Diffusion 3 (SD3) (Esser et al., 2024b). Regarding the configuration of model training, for the sake of simplicity, we adopted the default configuration in the PEFT (Mangrulkar et al., 2022) library. More details in Sup. A.

As shown in Tab. 1, TSM achieves state-of-the-art results on T2I-CompBench and is far ahead in domains of color, shape, texture, and spatial. For complex domain, which contains more complex prompts and metrics than others, the performance of the model deteriorates after employing vanilla LoRA for domain adaptation. However, TSM can still improve the model performance.

4.2. Post-Pretraining

Problem Definition and Dataset. Post-pretraining (Luo et al., 2022) refers to the task of continuing to train a pre-trained model on a general dataset. The goal is to further improve the general performance of the model. We conduct experiments on post-pretraining tasks in both image and video modalities. For image modality, we evaluate our post-trained model on T2I-CompBench (Huang et al., 2023) as in Sec. 4.1. For video modality, we use EvalCrafter (Liu et al., 2024b), a public benchmark for text-to-video generation using 700 diverse prompts. Specifically, we adopt Inception Score (IS) for video quality assessment. For motion quality, we consider Action Recognition (Action) and Amplitude Classification Score (Amplitude). We evaluate text-video alignment with Text-Text Consistency (BLIP-BLEU) and Object and Attributes Consistency (Color and Count).

Implementation Details. For image modality, we conduct both vanilla LoRA (Hu et al., 2021) and TSM experiments based on the pre-trained model PixArt-α (Chen et al., 2024d) and the training dataset SAM-LLaVA-Captions 10M (Chen et al., 2024d). For video modality, we conduct experiments based on the pre-trained VideoCrafter2 (Chen et al., 2024a) and use a 70k subset of OpenVid-1M (Nan et al., 2024) for post-pretraining. More details can be seen in Sup. A.

As shown in Tab. 2, the performance of models using vanilla LoRA for post-pretraining drops significantly while TSM improves model performance with the same training data. In Sup. C.1, we also provide the TSM post-pretrained PixArt-α evaluation result on COCO2014 (Lin et al., 2014).

4.3. Model Distillation

Problem Definition and Dataset. Model distillation (Gou et al., 2021) refers to the task of training a simplified and

Model	FT Method	Color↑	Shape↑	Texture↑	Spatial↑	Non-Spatial↑	Complex↑
SD1.5 UNet	Vanilla LoRA	51.57	44.76	52.68	15.45	31.69	32.83
	TSM 1-stage	56.48↑ 4.91	45.91↑ 1.15	57.08↑ 5.12	18.01↑ 2.56	31.77↑ 0.08	32.79↓ 0.04
	TSM 2-stage	57.12↑ 5.55	46.65↑ 1.89	58.16↑ 5.48	18.80↑ 3.35	31.83↑ 0.14	32.94↑ 0.11
PixArt-α DiT	Vanilla LoRA	46.53	43.75	53.37	23.08	30.97	34.75
	TSM 1-stage	52.84↑ 6.31	43.92↑ 0.17	54.07↑ 0.7	25.35↑ 2.27	31.03↑ 0.06	35.04↑ 0.29
	TSM 2-stage	54.66↑ 8.13	44.47↑ 0.72	57.12↑ 3.75	25.41↑ 2.33	31.05↑ 0.08	34.85↑ 0.10
SD3 MM-DiT	Vanilla LoRA	82.41	62.32	77.27	31.87	31.72	38.41
	TSM 1-stage	82.52↑ 0.11	62.94↑ 0.62	77.55↑ 0.28	33.08↑ 1.21	31.74↑ 0.02	38.54↑ 0.13
	TSM 2-stage	83.45↑ 1.04	63.16↑ 0.84	78.18↑ 0.91	34.50↑ 2.63	31.81↑ 0.09	38.71↑ 0.30

Table 4. **Domain Adaptation Ablation on T2I-CompBench.**

IMG Model	FT Method	Color↑	Shape↑	Texture↑	Spatial↑	Non-Spatial↑	Complex↑
PixArt-α	Vanilla LoRA	43.47	34.74	41.57	15.37	30.74	30.43
	TSM 1-stage	45.66 ↑ 2.19	37.06 ↑ 2.32	45.42 ↑ 3.85	22.32 ↑ 6.95	31.03 ↑ 0.29	32.65 ↑ 2.22
	TSM 2-stage	48.86 ↑ 5.39	37.97 ↑ 3.23	47.31 ↑ 5.74	16.18 ↑ 1.66	31.13 ↑ 0.39	32.96 ↑ 2.53
VID Model	**FT Method**	**IS↑**	**Action↑**	**Amplitude↑**	**BLIP-BLEU↑**	**Color↑**	**Count↑**
VC2	Vanilla LoRA	15.06	73.85	46.0	21.89	41.30	27.89
	TSM 1-stage	16.71 ↑ 1.65	79.07 ↑ 5.22	50.0 ↑ 4.0	23.99 ↑ 2.10	56.52 ↑ 15.22	55.48 ↑ 27.59
	TSM 2-stage	18.08 ↑ 3.02	80.77 ↑ 6.92	54.0 ↑ 8.0	24.26 ↑ 2.37	60.87 ↑ 19.57	60.38 ↑ 32.49

Table 5. **Image and Video Post-Pretraining Ablation on T2I-CompBench and EvalCrafter.**

Metric	FT Method	Value
FID↓	Vanilla LoRA	14.58
	TSM 1-stage	9.92 ↓ 4.66
	TSM 2-stage	9.90 ↓ 4.68
Patch -FID↓	Vanilla LoRA	15.43
	TSM 1-stage	11.88 ↓ 3.55
	TSM 2-stage	11.82 ↓ 3.61
CLIP-Score↑	Vanilla LoRA	0.3176
	TSM 1-stage	0.3208 ↑ %1.01
	TSM 2-stage	0.3212 ↑ %1.13

Table 6. **Model Distillation Ablation based on DMD2.**

Model	z_t	t	Color↑	Shape↑	Texture↑	Spatial↑	Non-Spatial↑	Complex↑
SD1.5 UNet	✓	✓	**57.12**	**46.65**	**58.16**	**18.80**	**31.83**	32.94
	✗	✓	51.42	44.09	53.46	13.56	31.75	**33.45**
	✓	✗	53.64	46.24	55.08	16.31	31.72	33.42
PixArt-α DiT	✓	✓	**54.66**	44.47	**57.12**	**25.41**	**31.05**	34.85
	✗	✓	45.37	42.49	52.09	24.84	30.99	34.83
	✓	✗	47.23	**44.69**	54.30	25.25	30.99	**34.86**
SD3 MM-DiT	✓	✓	**83.45**	**63.16**	**78.18**	**34.50**	**31.81**	38.71
	✗	✓	80.99	60.38	74.62	31.87	31.61	38.53
	✓	✗	82.55	62.25	76.68	31.53	31.74	**38.87**

Table 7. **Gating Ablation on T2I-CompBench.** The model performance is optimal when the router input has both z_t and t.

efficient model to replicate the behavior of a complex one. Since LoRA is widely used in model distillation, we explore the capabilities of TSM in this task. We conduct experiments on 30K prompts from COCO2014 (Lin et al., 2014) validation set. Following DMD2 (Yin et al., 2024a), we generate images from these prompts and calculate the Fréchet Inception Distance (FID) (Heusel et al., 2017) from these images with 40,504 real images from validation dataset.

Implementation Details. We distill a 4-step (i.e., 999, 749, 499, 249) generator from 1000 steps of SD1.5 (Rombach et al., 2022b). Following DMD2, we first train the model without a GAN loss, and then with the GAN loss on 500K real images from LAION-Aesthetic (Schuhmann et al., 2022). More implementation details are in Sup. A.

Tab. 3 shows the SOTA comparison on model distillation, where N_{params} refers to the trainable parameters and *Training Cost* is calculated based on a single A100 GPU. Notably, our TSM far outperforms LoRA (FID 9.90 $vs.$ 14.58) with an increase of less than 1M trainable parameters and 0.1 A100 days gain of training cost. Although we could not achieve the lowest FID due to our limited training resources, we obtain a competitive result while significantly reducing the training cost. This demonstrates the effectiveness and efficiency of our TSM in model distillation.

4.4. Ablation Studies

Overall Design. We conduct two-stage ablation experiments on domain adaptation, post-pretraining, and model distillation. As shown in Tab. 4, in domain adaptation, our TSM significantly outperforms the vanilla LoRA on three main generative model architectures (UNet, DiT, and MM-DiT), verifying the generalization of TSM on model architecture. The model and training settings of SD1.5, PixArt-α and SD3 can be seen in Sup. A. As shown in Tab. 5, in post-pretraining, TSM achieves huge improvements over vanilla LoRA on two modalities (image and video), verifying the generalization of TSM on visual modality. The experimental settings are same as Sec. 4.2. As shown in Tab. 6, in model distillation, TSM outperforms the vanilla LoRA on FID, Patch-FID (Lin et al., 2024b; Chai et al., 2022), and CLIP score (Radford et al., 2021b) on 30K prompts from COCO2014, demonstrating the generality of our TSM throughout various tasks. We also compare the trainable parameters in Tab. 9. In TSM 1-stage, the training parameters are the same as vanilla LoRA and only 1/4 in the 2-stage.

Model	n	r	$step$	Col↑	Sha↑	Tex↑	Spa↑	NS↑	Com↑
SD1.5 UNet	*w/o fine-tuning*			36.97	36.27	41.25	11.04	31.05	30.79
	1	4	4000	49.10	44.62	53.62	14.00	31.69	33.02
	1	32	4000	51.70	44.76	52.68	15.45	31.69	32.83
	1	4	32000	51.86	44.74	55.74	15.70	31.70	29.84
	2	4	4000	52.02	43.61	55.43	16.35	31.74	**33.13**
	2	4	16000	54.30	45.25	**57.26**	17.05	**31.79**	31.50
	4	4	4000	54.24	45.78	56.61	17.97	31.73	**33.13**
	4	4	8000	**55.85**	**46.45**	**58.06**	**18.32**	**31.77**	32.95
	8	4	4000	**56.48**	**45.91**	57.08	**18.01**	**31.77**	32.79
	16	4	2000	54.20	45.61	56.39	14.56	31.71	**33.16**
PixArt-α DiT	*w/o fine-tuning*			41.70	37.96	45.27	19.89	30.74	33.43
	1	4	4000	46.26	42.58	52.01	23.00	30.88	34.58
	1	32	4000	46.53	43.75	53.37	23.08	30.97	34.75
	1	4	32000	52.55	43.47	53.20	22.95	31.00	33.67
	2	4	4000	50.68	43.69	54.57	24.41	30.96	34.76
	2	4	16000	53.00	**44.43**	**55.08**	24.95	31.02	34.63
	4	4	4000	51.96	43.42	53.38	24.76	31.02	**34.98**
	4	4	8000	**52.77**	43.77	**55.48**	**25.64**	**31.06**	34.68
	8	4	4000	**52.84**	**43.92**	54.07	**25.35**	**31.03**	**35.04**
	16	4	2000	51.98	43.54	53.98	25.22	30.93	34.95
SD3 MM-DiT	*w/o fine-tuning*			80.33	58.49	74.27	26.44	31.43	**38.62**
	1	4	4000	81.28	61.31	76.65	31.28	31.70	38.55
	1	32	4000	82.41	62.32	77.27	31.87	31.72	38.41
	1	4	32000	81.82	62.53	76.81	32.94	31.73	**38.97**
	2	4	4000	81.74	61.82	76.68	32.01	31.73	38.44
	2	4	16000	82.60	62.71	77.80	32.98	**31.79**	38.61
	4	4	4000	82.24	62.00	77.11	32.20	**31.79**	38.35
	4	4	8000	**82.76**	**62.77**	**77.57**	**33.01**	**31.75**	38.54
	8	4	4000	**82.52**	**62.94**	**77.55**	**33.08**	31.74	38.54
	16	4	2000	82.34	62.66	76.90	32.73	31.74	38.60

Table 8. **TSM 1-Stage Ablation.** n, r and $step$ represent the number, rank and fine-tuning steps of TimeStep experts. Values in **red** and **blue** represent the optimal and suboptimal respectively. When n=1, TSM 1-stage is equal to vanilla LoRA; when n>1, it significantly outperforms vanilla LoRA.

Model	Method	Trainable (%)
SD1.5 UNet	Vanilla LoRA	0.09604
	TM 1-stage	0.09604
	TM 2-stage	0.02722
PixArt-α DiT	Vanilla LoRA	0.06764
	TM 1-stage	0.06764
	TM 2-stage	0.01344
SD3 MM-DiT	Vanilla LoRA	0.05635
	TM 1-stage	0.05635
	TM 2-stage	0.01311

Table 9. **Comparison of Trainable Parameters.** W and R represent the trainable LoRA and Router in TSM.

r	Col↑	Sha↑	Tex↑	Spa↑	NS↑	Com↑
SD1.5	36.97	36.27	41.25	11.04	31.05	30.79
1	54.63	44.66	55.35	13.23	31.66	31.84
4	56.48	45.91	57.08	**18.01**	31.77	32.79
16	57.86	44.99	58.13	14.11	31.82	33.20
64	**59.37**	**46.267**	**58.99**	15.40	**31.86**	**33.59**

Table 10. **TSM 1-Stage LoRA Rank r Ablation.** We choose SD1.5 as the pre-train model and set n=8 for 1-stage. It is obvious that, as r increases, the performance of the model also increases significantly.

Fostering Stage. We conduct TSM 1-stage ablation experiments for TimeStep experts' n, r, and fine-tuning $step$ on T2I-CompBench, based on SD1.5, PixArt-α, and SD3. More details can be seen in Sup. A. Notably, when n=1, TSM 1-stage degenerates to vanilla LoRA. As shown in Tab. 8, regardless of whether we train each LoRA for the same steps, introduce equivalent training costs ($n\times step$=32K) or the same amount of additional parameters, all n=2, 4, 8, 16 configurations significantly outperform vanilla LoRA. This highlights that the TSM 1-stage surpasses vanilla LoRA. Moreover, we can find that the performance of n=4 and n=8 is similar and better than n=16. Therefore, we believe that n=8 is enough for the division of the overall timesteps. In addition, we also conduct ablation experiments on LoRA rank r in 1-stage in Tab. 10, and the results show that the performance of the model will also improve as r increases.

Model	n_{core}	n_{context}	Col↑	Sha↑	Tex↑	Spa↑	NS↑	Com↑
SD1.5 UNet	4	-	55.85	46.45	58.06	18.32	31.77	32.95
	-	1,4	56.42	45.77	56.59	17.17	31.76	32.66
	4	1	56.93	46.92	57.95	18.02	31.79	32.71
	4	2	56.84	46.70	57.70	17.86	31.75	32.80
	4	8	56.96	46.12	59.00	18.43	31.74	32.76
	8	-	56.48	45.91	57.08	18.01	31.77	32.79
	-	1,8	54.56	45.52	56.30	17.90	31.78	33.27
	8	1	57.12	46.65	58.16	18.70	31.83	32.94
	8	2	56.20	46.58	58.04	18.17	31.78	32.91
	8	4	56.63	46.70	58.80	18.84	31.77	32.69
	8	1,2,4	57.59	46.18	57.69	17.91	31.82	32.78
PixArt-α DiT	4	-	52.77	43.77	55.48	25.64	31.06	34.68
	-	1,4	53.24	43.79	54.70	25.63	31.06	35.02
	4	1	53.57	44.29	56.26	25.55	31.04	34.58
	4	2	53.54	44.02	56.02	26.17	31.08	34.41
	4	8	52.70	43.66	55.62	25.37	31.06	34.68
	8	-	52.84	43.92	54.07	25.35	31.03	35.04
	-	1,8	51.93	43.87	54.00	25.67	31.03	35.08
	8	1	54.66	44.47	57.12	25.41	31.05	34.85
	8	2	54.33	44.10	55.75	25.82	31.05	34.80
	8	4	54.03	43.73	54.72	26.06	31.03	34.83
	8	1,2,4	54.80	44.26	56.30	26.00	31.05	34.78
SD3 MM-DiT	4	-	82.76	62.77	77.57	33.01	31.75	38.54
	-	1,4	81.38	62.73	77.19	33.65	31.69	38.65
	4	1	83.47	63.00	77.92	34.18	31.80	38.66
	4	2	83.14	63.09	77.87	34.36	31.81	38.63
	4	8	83.30	62.94	78.02	34.37	31.80	38.66
	8	-	82.52	62.94	77.55	33.08	31.74	38.54
	-	1,8	82.84	62.60	76.11	34.21	31.75	38.67
	8	1	83.45	63.16	78.18	34.50	31.81	38.71
	8	2	82.89	62.90	77.58	34.30	31.80	38.68
	8	4	82.78	62.99	77.71	34.15	31.79	38.68
	8	1,2,4	83.02	62.97	77.83	34.12	31.79	38.60

Table 11. **TSM 2-Stage Ablation on T2I-CompBench.** n_{core} and n_{context} refer to the number of core experts and context experts respectively. Values in green represent the improved performance compared to the 1-stage model with the same core experts, while gray indicate the decreased. The results show that the design of asymmetric TimeStep LoRA experts assembly is better than the symmetric case or without assembly, and n_1=8, n_2=1 can achieve stable performance improvement.

Assembling Stage. We conduct TSM 2-stage ablation experiments on T2I-CompBench, based on TSM 1-stage model with r=4. The training settings are same as Fostering Stage ablation. As shown in Tab. 11, we ablate the core expert and context expert. It shows that TSM 2-stage can improve model performance in most cases compared to TSM 1-stage. But surprisingly, the number of context LoRA and the performance in 1-stage have little impact on the performance in 2-stage. This is why we use the simplest n_2=1 of context LoRA in the experimental settings in Sec. 4.1, 4.2, 4.3. We also study on the symmetry of the TimeStep experts without core LoRA in Tab. 11, all the TimeStep experts are context LoRA. The experiment results show that the 2-stage performance of the symmetrical pattern is often worse than the asymmetrical pattern. Finally, as shown in Tab. 7, we conduct ablation experiments on the router's input, and the results show that it is necessary for the router to receive both feature z_t and timestep t as inputs. Ablation study on TSM

and MoE LoRA (Li et al., 2024a) can be seen in Sup. C.2.

Visualization. As shown in Fig. 1 (b), in the domain adaptation task, the TSM fine-tuned model revises the incorrect images generated by the pre-trained model, while LoRA could not. As shown in Fig. 1 (c), in the post-pretraining task, the TSM fine-tuned model improves the alignment between images/videos and text without degrading visual quality, while the LoRA fine-tuned model exhibits a significant decline in both visual quality and vision-text alignment. More visualization results can be seen in Sup. B.

5. Conclusion

We introduce the TimeStep Master (TSM) paradigm for diffusion model fine-tuning. TSM employs different LoRAs on different timestep intervals. Through the fostering and assembling stages, TSM effectively learns diverse noise levels via an asymmetrical mixture of TimeStep LoRA experts. Extensive experiments show that TSM outperforms existing approaches in domain adaptation, post-pretraining, and model distillation. TSM demonstrates strong generalization across various architectures and modalities, marking a significant advancement in efficient diffusion model tuning.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgement

This work is supported by the National Key R&D Program of China(NO.2022ZD0160505).

References

Balaji, Y., Nah, S., Huang, X., Vahdat, A., Song, J., Zhang, Q., Kreis, K., Aittala, M., Aila, T., Laine, S., et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

Bao, F., Xiang, C., Yue, G., He, G., Zhu, H., Zheng, K., Zhao, M., Liu, S., Wang, Y., and Zhu, J. Vidu: a highly consistent, dynamic and skilled text-to-video generator with diffusion models. *arXiv preprint arXiv:2405.04233*, 2024.

Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., et al. Improving image generation with better captions. *Computer Science. https://cdn.openai.com/papers/dall-e-3.pdf*, 2(3):8, 2023.

Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S. W., Fidler, S., and Kreis, K. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22563–22575, 2023.

Chai, L., Gharbi, M., Shechtman, E., Isola, P., and Zhang, R. Any-resolution training for high-resolution image synthesis. In *European Conference on Computer Vision*, pp. 170–188. Springer, 2022.

Chen, H., Xia, M., He, Y., Zhang, Y., Cun, X., Yang, S., Xing, J., Liu, Y., Chen, Q., Wang, X., et al. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023a.

Chen, H., Zhang, Y., Cun, X., Xia, M., Wang, X., Weng, C., and Shan, Y. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7310–7320, 2024a.

Chen, J., Ge, C., Xie, E., Wu, Y., Yao, L., Ren, X., Wang, Z., Luo, P., Lu, H., and Li, Z. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. *arXiv preprint arXiv:2403.04692*, 2024b.

Chen, J., Wu, Y., Luo, S., Xie, E., Paul, S., Luo, P., Zhao, H., and Li, Z. Pixart-δ: Fast and controllable image generation with latent consistency models. *arXiv preprint arXiv:2401.05252*, 2024c.

Chen, J., Yu, J., Ge, C., Yao, L., Xie, E., Wu, Y., Wang, Z., Kwok, J., Luo, P., Lu, H., et al. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *International Conference on Learning Representations*, 2024d.

Chen, T., Chen, X., Du, X., Rashwan, A., Yang, F., Chen, H., Wang, Z., and Li, Y. Adamv-moe: Adaptive multi-task vision mixture-of-experts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17346–17357, 2023b.

Chen, X., Wang, Y., Zhang, L., Zhuang, S., Ma, X., Yu, J., Wang, Y., Lin, D., Qiao, Y., and Liu, Z. Seine: Short-to-long video diffusion model for generative transition and prediction. In *International Conference on Machine Learning*, 2024e.

Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., and Jitsev, J. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2818–2829, 2023.

Choi, J. Y., Park, J. R., Park, I., Cho, J., No, A., and Ryu, E. K. Simple drop-in lora conditioning on attention layers will improve your diffusion model. *ArXiv*, abs/2405.03958, 2024.

Chowdhary, K. and Chowdhary, K. Natural language processing. *Fundamentals of artificial intelligence*, pp. 603–649, 2020.

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024a.

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024b.

Fang, G., Ma, X., and Wang, X. Remix-dit: Mixing diffusion transformers for multi-expert denoising. *ArXiv*, abs/2412.05628, 2024.

Farahani, A., Voghoei, S., Rasheed, K., and Arabnia, H. R. A brief review of domain adaptation. *Advances in data science and information engineering: proceedings from ICDATA 2020 and IKE 2020*, pp. 877–894, 2021.

Fei, Z., Fan, M., Yu, C., Li, D., and Huang, J. Scaling diffusion transformers to 16 billion parameters. *ArXiv*, abs/2407.11633, 2024.

Gafni, O., Polyak, A., Ashual, O., Sheynin, S., Parikh, D., and Taigman, Y. Make-a-scene: Scene-based text-to-image generation with human priors. In *European Conference on Computer Vision*, pp. 89–106. Springer, 2022.

Go, H., Lee, Y., Kim, J.-Y., Lee, S., Jeong, M., Lee, H. S., and Choi, S. Towards practical plug-and-play diffusion models. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1962–1971, 2022.

Go, H., Kim, J., Lee, Y., Lee, S., Oh, S., Moon, H., and Choi, S. Addressing negative transfer in diffusion models. *ArXiv*, abs/2306.00354, 2023.

Gou, J., Yu, B., Maybank, S. J., and Tao, D. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

Gu, Y., Wang, X., Wu, J. Z., Shi, Y., Chen, Y., Fan, Z., Xiao, W., Zhao, R., Chang, S., Wu, W., et al. Mix-of-show: Decentralized low-rank adaptation for multi-concept customization of diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

Ham, S., Woo, S., Kim, J.-Y., Go, H., Park, B., and Kim, C. Diffusion model patching via mixture-of-prompts. *ArXiv*, abs/2405.17825, 2024.

Hang, T., Gu, S., Li, C., Bao, J., Chen, D., Hu, H., Geng, X., and Guo, B. Efficient diffusion training via min-snr weighting strategy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

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

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Conference and Workshop on Neural Information Processing Systems*, 2020.

Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D. P., Poole, B., Norouzi, M., Fleet, D. J., et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022a.

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022b.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Huang, K., Sun, K., Xie, E., Li, Z., and Liu, X. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747, 2023.

Khachatryan, L., Movsisyan, A., Tadevosyan, V., Henschel, R., Wang, Z., Navasardyan, S., and Shi, H. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15954–15964, 2023.

Lee, Y., Kim, J.-Y., Go, H., Jeong, M., Oh, S., and Choi, S. Multi-architecture multi-expert diffusion models. *ArXiv*, abs/2306.04990, 2023.

Li, D., Ma, Y., Wang, N., Ye, Z., Cheng, Z., Tang, Y., Zhang, Y., Duan, L., Zuo, J., Yang, C., and Tang, M. Mixlora: Enhancing large language models fine-tuning with lora-based mixture of experts. *arXiv preprint arXiv:2404.15159*, 2024a.

Li, J., Feng, W., Fu, T.-J., Wang, X., Basu, S., Chen, W., and Wang, W. Y. T2v-turbo: Breaking the quality bottleneck of video consistency model with mixed reward feedback. *arXiv preprint arXiv:2405.18750*, 2024b.

Li, Y., Liu, H., Wu, Q., Mu, F., Yang, J., Gao, J., Li, C., and Lee, Y. J. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22511–22521, 2023.

Lin, H., Cho, J., Zala, A., and Bansal, M. Ctrl-adapter: An efficient and versatile framework for adapting diverse controls to any diffusion model. *arXiv preprint arXiv:2404.09967*, 2024a.

Lin, S., Wang, A., and Yang, X. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024b.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Liu, B., Akhgari, E., Visheratin, A., Kamko, A., Xu, L., Shrirao, S., Souza, J., Doshi, S., and Li, D. Playground v3: Improving text-to-image alignment with deep-fusion large language models. *arXiv preprint arXiv:2409.10695*, 2024a.

Liu, N., Li, S., Du, Y., Torralba, A., and Tenenbaum, J. B. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pp. 423–439. Springer, 2022.

Liu, X., Zhang, X., Ma, J., Peng, J., et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2023.

Liu, Y., Cun, X., Liu, X., Wang, X., Zhang, Y., Chen, H., Liu, Y., Zeng, T., Chan, R., and Shan, Y. Evalcrafter: Benchmarking and evaluating large video generation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22139–22149, 2024b.

Liu, Y., Zhang, K., Li, Y., Yan, Z., Gao, C., Chen, R., Yuan, Z., Huang, Y., Sun, H., Gao, J., et al. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024c.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022.

Luo, H., Ji, L., Zhong, M., Chen, Y., Lei, W., Duan, N., and Li, T. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304, 2022.

Luo, S., Tan, Y., Patil, S., Gu, D., von Platen, P., Passos, A., Huang, L., Li, J., and Zhao, H. Lcm-lora: A universal stable-diffusion acceleration module. *arXiv preprint arXiv:2311.05556*, 2023a.

Luo, Z., Chen, D., Zhang, Y., Huang, Y., Wang, L., Shen, Y., Zhao, D., Zhou, J., and Tan, T. Videofusion: Decomposed diffusion models for high-quality video generation. *arXiv preprint arXiv:2303.08320*, 2023b.

Lyu, M., Yang, Y., Hong, H., Chen, H., Jin, X., He, Y., Xue, H., Han, J., and Ding, G. One-dimensional adapter to rule them all: Concepts diffusion models and erasing applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7559–7568, 2024.

Ma, Q., Ning, X., Liu, D., Niu, L., and Zhang, L. Decouple-then-merge: Towards better training for diffusion models. *ArXiv*, abs/2410.06664, 2024.

Mangrulkar, S., Gugger, S., Debut, L., Belkada, Y., Paul, S., and Bossan, B. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022.

Mou, C., Wang, X., Xie, L., Wu, Y., Zhang, J., Qi, Z., and Shan, Y. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 4296–4304, 2024.

Nan, K., Xie, R., Zhou, P., Fan, T., Yang, Z., Chen, Z., Li, X., Yang, J., and Tai, Y. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. *arXiv preprint arXiv:2407.02371*, 2024.

Nguyen, T. H. and Tran, A. Swiftbrush: One-step text-to-image diffusion model with variational score distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7807–7816, 2024.

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

Park, B., Go, H., Kim, J.-Y., Woo, S., Ham, S., and Kim, C. Switch diffusion transformer: Synergizing denoising tasks with sparse mixture-of-experts. In *European Conference on Computer Vision*, 2024.

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

Peebles, W. S. and Xie, S. Scalable diffusion models with transformers. *2023 IEEE/CVF International Conference on Computer Vision*, pp. 4172–4182, 2022. URL https://api.semanticscholar.org/CorpusID:254854389.

Peng, B., Wang, J., Zhang, Y., Li, W., Yang, M.-C., and Jia, J. Controlnext: Powerful and efficient control for image and video generation. *arXiv preprint arXiv:2408.06070*, 2024.

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021a. URL https://api.semanticscholar.org/CorpusID:231591445.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021b.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. In *International conference on machine learning*, pp. 8821–8831. Pmlr, 2021.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Ran, L., Cun, X., Liu, J.-W., Zhao, R., Zijie, S., Wang, X., Keppo, J., and Shou, M. Z. X-adapter: Adding universal compatibility of plugins for upgraded diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8775–8784, 2024.

Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Susano Pinto, A., Keysers, D., and Houlsby, N. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34: 8583–8595, 2021.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022a.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022b.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., and Aberman, K. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35: 36479–36494, 2022.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35: 25278–25294, 2022.

Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Machine Learning*, 2021.

Sun, H., Lei, T., Zhang, B., Li, Y., Huang, H., Pang, R., Dai, B., and Du, N. Ec-dit: Scaling diffusion transformers with adaptive expert-choice routing. *ArXiv*, abs/2410.02098, 2024.

Wang, J., Yuan, H., Chen, D., Zhang, Y., Wang, X., and Zhang, S. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023.

Wang, Z., Sha, Z., Ding, Z., Wang, Y., and Tu, Z. Tokencompose: Text-to-image diffusion with token-level supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8553–8564, 2024.

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

Xing, Z., Dai, Q., Hu, H., Wu, Z., and Jiang, Y.-G. Simda: Simple diffusion adapter for efficient video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7827–7839, 2024.

Xu, C., Song, T., Feng, W., Li, X., Ge, T., Zheng, B., and Wang, L. Accelerating image generation with sub-path linear approximation model. *arXiv preprint arXiv:2404.13903*, 2024a.

Xu, Y., Zhao, Y., Xiao, Z., and Hou, T. Ufogen: You forward once large scale text-to-image generation via diffusion gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8196–8206, 2024b.

Xue, Z., Song, G., Guo, Q., Liu, B., Zong, Z., Liu, Y., and Luo, P. Raphael: Text-to-image generation via large mixture of diffusion paths. *Advances in Neural Information Processing Systems*, 36, 2024.

Ye, H., Zhang, J., Liu, S., Han, X., and Yang, W. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

Yin, T., Gharbi, M., Park, T., Zhang, R., Shechtman, E., Durand, F., and Freeman, W. T. Improved distribution matching distillation for fast image synthesis. *arXiv preprint arXiv:2405.14867*, 2024a.

Yin, T., Gharbi, M., Zhang, R., Shechtman, E., Durand, F., Freeman, W. T., and Park, T. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6613–6623, 2024b.

Yu, J., Wang, Y., Zhao, C., Ghanem, B., and Zhang, J. Freedom: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23174–23184, 2023.

Zhang, H., Lu, Y., Alkhouri, I., Ravishankar, S., Song, D., and Qu, Q. Improving training efficiency of diffusion models via multi-stage framework and tailored multi-decoder architecture. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7372–7381, 2024.

Zhang, L., Rao, A., and Agrawala, M. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023.

Zhang, Y. and Hooi, B. Hipa: Enabling one-step text-to-image diffusion models via high-frequency-promoting adaptation. *arXiv preprint arXiv:2311.18158*, 2023.

Zhao, W., Bai, L., Rao, Y., Zhou, J., and Lu, J. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

Zhuang, S., Li, K., Chen, X., Wang, Y., Liu, Z., Qiao, Y., and Wang, Y. Vlogger: Make your dream a vlog. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8806–8817, 2024.

A. More Implementation Details

A.1. LoRA and TSM Training Configuration on Domain Adaptation

A.1.1. Stable Diffusion 1.5 (Rombach et al., 2022a)

For SD1.5, in vanilla LoRA and TSM 1-stage, we employ LoRA on the *to_q*, *to_k*, *to_v* and *to_out.0* modules of the UNet and *q_proj* and *v_proj* modules of CLIP text encoders. We set LoRA r, α=4, and employ zero initialization for all matrix B. The learning rate is 1e-4 and the weight decay is 1e-2 for both UNet and text encoder. At TSM assembling stage (2-stage), we add router to the module which is equipped with LoRA and set TimeStep experts n_1=8, n_2=1. We train 4K steps for vanilla LoRA and two stages of TSM. The global batch size is 64. We use the AdamW optimizer with β_1=0.9, β_2=0.999. For both UNet and text encoder, the learning rate is set to 1e-4 and the weight decay to 1e-2.

A.1.2. Stable Diffusion 2 (Rombach et al., 2022a)

For SD2, in vanilla LoRA and TSM 1-stage, we employ LoRA on the *to_q*, *to_k*, *to_v* and *to_out.0* modules of the UNet and *q_proj* and *v_proj* modules of CLIP text encoders. We set LoRA r, α=4, and employ zero initialization for all matrix B. The learning rate is 1e-4 and the weight decay is 1e-2 for both UNet and text encoder. At TSM assembling stage (2-stage), we add router to the module which is equipped with LoRA and set TimeStep experts n_1=8, n_2=1. We train 4K steps for vanilla LoRA and two stages of TSM. The global batch size is 64. We use the AdamW optimizer with β_1=0.9, β_2=0.999. For both UNet and text encoder, the learning rate is set to 1e-4 and the weight decay to 1e-2.

A.1.3. PixArt-α (Chen et al., 2024d)

In vanilla LoRA and TSM 1-stage, we employ LoRA on the *to_q*, *to_k*, *to_v* and *to_out.0* modules of the DiT (Peebles & Xie, 2022) and *q,v* modules of T5 text encoder (Raffel et al., 2020). We set LoRA r, α=4, and employ zero initialization for all matrix B. The learning rate is 2e-5 and the weight decay is 1e-2 for both DiT and text encoder. At TSM assembling stage (2-stage), we add router to the module which is equipped with LoRA and set TimeStep experts n_1=8, n_2=1. We train 4K steps for vanilla LoRA and two stages of TSM. The global batch size is 64. We use the AdamW optimizer with β_1=0.9, β_2=0.999. For both DiT and text encoder, the learning rate is set to 2e-5 and the weight decay to 1e-2.

A.1.4. Stable Diffusion 3 (Esser et al., 2024b)

For SD3, in vanilla LoRA and TSM fostering stage (1-stage), we employ LoRA on the *to_q*, *to_k*, *to_v* and *to_out.0* modules of the MM-DiT and *q_proj*, *k_proj*, *v_proj* and *out_proj* modules of two CLIP text encoders (Radford et al., 2021a; Cherti et al., 2023). We set LoRA r, α=4, and employ zero initialization for all matrix B. At TSM assembling stage (2-stage), we add router to the module which is equipped with LoRA and set TimeStep experts n_1=8, n_2=1. We train 4K steps for vanilla LoRA and two stages of TSM. The global batch size is 64. We use the AdamW optimizer with β_1=0.9, β_2=0.999. For MM-DiT, the learning rate is set to 1e-5 and the weight decay to 1e-4. For text encoder, the learning rate is set to 5e-6 and the weight decay to 1e-3.

A.2. LoRA and TSM Training Configuration on Post Pretraining

A.2.1. PixArt-α (Chen et al., 2024d)

In vanilla LoRA and TSM 1-stage, we employ LoRA on the *to_q*, *to_k*, *to_v* and *to_out.0* modules of the DiT (Peebles & Xie, 2022) and *q,v* modules of T5 text encoder (Raffel et al., 2020). We set LoRA r, α=4, and employ zero initialization for all matrix B. The learning rate is 2e-5 and the weight decay is 1e-2 for both DiT and text encoder. At TSM assembling stage (2-stage), we add router to the module which is equipped with LoRA and set TimeStep experts n_1=8, n_2=1. We train 4K steps for vanilla LoRA and two stages of TSM. The global batch size is 64. We use the AdamW optimizer with β_1=0.9, β_2=0.999. For both DiT and text encoder, the learning rate is set to 2e-5 and the weight decay to 1e-2.

A.2.2. VideoCrafter2 (Chen et al., 2024a)

In vanilla LoRA and TSM 1-stage, we inject LoRA on the *k, v* modules in both spatial and temporal layers of the 3D-UNet and *out_proj* module of OpenCLIP (Cherti et al., 2023) text encoder. We set LoRA r, α=16 and adopt $lora\_dropout$=0.01 only in the 3D-UNet. In TSM 2-stage, we add router to the module where LoRA is injected and set TimeStep experts n_1=8, n_2=4. We train 5K steps for vanilla LoRA and two stages of TSM. The global batch size is 32. We use the same

optimizer setting as in image modality. The learning rate is 2e-4 and the weight decay is 1e-2 for both UNet and text encoder.

A.3. LoRA and TSM Training Configuration for Stable Diffusion 1.5 (Rombach et al., 2022a) on Model Distillation

We employ LoRA with r=64, α=8 on *to_q*, *to_k*, *to_v*, *to_out.0*, *proj_in*, *proj_out*, *ff.net.0.proj*, *ff.net.2*, *conv1*, *conv2*, *conv_shortcut*, *downsamplers.0.conv*, *upsamplers.0.conv* and *time_emb_proj* modules of UNet. In vanilla LoRA, we train for 40K steps without GAN loss and 5K steps with it. In TSM 1-stage, we train the experts at 999 and 749 timesteps for 20K steps without GAN loss and 5K steps with it. At 499 and 249 timesteps, we reduce training without GAN loss to 5K steps and increase training with real image guidance to 20K and 40K steps respectively. In TSM 2-stage, we train the router and freeze other modules with n_1=4, n_2=1 TimeStep experts. We only train it for 2K steps with GAN loss, due to the little N_{params} (<1M). The batch size is 32 without GAN loss and 16 with it (4 times for vanilla LoRA). Other settings are consistent with DMD2.

A.4. LoRA and TSM Training Configuration on Overrall Design Ablation

A.4.1. Stable Diffusion 1.5 (Rombach et al., 2022a)

For SD1.5, in vanilla LoRA and TSM 1-stage, we employ LoRA on the *to_q*, *to_k*, *to_v* and *to_out.0* modules of the UNet and *q_proj* and *v_proj* modules of CLIP text encoders. We set LoRA r, α=4, and employ zero initialization for all matrix B. The learning rate is 1e-4 and the weight decay is 1e-2 for both UNet and text encoder. At TSM assembling stage (2-stage), we add router to the module which is equipped with LoRA and set TimeStep experts n_1=8, n_2=1. We train 4K steps for vanilla LoRA and two stages of TSM. The global batch size is 64. We use the AdamW optimizer with β_1=0.9, β_2=0.999. For both UNet and text encoder, the learning rate is set to 1e-4 and the weight decay to 1e-2.

A.4.2. PixArt-α (Chen et al., 2024d)

In vanilla LoRA and TSM 1-stage, we employ LoRA on the *to_q*, *to_k*, *to_v* and *to_out.0* modules of the DiT (Peebles & Xie, 2022) and *q,v* modules of T5 text encoder (Raffel et al., 2020). We set LoRA r, α=4, and employ zero initialization for all matrix B. The learning rate is 2e-5 and the weight decay is 1e-2 for both DiT and text encoder. At TSM assembling stage (2-stage), we add router to the module which is equipped with LoRA and set TimeStep experts n_1=8, n_2=1. We train 4K steps for vanilla LoRA and two stages of TSM. The global batch size is 64. We use the AdamW optimizer with β_1=0.9, β_2=0.999. For both DiT and text encoder, the learning rate is set to 2e-5 and the weight decay to 1e-2.

A.4.3. Stable Diffusion 3 (Esser et al., 2024b)

For SD3, in vanilla LoRA and TSM fostering stage (1-stage), we employ LoRA on the *to_q*, *to_k*, *to_v* and *to_out.0* modules of the MM-DiT and *q_proj*, *k_proj*, *v_proj* and *out_proj* modules of two CLIP text encoders (Radford et al., 2021a; Cherti et al., 2023). We set LoRA r, α=4, and employ zero initialization for all matrix B. At TSM assembling stage (2-stage), we add router to the module which is equipped with LoRA and set TimeStep experts n_1=8, n_2=1. We train 4K steps for vanilla LoRA and two stages of TSM. The global batch size is 64. We use the AdamW optimizer with β_1=0.9, β_2=0.999. For MM-DiT, the learning rate is set to 1e-5 and the weight decay to 1e-4. For text encoder, the learning rate is set to 5e-6 and the weight decay to 1e-3.

A.4.4. Stable Diffusion 1.5 (Rombach et al., 2022a)

For SD1.5, in vanilla LoRA and TSM 1-stage, we employ LoRA on the *to_q*, *to_k*, *to_v* and *to_out.0* modules of the UNet and *q_proj* and *v_proj* modules of CLIP text encoders. We set LoRA r, α=4, and employ zero initialization for all matrix B. The learning rate is 1e-4 and the weight decay is 1e-2 for both UNet and text encoder.

A.4.5. PixArt-α (Chen et al., 2024d)

In vanilla LoRA and TSM 1-stage, we employ LoRA on the *to_q*, *to_k*, *to_v* and *to_out.0* modules of the DiT (Peebles & Xie, 2022) and *q,v* modules of T5 text encoder (Raffel et al., 2020). We set LoRA r, α=4, and employ zero initialization for all matrix B. The learning rate is 2e-5 and the weight decay is 1e-2 for both DiT and text encoder.

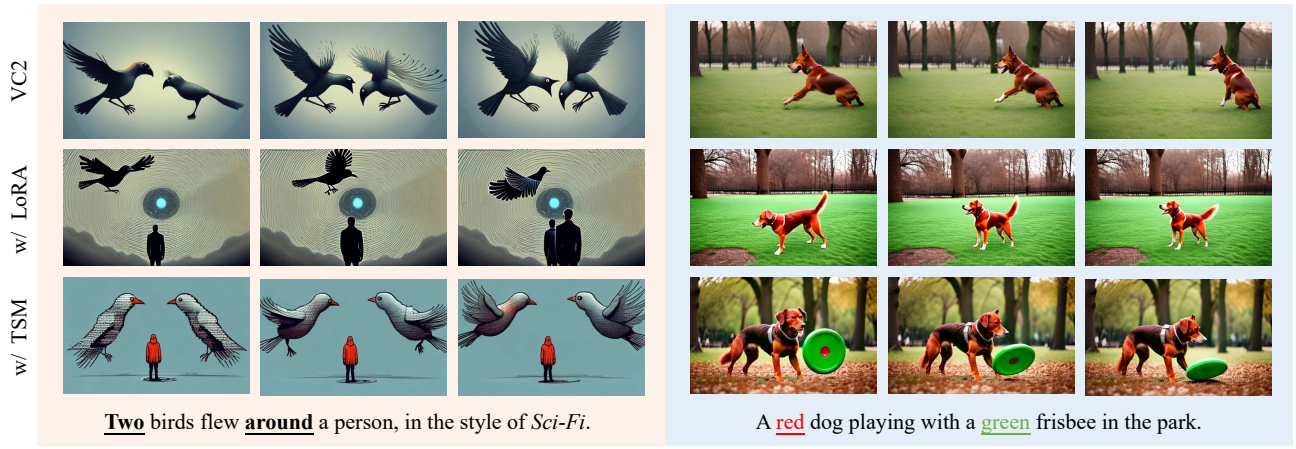

Figure 4. **Comparison on Video Modality.** The videos generated by the LoRA-tuned model are not aligned with the prompts, while our TSM facilitates high-quality and consistent video generation.

A.4.6. STABLE DIFFUSION 3 (ESSER ET AL., 2024B)

For SD3, in vanilla LoRA and TSM fostering stage (1-stage), we employ LoRA on the *to_q*, *to_k*, *to_v* and *to_out.0* modules of the MM-DiT and *q_proj*, *k_proj*, *v_proj* and *out_proj* modules of two CLIP text encoders (Radford et al., 2021a; Cherti et al., 2023). We set LoRA r, α=4, and employ zero initialization for all matrix B.

B. More Visualization result

Figure 5. **Comparison on Model Distillation.** The images generated by our TSM better align with the prompts, outperforming the vanilla LoRA, and even surpassing the teacher SD1.5 in some cases.

Method	FID ↓
PixArt-α	23.15
PixArt-α + TSM	**16.78**

Table 12. **Post-pretraining Evaluation on COCO2014.**

Method	Color↑	Shape↑	Texture↑	Spatial↑	Non-Spatial↑	Complex↑
SD1.5 (Rombach et al., 2022b)	36.97	36.27	41.25	11.04	31.05	30.79
SD1.5 + MoE LoRA (Li et al., 2024a)	52.26	39.83	51.38	11.71	31.34	32.00
SD1.5+TSM 1-stage	56.48	34.91	57.08	18.01	31.77	32.79
SD1.5+TSM 2-stage (Symmetrical)	54.56	45.52	56.30	17.9	31.78	**33.27**
SD1.5+TSM 2-stage (Asymmetrical)	**57.59**	**46.18**	**57.69**	**17.91**	**31.82**	32.78

Table 13. **TSM and MoE LoRA Ablation on T2I-CompBench.**

B.1. Post-Pretraining

As shown in Fig. 1 (c), in the post-pretraining task, the TSM fine-tuned model improves the alignment between images/videos and text without degrading visual quality, while the LoRA fine-tuned model exhibits a significant decline in both visual quality and vision-text alignment.

B.2. Model Distillation

As shown in Fig. 5, in model distillation task, the TSM fine-tuned model is more aligned with the prompts, outperforming LoRA.

C. More Experiments

C.1. Post-Pretraining Model FID Evaluation

Benchmark. We conduct experiments on 30K prompts from COCO2014 (Lin et al., 2014) validation set. We generate images from these prompts and compare these images with 40,504 real images from the same validation set to calculate the Fréchet Inception Distance (FID) (Heusel et al., 2017).

Post-pretraining Configuration. We perform post-pretraining based on the pre-trained PixArt-α (Chen et al., 2024d). In vanilla LoRA and TSM 1-stage, we employ LoRA on the *to_q*, *to_k*, *to_v* and *to_out.0* modules of the DiT (Peebles & Xie, 2022) and *q,v* modules of T5 text encoder (Raffel et al., 2020). We set LoRA r, α=4, and employ zero initialization for all matrix B. The learning rate is 2e-5 and the weight decay is 1e-2 for both DiT and text encoder. At TSM assembling stage (2-stage), we add router to the module which is equipped with LoRA and set TimeStep experts n_1=8, n_2=1. We train 4K steps for vanilla LoRA and two stages of TSM. The global batch size is 64. We use the AdamW optimizer with β_1=0.9, β_2=0.999. For both DiT and text encoder, the learning rate is set to 2e-5 and the weight decay to 1e-2.

As shown in Tab. 12, after using TSM to fine-tune the original pre-training model on the original pre-training dataset, the zero-shot generation capability of the pre-training model can still be improved.

C.2. Ablation Experiments on TSM and MoE LoRA

Benchmark. We conduct TSM and MoE LoRA ablation experiments on T2I-CompBench (Huang et al., 2023) based on pre-trained Stable Diffusion 1.5 (Rombach et al., 2022a) for domain adaptation task. We set up four experiments with different settings for ablation: MoE LoRA, TSM 1-stage, using the traditional symmetric MoE architecture to ensemble multiple LoRAs obtained from TSM 1-stage, and TSM 2-stage.

Domain Adaptation Configuration. We conduct experiments on the *to_q*, *to_k*, *to_v* and *to_out.0* modules of the UNet and *q_proj* and *v_proj* modules of CLIP text encoders. We set LoRA r, α=4, and employ zero initialization for all matrix B. The learning rate is 1e-4 and the weight decay is 1e-2 for both UNet and text encoder. At TSM assembling stage (2-stage), we add router to the module which is equipped with LoRA and set TimeStep experts n_1=8, n_2=1. We train 4K steps for vanilla LoRA and two stages of TSM. The global batch size is 64. We use the AdamW optimizer with β_1=0.9, β_2=0.999. For both UNet and text encoder, the learning rate is set to 1e-4 and the weight decay to 1e-2.

The results of the ablation experiment show that the effect of our TSM is far better than that of MoE LoRA, and the asymmetric MoE architecture also has obvious advantages compared to the symmetric model.