Diffusion Model Patching via Mixture-of-Prompts

Seokil Ham[1]*, Sangmin Woo[1]*, Jin-Young Kim[2], Hyojun Go[2], Byeongjun Park[1], Changick Kim[1],

[1]KAIST
[2]Twelve Labs
[1]{gkatjrdlf, smwoo95, pbj3810, changick}@kaist.ac.kr, [2]{seago0828, gohyojun15}@gmail.com

Abstract

We present Diffusion Model Patching (DMP), a simple method to boost the performance of pre-trained diffusion models that have *already reached convergence*, with a negligible increase in parameters. DMP inserts a small, learnable set of prompts into the model's input space while keeping the original model frozen. The effectiveness of DMP is not merely due to the addition of parameters but stems from its dynamic gating mechanism, which selects and combines a subset of learnable prompts at every timestep (i.e., reverse denoising steps). This strategy, which we term "mixture-of-prompts", enables the model to draw on the distinct expertise of each prompt, essentially "patching" the model's functionality at every timestep with minimal yet specialized parameters. Uniquely, DMP enhances the model by further training on the original dataset already used for pre-training, even in a scenario where significant improvements are typically not expected due to model convergence. Notably, DMP significantly enhances the FID of converged DiT-L/2 by 10.38% on FFHQ, achieved with only a 1.43% parameter increase and 50K additional training iterations.

Project Page — https://sangminwoo.github.io/DMP/
Extended Paper — https://arxiv.org/abs/2405.17825

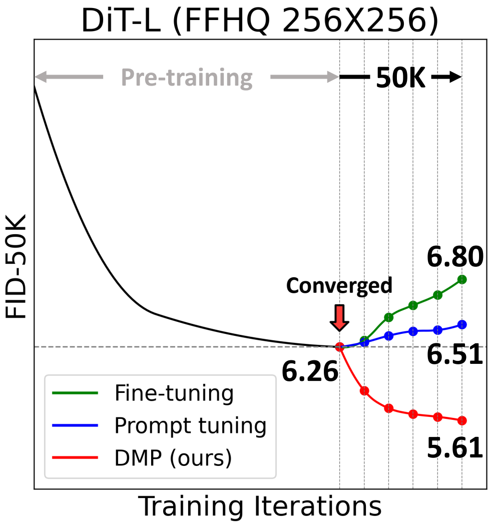


Figure 1: **Further training** of the *fully converged* DiT-L/2 model using the *same dataset as the pre-training phase*. Our method, DMP achieves a 10.38% FID improvement in just 50K iterations, while other methods exhibit overfitting.

Introduction

The rapid progress in generative modeling has been largely driven by the advancement of diffusion models (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020), which have gained attention for their desirable properties, such as stable training, smooth model scaling, and good mode coverage (Nichol and Dhariwal 2021). Diffusion models have set new standards in generating high-quality, diverse samples that closely match the distribution of various datasets (Dhariwal and Nichol 2021; Ramesh et al. 2021; Saharia et al. 2022b; Poole et al. 2022).

Diffusion models are characterized by their multi-step denoising process, which progressively refines random noise into structured outputs, such as images. Each step aims to denoise a noised input, gradually converting completely random noise into a meaningful image. Despite all denoising steps share the same goal of generating high-quality images, each step has distinct characteristics that contribute to shaping the final output (Go et al. 2023; Park et al. 2023). The visual concepts that diffusion models learn vary based on the noise ratio of input (Choi et al. 2022). At higher noise levels (timestep t is close to T), where images are highly corrupted and thus contents are unrecognizable, the models focus on recovering global structures and colors. As the noise level decreases and images become less corrupted (timestep t is close to 0), the task of recovering images becomes more straightforward, and diffusion models learn to recover fine-grained details. Recent studies (Balaji et al. 2022; Choi et al. 2022; Go et al. 2023; Park et al. 2023) suggest that considering *stage-specificity* is beneficial, as it aligns better with the nuanced requirements of different stages in the generation process. However, many existing diffusion models do not explicitly consider this aspect.

Our goal is to enhance *already converged* diffusion models by introducing stage-specific capabilities using the same

*These authors contributed equally.

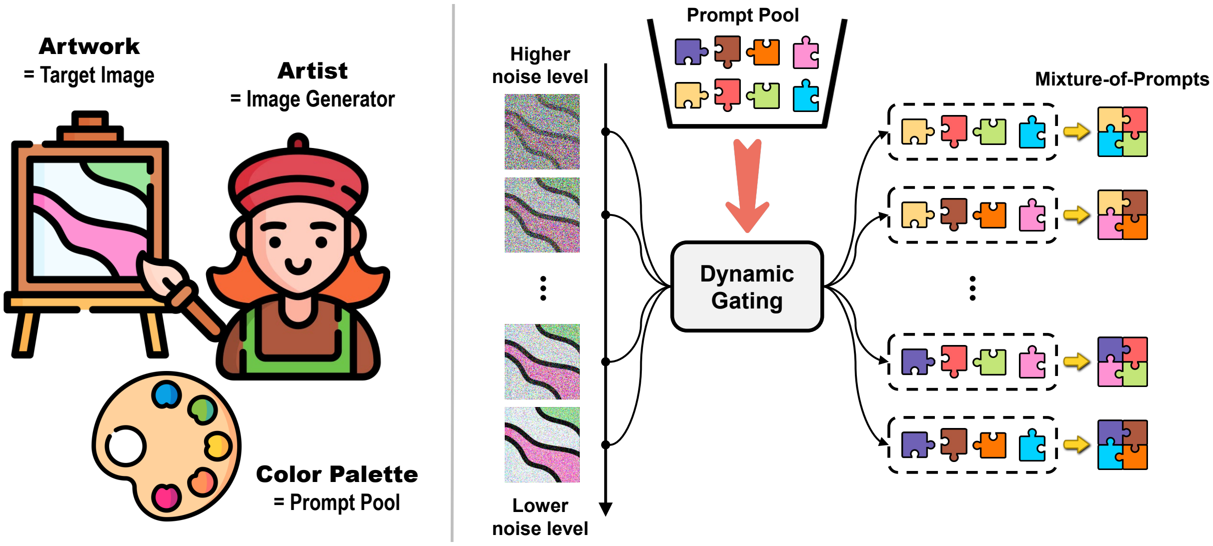


Figure 2: **Overview of DMP**. We take inspiration from prompt tuning (Lester, Al-Rfou, and Constant 2021) and aim to enhance *already converged* diffusion models. Our approach incorporates a pool of prompts within the input space, with each prompt learned to excel at certain stages of the denoising process. At every step, a unique blend of prompts (*i.e.*, mixture-of-prompts) is constructed via dynamic gating based on the current noise level. This mechanism is similar to an skilled artist choosing the appropriate color combinations to refine different aspects of their artwork for specific moments. Importantly, our method keeps the diffusion model itself unchanged, and only use the original training dataset for further training.

pre-training dataset. We propose Diffusion Model Patching (DMP), a method that equips pre-trained diffusion models with an enhanced toolkit, enabling them to navigate the generation process with greater finesse and precision. An overview of DMP is shown in Fig. 2. DMP consists of two main components: **(1) Learnable prompts**: A small pool of prompts (Lester, Al-Rfou, and Constant 2021), optimized for particular stages of the denoising process, act as "experts" for certain denoising steps (or noise levels). This lightweight addition adjusts only small parameters at the input space and enables the model to be directed towards specific behaviors for each stage without retraining the entire model. **(2) Dynamic gating mechanism:** This mechanism creates *mixture-of-prompts* by adaptively combining prompts based on the noise level of the input, enabling the model to leverage specialized knowledge for each stage.

By incorporating these components, we continue training the converged diffusion models using the original dataset on which they were pre-trained. Given that the model has *already converged*, it is generally assumed that conventional fine-tuning would not lead to significant improvements or may even cause overfitting. However, DMP provides the model with a nuanced understanding of each denoising step, leading to enhanced performance, even when trained on the same data distribution. As shown in Fig. 1, it boosts the performance of DiT-L/2 (Peebles and Xie 2022) by 10.38% with only 50K iterations on the FFHQ dataset (Karras, Laine, and Aila 2019).

While simple, DMP offers several key strengths:

❶ **Data Efficiency**: DMP boosts model performance using the original dataset, without requiring any external datasets. This is particularly noteworthy as further training of already converged diffusion models on the same dataset typically does not lead to performance gains. DMP differs from general fine-tuning (Deng et al. 2009), which often transfers knowledge across different datasets.

❷ **Computational Efficiency**: DMP patches pre-trained diffusion models by slightly modifying their input space, without updating the model itself. DMP contrasts with methods that train diffusion models from scratch for denoising-stage-specificity (Choi et al. 2022; Hang et al. 2023; Go et al. 2023; Park et al. 2023, 2024), which can be computationally expensive and storage-intensive.

❸ **Parameter Efficiency**: DMP adds only a negligible number of parameters, approximately 1.43% of the total model parameters (based on DiT-L/2). This ensures that performance enhancements are achieved cost-effectively.

❹ **Simplified Training**: DMP eliminates the need to train multiple expert networks for different denoising stages. Instead, it uses a few prompts to learn nuanced behaviors specific to each step. This simplifies the model architecture and training process compared to prior methods (Balaji et al. 2022; Feng et al. 2023; Xue et al. 2023; Park et al. 2024).

Related Work

Diffusion models with stage-specificity. Diffusion models (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2020) have shown remarkable versatility across diverse modalities, including images (Ramesh et al. 2021; Saharia et al. 2022a), audios (Kong et al. 2020), texts (Li et al. 2022) and 3D (Woo et al. 2023). Recent efforts to improve stage-specificity in diffusion models have advanced both *architectural* and *optimization* fronts: **(1)** On the **architectural** front, eDiff-I (Balaji et al. 2022), ERNIE-ViLG 2.0 (Feng et al. 2023), and RAPHAEL (Xue et al. 2023) introduced the concept of utilizing multiple expert denoisers, each tailored to specific noise levels, thereby augmenting the model's capacity. DTR (Park et al. 2023) refined diffusion model architectures by allocating different channel combinations for each denoising step. **(2)** From an **optimization** perspective, P2 Weight (Choi et al. 2022) and Min-SNR Weight (Hang et al. 2023) accelerated convergence by framing diffusion training as a multi-task learning problem (Caruana 1997), where loss weights are adjusted based on task difficulty at each timestep. Go *et al.* (Go et al. 2023) mitigated learning conflicts of multiple denoising stages by clustering similar stages based on their signal-to-noise ratios (SNRs). While these approaches often require training from scratch or using multiple expert networks—demanding significant computational and storage resources—our method achieves stage-specificity by leveraging a single pre-trained model without modifying its parameters.

Parameter-efficient Fine-tuning (PEFT) in Diffusion models. PEFT offers a way to enhance models by tuning a small number of (extra) parameters, avoiding the need to retrain the entire model and reducing computational/storage costs (Xiang et al. 2023). This is particularly appealing given the complexity and parameter-dense nature of diffusion models (Rombach et al. 2022; Peebles and Xie 2022), where directly training diffusion models from scratch is impractical. Recent advancements in this field can be broadly categorized into three streams: **(1)** T2i-Adapter (Mou et al. 2023), SCEdit (Jiang et al. 2023), ControlNet (Zhang, Rao, and Agrawala 2023) and CDMs (Golatkar et al. 2023) utilize **adapters** (Houlsby et al. 2019; Hu et al. 2021; Chen et al. 2022) or **side-tuning** (Zhang et al. 2020; Sung, Cho, and Bansal 2022) to modify the neural network's behavior at specific layers. **(2)** Textual Inversion (Hertz et al. 2022; Gal et al. 2022) use a concept similar to **prompt tuning** (Li and Liang 2021; Lester, Al-Rfou, and Constant 2021; Jia et al. 2022; Zhou et al. 2022) that modifies input or textual representations to influence subsequent processing without changing the function itself. **(3)** CustomDiffusion (Kumari et al. 2023), SVDiff (Han et al. 2023), and DiffFit (Xie et al. 2023) focus on **partial parameter tuning** (Zaken, Ravfogel, and Goldberg 2021; Xu et al. 2021; Lian et al. 2022), fine-tuning specific parameters of the neural network, such as bias terms. These methods have shown success in personalizing models with new datasets. In contrast, our approach targets in-domain performance enhancements, improving pre-trained diffusion models using their original datasets while remaining computationally efficient.[1]

Diffusion Model Patching (DMP)

We propose DMP, a simple yet effective method to enhance *already converged* diffusion models by leveraging leveraging stage-specific knowledge during the denoising process.[2] DMP fine-tunes models using their original pre-training dataset without altering backbone parameters, enabling them to adapt to the nuanced requirements of different denoising stages. DMP comprises two key components: (1) a pool of learnable prompts and (2) a dynamic gating mechanism. First, a small number of learnable prompts are attached to the model's input space. Second, the dynamic gating mechanism blends prompts dynamically based on the noise levels of the input image. The overall framework of DMP is shown in Fig. 3.

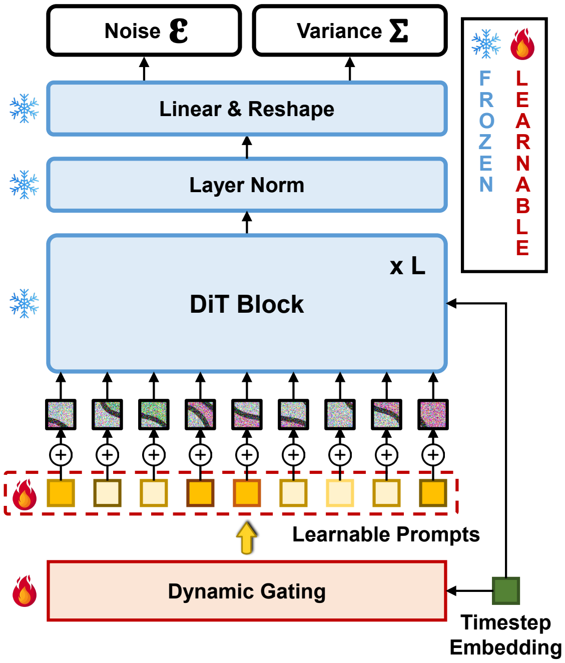


Figure 3: **DMP framework with DiT (Peebles and Xie 2022).** DMP adaptively generates optimal prompts for specific timesteps. It fine-tunes diffusion models using the original pre-training dataset and operates entirely through prompt-based tuning in the input space. Our DMP requires no modifications to the model architecture or training process, ensuring seamless integration and efficiency.

Motivation. Each denoising stage involves different levels of difficulty and goals, depending on the noise level (Choi et al. 2022; Hang et al. 2023; Go et al. 2023; Park et al. 2023). Prompt tuning (Li and Liang 2021; Lester, Al-Rfou, and Constant 2021; Jia et al. 2022) assumes that if a pre-trained model already has sufficient knowledge, carefully constructed prompts can extract knowledge for a specific downstream task from the frozen model. Likewise, we hypothesize that pre-trained diffusion models inherently possess knowledge for all denoising stages. By learning a mixture-of-prompts specific to each stage, we can *patch* models with stage-specific expertise.

Architecture. As our base architecture, we employ DiT (Peebles and Xie 2022) (a transformer-based model) and Stable Diffusion (Rombach et al. 2022) (a UNet-based model), both operating in the latent space. Input images are processed into latent codes (*e.g.*, $32 \times 32 \times 4$ for $256 \times 256 \times 3$ images) using a pre-trained VAE (Kingma and Welling 2013). Learnable prompts are then sized to match the in-

[1] More related work is in Appendix of the extended paper.

[2] See preliminaries in Appendix of the extended paper.

put dimensions ($N \times D$ for DiT, $H \times W \times D$ for Stable Diffusion).

Learnable prompts. Our goal is to efficiently enhance the model with denoising-stage-specific knowledge, adjusting small parameters within the input space. To achieve this, we start with a pre-trained DiT model as an example for explanation. Firstly, we insert N learnable continuous embeddings of dimension D, i.e., *prompts*, into the input space of each DiT block. The set of learnable prompts is denoted as:

$$\boldsymbol{P} = \{\boldsymbol{p^{(i)}} \in \mathbb{R}^{N \times D} \mid i \in \{0, \ldots, L-1\}\}. \qquad (1)$$

Here, $\boldsymbol{p^{(i)}}$ denotes the prompts for the i-th DiT block and L is the total number of DiT blocks. Unlike prior methods (Li and Liang 2021; Lester, Al-Rfou, and Constant 2021; Jia et al. 2022; Wang et al. 2022), that prepend prompts to the input sequence, we directly add them to the input, avoiding an increase in sequence length and maintaining nearly the same computation speed as before. Moreover, each prompt added to the input patch provides a direct signal to help denoise specific spatial parts. This design choice allows the model to focus on different aspects of the input at each timestep, aiding in specialized denoising steps. The output of i-th DiT block is computed as:

$$\boldsymbol{x^{(i+1)}} = \text{Block}^{(i)}_{frozen}(\boldsymbol{p^{(i)}_{learn}} + \boldsymbol{x^{(i)}}), \qquad (2)$$

where only the prompts are updated during training, while the backbone parameters remain frozen.

Dynamic gating. In Eq. 2, the same prompts are used throughout the training, thus they will learn denoising-stage-agnostic knowledge. To patch the model with stage-specific knowledge, we introduce dynamic gating. This mechanism blends prompts in varying proportions based on the noise level of an input image. Specifically, we use a timestep embedding $\boldsymbol{t}$ to represent the noise level at each step of the generation process. For a given $\boldsymbol{t}$, the gating network $\mathcal{G}$ creates the stage-specific mask of shape $N \times 1$ used for generating mixtures-of-prompts, thereby redefining Eq. 2 as:

$$\boldsymbol{x^{(i+1)}} = \text{Block}^{(i)}_{frozen}\big(\sigma(\mathcal{G}_{learn}([\boldsymbol{t}; i])) \odot \boldsymbol{p}^{(i)}_{learn} + \boldsymbol{x}^{(i)}\big), \qquad (3)$$

where σ is the softmax function and $\odot$ denotes element-wise multiplication. In practice, $\mathcal{G}$ is implemented as a simple linear layer. It additionally takes the DiT block depth i as input to produce different results based on the depth. This dynamic gating mechanism effectively handles varying noise levels using only a small number of prompts. It also provides the model with the flexibility to use different sets of prompt knowledge at different stages of the generation process.

Training

Zero-initialization. We empirically found that random initialization of prompts disrupts the early training process, leading to instability and divergence. To ensure stable further training of a pre-trained diffusion model, we start by zero-initializing the prompts. With the prompt addition strategy that we selected before, zero-initialization prevents harmful noise from affecting the deep features of neural network layers and preserves the original knowledge at the beginning of training. As training progresses, the model gradually incorporate additional signals from the prompts.

Model	#Parameters
DiT-B/2	130M
+ DMP	132.5M (+1.96%)
DiT-L/2	458M
+ DMP	464.5M (+1.43%)
SD v1.5	890M
+ DMP	892M (+0.32%)

Table 1: **Parameters.** Default image size is 256×256. SD v1.5 indicates Stable Diffusion v1.5 (Rombach et al. 2022).

Prompt balancing loss. We adopt two soft constraints from Shazeer *et al.* (Shazeer et al. 2017) to balance the activation of mixtures-of-prompts. (1) Load balancing: In a mixture-of-experts setup (Jacobs et al. 1991; Jordan and Jacobs 1994), Eigen *et al.* (Eigen, Ranzato, and Sutskever 2013) noted that once experts are selected, they tend to be consistently chosen. In our setup, the load balancing loss prevents the gating network $\mathcal{G}$ from overly favoring a few prompts with larger weights and encourages all prompts to uniformly selected. (2) Importance balancing: Despite having similar overall load, prompts might still be activated with imbalanced weights. For instance, one prompt might be assigned with larger weights for a few denoising steps, while another might have smaller weights for many steps. The load balancing loss ensures that prompts are activated with similar overall importance across all denoising steps.[3]

Prompt efficiency. Table 1 presents the model parameters for various versions of the DiT architecture (Peebles and Xie 2022) ranging from DiT-B/2 to DiT-L/2 (where "2" denotes the patch size K) and Stable Diffusion v1.5 with and without the DMP. Assuming a fixed 256×256 resolution, using the DMP increases DiT-B/2 parameters by 1.96%. For the largest model, Stable Diffusion v1.5, the use of DMP results in a 0.3% increase to 892M parameters. The proportion of DMP parameters to total model parameters decreases as the model size increases, allowing for tuning only a small number of parameters compared to the entire model.

Experiments

We evaluate the effectiveness of DMP on various image generation tasks using *already converged* pre-trained diffusion models. Unlike conventional fine-tuning or prompt tuning, the original dataset from the pre-training phase is used for further training. We evaluated image quality using FID (Heusel et al. 2017) score, which measures the distance between feature representations of generated and real images using an Inception-v3 model (Szegedy et al. 2016). [4]

Datasets & Tasks. We used three datasets for our experiments: (1) FFHQ (Karras, Laine, and Aila 2019) (for *unconditional image generation*) contains 70,000 training images of human faces. (2) MS-COCO (Lin et al. 2014) (for *text-to-image generation*) includes 82,783 training images and 40,504 validation images, each annotated with 5 descrip-

[3] For further details about prompt balancing loss, see Appendix.

[4] Implementation details are in Appendix of the extended paper.

Resolution (256 × 256)	FFHQ
Model	FID↓
Pre-trained (iter: 600K)	
DiT-B/2	6.27
Further Training (iter: 30K)	
+ Fine-tuning	$6.57_{(+0.30)}$
+ Prompt tuning	$6.81_{(+0.54)}$
+ DMP	$\mathbf{5.87_{(-0.40)}}$

Resolution (256 × 256)	COCO
Model	FID↓
Pre-trained (iter: 450K)	
DiT-B/2	7.33
Further Training (iter: 40K)	
+ Fine-tuning	$7.51_{(+0.18)}$
+ Prompt tuning	$7.37_{(+0.04)}$
+ DMP	$\mathbf{7.12_{(-0.21)}}$

Resolution (512 × 512)	Laion5B
Model	FID↓
Pre-trained (iter: 1.5M)	
Stable Diffusion v1.5	47.18
Further Training (iter: 50K)	
+ Fine-tuning	$52.99_{(+5.81)}$
+ Prompt tuning	$35.93_{(-11.25)}$
+ DMP	$\mathbf{35.44_{(-11.74)}}$

Table 2: **Evaluating pre-trained diffusion models with different further training methods.** Importantly, we use the same dataset as in the pre-training for further training. We set two baselines for comparison: (1) *full fine-tuning* to update the entire model parameters. (2) *naive prompt tuning* (Lester, Al-Rfou, and Constant 2021) (equivalent to Eq. 2). ↓: The lower the better.

FFHQ 256 × 256	Further training iterations					
DiT-L/2 *(iter: 250K)*	+0K	+10K	+20K	+30K	+40K	+50K
+ Fine-tuning	6.26	6.32	6.53	6.64	6.73	$6.80_{(+0.54)}$
+ Prompt tuning	6.26	6.30	6.36	6.40	6.42	$6.51_{(+0.25)}$
+ DMP	6.26	5.88	5.72	5.67	5.64	$\mathbf{5.61}_{(-0.65)}$

Table 3: **Comparison of further training techniques across iterations** on FFHQ 256×256.

tive captions. (3) Laion5B (Schuhmann et al. 2022) (for *Stable Diffusion*) consists of 5.85B image-text pairs, which is known to be used to train Stable Diffusion (Rombach et al. 2022).

Comparative Study

Effectiveness of DMP. In Tab. 2, we compare DMP against two further training baselines – (1) full fine-tuning and (2) naive prompt tuning (Lester, Al-Rfou, and Constant 2021) (equivalent to Eq. 2) – across various datasets for unconditional/conditional image generation tasks. We employ pre-trained DiT models (Peebles and Xie 2022) that have already reached full convergence as our backbone. To ensure that the observed enhancement is not due to cross-dataset knowledge transfer, we further train the models using the same dataset used for pre-training. As expected, fine-tuning does not provide additional improvements to an already converged model in all datasets and even result in overfitting. Naive prompt tuning also fails to improve performance in almost datasets and instead lead to a decrease in performance. DMP enhances the FID across all datasets with only 30∼50K iterations, enabling the model to generate images of superior quality. This indicates that the performance gains achieved by DMP are not merely a result of increasing parameters, but rather from its novel mixture-of-prompts strategy. This strategy effectively patches diffusion models to operate slightly differently at each denoising step. Moreover, the significant improvement on Stable Diffusion v1.5 (Rombach et al. 2022) with Laion5B (Schuhmann et al. 2022) demonstrates DMP's expandability to different architectures and resolutions.

case	FID↓
attention	6.41
linear	**5.87**

(a) **Gating architecture**

case	FID↓
hard	5.96
soft	**5.87**

(b) **Gating type**

case	FID↓
uniform	5.97
distinct	**5.87**

(c) **Prompt selection**

case	FID↓
prepend	6.79
add	**5.87**

(d) **Prompt position**

depth	0	1	6	**12**
FID↓	6.27	6.13	5.99	**5.87**

(e) **Prompt depth**

importance	0	0	1	1	2	**1**
load	0	1	0	2	1	**1**
FID↓	6.11	5.96	5.97	5.95	5.95	**5.87**

(f) **Prompt balancing loss**

Table 4: **DMP ablations.** DiT-B/2 (Peebles and Xie 2022) pre-trained on FFHQ 256×256 (Karras, Laine, and Aila 2019) is further trained for 30K iterations with DMP (Baseline FID = 6.27). ↓: The lower the better.

Effects across training iterations. In Tab. 3, further training of a DiT-L/2 model reveals interesting dynamics. First, fine-tuning fails to increase performance beyond its already converged state and even tends to overfit, leading to performance degradation. We also found that prompt tuning, which uses a small number of extra parameters, actually harms performance, possibly because these extra parameters act as noise in the model's input space. In contrast, DMP, which also uses the set of parameters as prompt tuning, significantly boosts performance. The key difference between them lies in the use of a gating function: prompts are shared across all timesteps, while DMP activates prompts differently for each timestep. This distinction allows DMP, with a fixed number of parameters, to scale across thousands of timesteps by creating mixtures-of-prompts. By patching stage-specificity into the pre-trained diffusion model, DMP achieves a 10.38% FID gain in just 50K iterations.

Design Choices

Gating architecture. Our DMP framework incorporates a dynamic gating mechanism to select mixture-of-prompts.

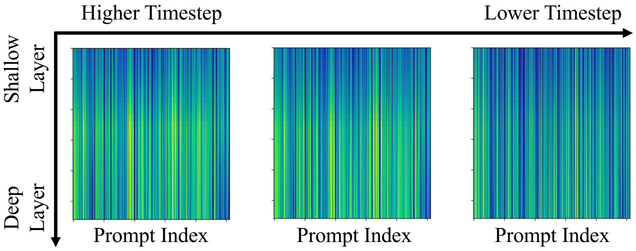


Figure 4: **Prompt activation.** Brighter indicates stronger.

We compare the impact of two gating architectures in Tab. 4a: linear gating *vs.* attention gating. The linear gating utilizes a single linear layer, taking a timestep embedding as input to produce a weighting mask for each learnable prompt. On the other hand, attention gating utilizes an attention layer (Vaswani et al. 2017), treating learnable prompts as a query and timestep embeddings as key and value, resulting in weighted prompts directly. Upon comparing the two gating architectures, we found that linear gating achieves better FID (5.87) compared to attention gating (6.41). As a result, we adopt linear gating as our default setting.

Gating type. In Tab. 4b, we evaluate two design choices for creating mixture-of-prompts: hard *vs.* soft selection. With hard selection, we choose the top-192 prompts out of 256 prompts (75% of total prompts) based on the gating probabilities, using them only with a weight of 1.

Whereas, soft selection uses all prompts but assigns different weights to each. Soft selection leads to further improvement, whereas hard selection results in worse performance. Therefore, we set the soft selection as our default setting.

Prompt selection. By default, DMP inserts learnable prompts into the input space of every blocks in the diffusion model. Two choices arise in this context: (1) uniform: gating function $\mathcal{G}$ in Eq. 3 receives only the timestep embedding $\boldsymbol{t}$ as input and applies common weights to the prompts at every block, thus prompt selection is consistent across all blocks. (2) distinct: $\mathcal{G}$ processes not only $\boldsymbol{t}$ but also current block depth i as inputs, generating different weights for each block. As shown in Tab. 4c, using distinct prompt selections leads to enhanced performance. Therefore, we input both the timestep embedding and current block depth information to the gating function, enabling the distinct prompt combinations for each block depth as our default setting.

Prompt position. While previous prompt tuning approaches (Li and Liang 2021; Lester, Al-Rfou, and Constant 2021; Jia et al. 2022; Zhou et al. 2022; Wang et al. 2022) typically prepend learnable prompts to image tokens, we directly add prompts element-wise to image tokens to maintain the input sequence length. Table 4d ablates different choices for inserting prompts into the input space and their impact on performance. We compare two methods: prepend *vs.* add. For "add", we use 256 prompts to match the number of image tokens, and for "prepend", we utilize 50 prompts. Although "prepend" should ideally use 256 tokens for a fair comparison, we limit it to 50 tokens due to severe divergence, even when both methods are equally zero-initialized. The results show that "add" method improves performance with a stable optimization process, achieving a 5.87 FID compared to the baseline of 6.27 FID. On the other hand, the "prepend" leads to a drop in performance, with an FID of 6.79. Additionally, "add" has the advantage of not increasing computation overhead. Based on these findings, we set "add" as our default for stable optimization.

Prompt depth. To investigate the impact of the number of blocks in which prompts are inserted, we conduct an ablation study using the DiT-B/2 model, which comprises 12 DiT blocks. We evaluate the performance differences when applying mixtures-of-prompts at various depths in Tab. **??**: only at the first block, up to half of the blocks, and across all blocks. Regardless of the depth, performance consistently improves compared to the baseline with no prompts (FID=6.27). Our findings indicate a positive correlation between prompt depth and performance, with better results achieved using mixture-of-prompts across more blocks. The prompts selected for each block are illustrated in Fig. 4.

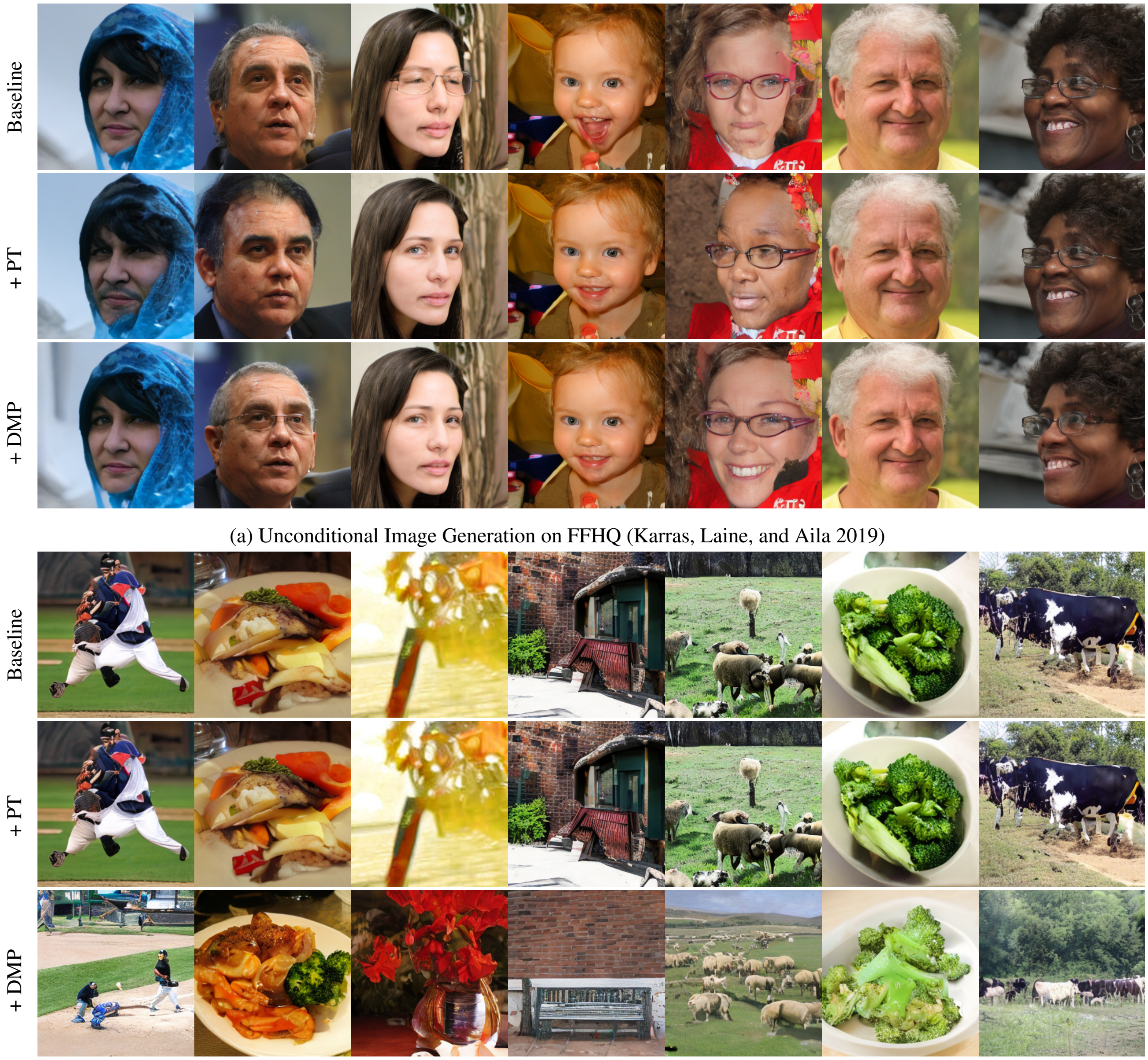
Prompt balancing. Prompt balancing loss acts as a soft constraint for the gating function, mitigating biased selections of prompts when producing a mixture-of-prompts. We study the impact of two types of balancing losses by altering the coefficient values for load balancing loss and importance balancing loss. As shown in Tab. 4f, using both types of losses equally enables the diffusion model to reach its peak performance. This indicates that balancing both the number and weight of the activated prompts across different timesteps is crucial for creating an effective mixture-of-prompts. Consequently, we employ equal proportions of importance balancing and load balancing losses for prompt balancing.

Analysis

Prompt activation. The gating function plays a pivotal role in dynamically crafting mixtures-of-prompts from a set of learnable prompts, based on the noise level present in the input. This is depicted in Fig. 4, where the activation is visually highlighted using colors. As the denoising process progresses, the selection of prompts exhibits significant variation across different timesteps. At higher timesteps with high noise levels, the gating function tends to utilize a broader array of prompts. Conversely, at lower timesteps, as the noise diminishes, the prompts become more specialized, focusing narrowly on specific features of the input that demand closer attention. This strategic deployment of prompts allows the model to form specialized "experts" at each denoising step, catering to the specific needs dictated by the input's noise characteristics and enhancing the model's performance.

Qualitative analysis. Figure 5 presents a visual comparison between three methods: the baseline DiT model, prompt tuning (Lester, Al-Rfou, and Constant 2021), and our DMP. These methods are evaluated on unconditional, text-to-image generation tasks using FFHQ (Karras, Laine, and Aila 2019) and COCO (Lin et al. 2014), respectively. DMP generates realistic and natural images with fewer artifacts.

Additional results and analysis. Appendix provides a theoretical grounding of DMP, experiments on training diffu-



(a) Unconditional Image Generation on FFHQ (Karras, Laine, and Aila 2019)

(b) Text-to-Image Generation on MS-COCO (Lin et al. 2014)

Figure 5: **Qualitative comparison** among the baseline (DiT-B/2 (Peebles and Xie 2022)), naive prompt tuning (PT), and DMP applied to the baseline on two datasets: (a) FFHQ and (b) MS-COCO.

sion models from scratch with DMP, applying DMP on DiT-XL/2, comparisons with LoRA (Hu et al. 2021), and ablations of gating conditions. It also examines the structural bias of DMP and provides additional qualitative results.

Conclusion

We introduced Diffusion Model Patching (DMP), a simple method for further enhancing pre-trained diffusion models that have already converged. By incorporating learnable prompts and leveraging dynamic gating, DMP adapts the model's behavior dynamically across thousands of denoising steps. This design enables DMP to effectively address the timestep-specific variations in each denoising stage, which are often overlooked in existing diffusion models. Our results demonstrate that DMP achieves significant performance gains without the need for extensive retraining. For the DiT-L/2 backbone, DMP improves a 10.38% of FID after just 50K iterations, with a minor parameter increase of 1.43% on the FFHQ 256×256 dataset. Additionally, its adaptability across different models underscores its potential as a versatile enhancement method for diffusion models.

References

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

Caruana, R. 1997. Multitask learning. *Machine learning*, 28: 41–75.

Chen, S.; Ge, C.; Tong, Z.; Wang, J.; Song, Y.; Wang, J.; and Luo, P. 2022. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35: 16664–16678.

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

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

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

Eigen, D.; Ranzato, M.; and Sutskever, I. 2013. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*.

Feng, Z.; Zhang, Z.; Yu, X.; Fang, Y.; Li, L.; Chen, X.; Lu, Y.; Liu, J.; Yin, W.; Feng, S.; et al. 2023. ERNIE-ViLG 2.0: Improving text-to-image diffusion model with knowledge-enhanced mixture-of-denoising-experts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10135–10145.

Gal, R.; Alaluf, Y.; Atzmon, Y.; Patashnik, O.; Bermano, A. H.; Chechik, G.; and Cohen-Or, D. 2022. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*.

Go, H.; Kim, J.; Lee, Y.; Lee, S.; Oh, S.; Moon, H.; and Choi, S. 2023. Addressing Negative Transfer in Diffusion Models. *arXiv preprint arXiv:2306.00354*.

Golatkar, A.; Achille, A.; Swaminathan, A.; and Soatto, S. 2023. Training data protection with compositional diffusion models. *arXiv preprint arXiv:2308.01937*.

Han, L.; Li, Y.; Zhang, H.; Milanfar, P.; Metaxas, D.; and Yang, F. 2023. Svdiff: Compact parameter space for diffusion fine-tuning. *arXiv preprint arXiv:2303.11305*.

Hang, T.; Gu, S.; Li, C.; Bao, J.; Chen, D.; Hu, H.; Geng, X.; and Guo, B. 2023. Efficient diffusion training via min-snr weighting strategy. *arXiv preprint arXiv:2303.09556*.

Hertz, A.; Mokady, R.; Tenenbaum, J.; Aberman, K.; Pritch, Y.; and Cohen-Or, D. 2022. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*.

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

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

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, 2790–2799. PMLR.

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

Jacobs, R. A.; Jordan, M. I.; Nowlan, S. J.; and Hinton, G. E. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1): 79–87.

Jia, M.; Tang, L.; Chen, B.-C.; Cardie, C.; Belongie, S.; Hariharan, B.; and Lim, S.-N. 2022. Visual prompt tuning. In *European Conference on Computer Vision*, 709–727. Springer.

Jiang, Z.; Mao, C.; Pan, Y.; Han, Z.; and Zhang, J. 2023. SCEdit: Efficient and Controllable Image Diffusion Generation via Skip Connection Editing. *arXiv preprint arXiv:2312.11392*.

Jordan, M. I.; and Jacobs, R. A. 1994. Hierarchical mixtures of experts and the EM algorithm. *Neural computation*, 6(2): 181–214.

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

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kong, Z.; Ping, W.; Huang, J.; Zhao, K.; and Catanzaro, B. 2020. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*.

Kumari, N.; Zhang, B.; Zhang, R.; Shechtman, E.; and Zhu, J.-Y. 2023. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1931–1941.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Li, X.; Thickstun, J.; Gulrajani, I.; Liang, P. S.; and Hashimoto, T. B. 2022. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35: 4328–4343.

Li, X. L.; and Liang, P. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Lian, D.; Zhou, D.; Feng, J.; and Wang, X. 2022. Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems*, 35: 109–123.

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

Mou, C.; Wang, X.; Xie, L.; Zhang, J.; Qi, Z.; Shan, Y.; and Qie, X. 2023. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*.

Nichol, A. Q.; and Dhariwal, P. 2021. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, 8162–8171. PMLR.

Park, B.; Go, H.; Kim, J.-Y.; Woo, S.; Ham, S.; and Kim, C. 2024. Switch Diffusion Transformer: Synergizing Denoising Tasks with Sparse Mixture-of-Experts. *arXiv preprint arXiv:2403.09176*.

Park, B.; Woo, S.; Go, H.; Kim, J.-Y.; and Kim, C. 2023. Denoising Task Routing for Diffusion Models. *arXiv preprint arXiv:2310.07138*.

Peebles, W.; and Xie, S. 2022. Scalable Diffusion Models with Transformers. *arXiv preprint arXiv:2212.09748*.

Poole, B.; Jain, A.; Barron, J. T.; and Mildenhall, B. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*.

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 *International Conference on Machine Learning*, 8821–8831. PMLR.

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

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. 2022a. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35: 36479–36494.

Saharia, C.; Ho, J.; Chan, W.; Salimans, T.; Fleet, D. J.; and Norouzi, M. 2022b. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4): 4713–4726.

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

Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; and Dean, J. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, 2256–2265. PMLR.

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

Sung, Y.-L.; Cho, J.; and Bansal, M. 2022. Lst: Ladder side-tuning for parameter and memory efficient transfer learning. *Advances in Neural Information Processing Systems*, 35: 12991–13005.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

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

Wang, Z.; Zhang, Z.; Lee, C.-Y.; Zhang, H.; Sun, R.; Ren, X.; Su, G.; Perot, V.; Dy, J.; and Pfister, T. 2022. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 139–149.

Woo, S.; Park, B.; Go, H.; Kim, J.-Y.; and Kim, C. 2023. HarmonyView: Harmonizing Consistency and Diversity in One-Image-to-3D. *arXiv preprint arXiv:2312.15980*.

Xiang, C.; Bao, F.; Li, C.; Su, H.; and Zhu, J. 2023. A closer look at parameter-efficient tuning in diffusion models. *arXiv preprint arXiv:2303.18181*.

Xie, E.; Yao, L.; Shi, H.; Liu, Z.; Zhou, D.; Liu, Z.; Li, J.; and Li, Z. 2023. DiffFit: Unlocking Transferability of Large Diffusion Models via Simple Parameter-Efficient Fine-Tuning. *arXiv preprint arXiv:2304.06648*.

Xu, R.; Luo, F.; Zhang, Z.; Tan, C.; Chang, B.; Huang, S.; and Huang, F. 2021. Raise a child in large language model: Towards effective and generalizable fine-tuning. *arXiv preprint arXiv:2109.05687*.

Xue, Z.; Song, G.; Guo, Q.; Liu, B.; Zong, Z.; Liu, Y.; and Luo, P. 2023. Raphael: Text-to-image generation via large mixture of diffusion paths. *arXiv preprint arXiv:2305.18295*.

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

Zhang, J. O.; Sax, A.; Zamir, A.; Guibas, L.; and Malik, J. 2020. Side-tuning: a baseline for network adaptation via additive side networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, 698–714. Springer.

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

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9): 2337–2348.