Supplementary Material
DiffFit: Unlocking Transferability of Large Diffusion Models via Simple Parameter-Efficient Fine-Tuning

Enze Xie[1], Lewei Yao[1], Han Shi[1], Zhili Liu[1], Daquan Zhou[2], Zhaoqiang Liu[1], Jiawei Li[1], Zhenguo Li[1]
[1]Huawei Noah's Ark Lab [2]National University of Singapore

1. More Applications

1.1. Combine DiffFit with ControlNet

Setup. To verify that DiffFit can be applied to large-scale text-to-image generation models, we conducted experiments on the recent popular ControlNet [29] model. We used the official code of ControlNet[1], and chose the semantic segmentation mask as the additional condition. We used the COCO-Stuff [2] as our dataset and the text prompt was generated by the BLIP [15]. The model was trained on the training set and evaluated on the validation set using 8 V100 GPUs with a batch size of 2 per GPU and 20 epochs. Unless otherwise stated, we followed the official implementation of ControlNet for all hyper-parameters including the learning rate and the resolution.

Combination. ControlNet [29] enhances the original Stable Diffusion (SD) by incorporating a conditioning network comprising two parts: a set of zero convolutional layers initialized to zero and a trainable copy of 13 layers of SD initialized from a pre-trained model.

In our DiffFit setting, we freeze the entire trainable copy part and introduce a scale factor of γ in each layer. Then, we unfreeze the γ and all the bias terms. Note that in ControlNet, the zero convolutional layers are required to be trainable. Table 1 shows that DiffFit has 11.2M trainable parameters while the zero convolutional layers contribute 10.8M parameters. Reducing the trainable parameters in zero convolutional layers is plausible, but it is beyond the scope of this paper.

Results. Table 1 displays the results obtained from the original ControlNet and the results from combining DiffFit with ControlNet. Our results show that the addition of DiffFit leads to comparable FID and CLIP scores while significantly reducing the number of trainable parameters. In addition, we note that ControlNet primarily focuses on fine-grained controllable generation while FID and CLIP score may not accurately reflect the overall performance of a model. Figures 1 and 2 compare the results from ControlNet and DiffFit. When using the same mask and text prompt as conditions, fine-tuning with DiffFit produces more visually appealing results compared to fine-tuning with the original ControlNet.

Our experimental results suggest that DiffFit has great potential to improve the training of other types of advanced generative models.

1.2. Combine DiffFit with DreamBooth

Setup. Our DiffFit can also be easily combined with the DreamBooth [22]. DreamBooth is a personalized text-to-image diffusion model that fine-tunes a pre-trained text-to-image model using a few reference images of a specific subject, allowing it to synthesize fully-novel photorealistic images of the subject contextualized in different scenes. Our implementation on DreamBooth utilizes the codebase from Diffusers[2], with Stable Diffusion as the base text-to-image model.

Implementation detail. Given 3-5 input images, the original DreamBooth fine-tunes the subject embedding and the entire text-to-image (T2I) model. In contrast, our DiffFit simplifies the fine-tuning process by incorporating scale factor γ into all attention layers of the model and requiring only the γ, bias term, LN, and subject embedding to be fine-tuned.

Results. We compare the original full-finetuning approach of DreamBooth with the fine-tuning using LoRA and DiffFit, as shown in Figures 3 and 4. First, we observe that all three methods produce similar generation quality. Second, we find that the original full-finetuning approach is storage-intensive, requiring the storage of 859M parameters for one subject-driven

[1] https://github.com/lllyasviel/ControlNet
[2] https://github.com/huggingface/diffusers/tree/main/examples/dreambooth

fine-tuning. In contrast, LoRA and DiffFit are both parameter-efficient (requiring the storage of less than 1% parameters), with DiffFit further reducing trainable parameters by about 26% compared to LoRA.

1.3. Combine DiffFit with LLaMA and Alpaca

Setup. LLaMA [24] is an open and efficient large language model (LLM) from Meta, ranging from 7B to 65B parameters. Alpaca is a model fully fine-tuned from the LLaMA 7B model on 52K instruction-following demonstrations and behaves qualitatively similarly to OpenAI's ChatGPT model `text-davinci-003`.

Combination. We use the original code from Alpaca[3] as the baseline and compare it with Alpaca-LoRA[4]. Since LLaMA and DiT have a similar Transformer design, the fine-tuning strategy of LLaMA-DiffFit is exactly the same as that of DiT-DiffFit, except that the learning rate is set to 3e-4 to align with Alpaca-LoRA.

Results. In Table 2, we showcase the instruction tuning results on llama using Alpaca, LoRA, and DiffFit. The trainable parameters of Alpaca, LoRA, and DiffFit are ***7B, 6.7M, and 0.7M***. The model fine-tuned with DiffFit exhibits strong performance in natural language question-answering tasks (e.g., common sense questioning, programming, etc.). This validates the efficacy of DiffFit for instruction tuning in NLP models, highlighting DiffFit as a flexible and general fine-tuning approach.

Method	FID ↓	CLIP Score ↑	Total Params (M)	Trainable Params (M)
ControlNet	20.1	0.3067	1343	361
ControlNet + DiffFit	19.5	0.3064	1343	11.2

Table 1: Results of original ControlNet and combined DiffFit on COCO-Stuff dataset.

2. More Implementation Details

2.1. Adding the Scale Factor γ in Self-Attention

This section demonstrates the incorporation of scale factor γ into the self-attention modules. Traditionally, a self-attention layer comprises two linear operations: (1) QKV-Linear operation and (2) Project-Linear operation. Consistent with the approach discussed above in the paper, the learnable scale factor is initialized to 1.0 and subsequently fine-tuned. Algorithm 1 presents the Pytorch-style pseudo code.

2.2. Dataset Description

This section describes 8 downstream datasets/tasks that we have utilized in our experiments to fine-tune the DiT with our DiffFit method.

Food101 [1]. This dataset contains 101 food categories, totaling 101,000 images. Each category includes 750 training images and 250 manually reviewed test images. The training images were kept intentionally uncleaned, preserving some degree of noise, primarily vivid colors and occasionally incorrect labels. All images have been adjusted to a maximum side length of 512 pixels.

SUN 397 [28]. The SUN benchmark database comprises 108,753 images labeled into 397 distinct categories. The quantities of images vary among the categories, however, each category is represented by a minimum of 100 images. These images are commonly used in scene understanding applications.

DF20M [21]. DF20 is a new fine-grained dataset and benchmark featuring highly accurate class labels based on the taxonomy of observations submitted to the Danish Fungal Atlas. The dataset has a well-defined class hierarchy and a rich observational metadata. It is characterized by a highly imbalanced long-tailed class distribution and a negligible error rate. Importantly, DF20 has no intersection with ImageNet, ensuring unbiased comparison of models fine-tuned from ImageNet checkpoints.

Caltech 101 [11]. The Caltech 101 dataset comprises photos of objects within 101 distinct categories, with roughly 40 to 800 images allocated to each category. The majority of the categories have around 50 images. Each image is approximately 300×200 pixels in size.

[3] https://github.com/tatsu-lab/stanford_alpaca
[4] https://github.com/tloen/alpaca-lora

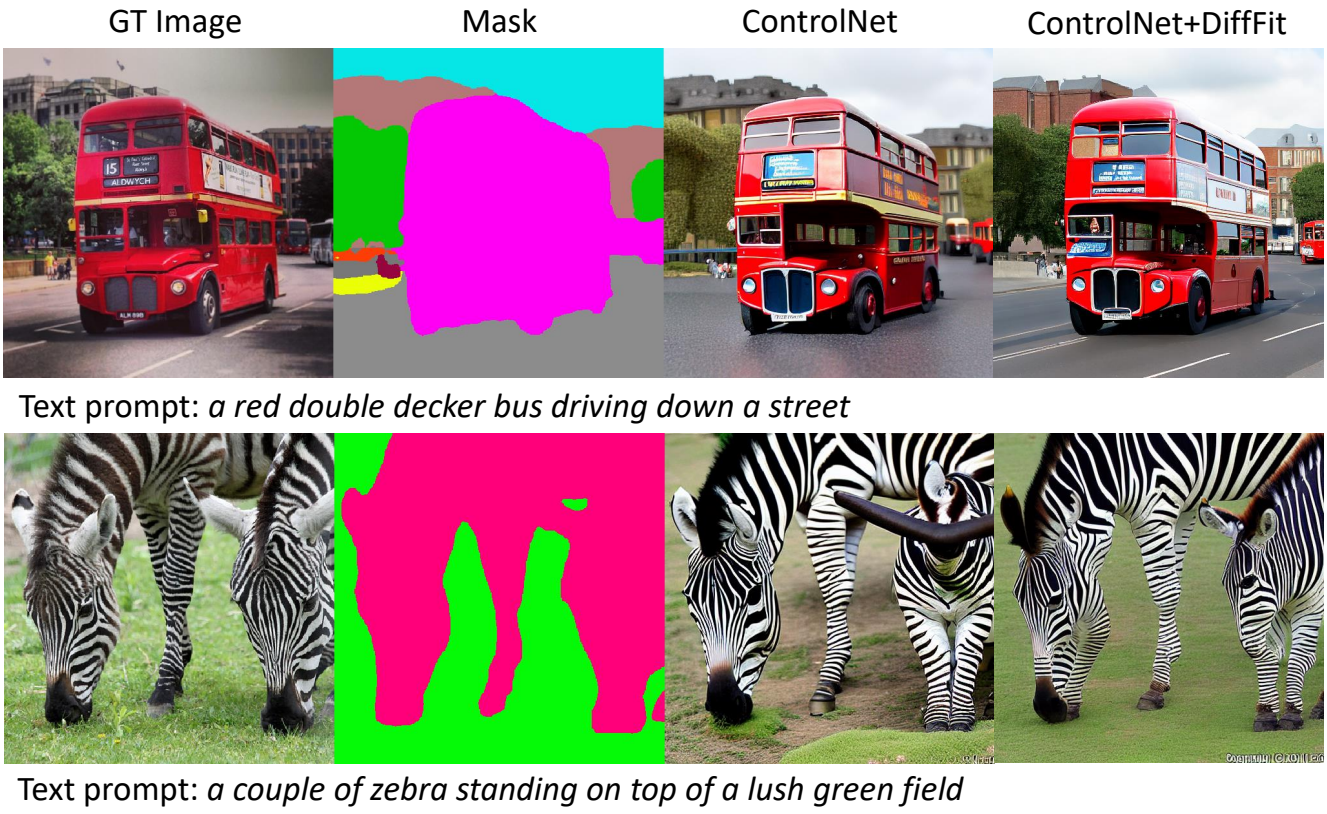

Figure 1: Visualization of original ControlNet and combined DiffFit on COCO-Stuff dataset.

CUB-200-2011 [27]. CUB-200-2011 (Caltech-UCSD Birds-200-2011) is an expansion of the CUB-200 dataset by approximately doubling the number of images per category and adding new annotations for part locations. The dataset consists of 11,788 images divided into 200 categories.

ArtBench-10 [16]. ArtBench-10 is a class-balanced, standardized dataset comprising 60,000 high-quality images of artwork annotated with clean and precise labels. It offers several advantages over previous artwork datasets including balanced class distribution, high-quality images, and standardized data collection and pre-processing procedures. It contains 5,000 training images and 1,000 testing images per style.

Oxford Flowers [20]. The Oxford 102 Flowers Dataset contains high quality images of 102 commonly occurring flower categories in the United Kingdom. The number of images per category range between 40 and 258. This extensive dataset provides an excellent resource for various computer vision applications, especially those focused on flower recognition and classification.

Stanford Cars [13]. In the Stanford Cars dataset, there are 16,185 images that display 196 distinct classes of cars. These images are divided into a training and a testing set: 8,144 images for training and 8,041 images for testing. The distribution of samples among classes is almost balanced. Each class represents a specific make, model, and year combination, e.g., the 2012 Tesla Model S or the 2012 BMW M3 coupe.

3. More FID Curves

We present additional FID curves for the following datasets: Stanford Cars, SUN 397, Caltech 101, and DF20M as illustrated in Figures 5 through 8, respectively. The results demonstrate that our DiffFit achieves rapid convergence, delivers compelling FID scores compared to other parameter-efficient finetuning strategies, and is competitive in the full-finetuning setting.

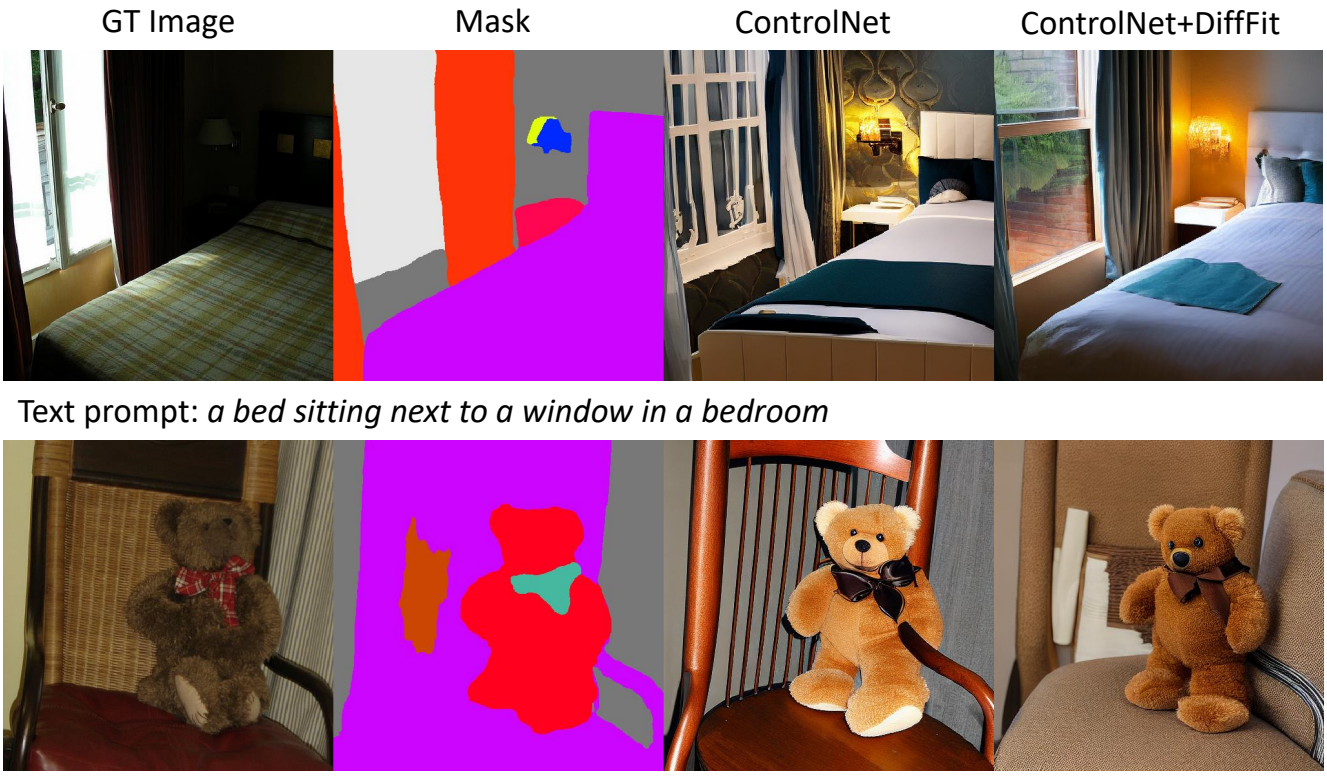

Figure 2: Visualization of original ControlNet and combined DiffFit on COCO-Stuff dataset.

4. More Theoretical Analysis

In this section, we will provide a more detailed analysis of the effect of scaling factors. Specifically, we will present a formal version and proof of Theorem 1, which was informally stated in the main document.

Our theoretical results are closely related to the learning of neural networks. It is worth noting that the majority of works in this area concentrate on the simplified setting where the neural network only has one non-linearly activated hidden layer with no bias terms, see, e.g., [31, 10, 8, 30, 9, 25]. As our contributions are mainly experimental, we only provide some intuitive theoretical analysis to reveal the effect of scaling factors. To this end, we will consider the following simplified settings.

- Following the approach of [12, 5], we replace the noise neural network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$ used in the diffusion model with $\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}}$,[5] where $\mathbf{x}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$ is a neural network that approximates the original signal. In addition, in the sampling process, we assume a single sampling step from the end time $t = E$ to the initial time $t = 0$, which gives $\mathbf{x}_0 = \mathbf{x}_{\boldsymbol{\theta}}(\mathbf{x}_E, E)$. For brevity, we denote $\mathbf{x}_{\boldsymbol{\theta}}(\cdot, E)$ by $G(\cdot)$, and we assume that $G(\mathbf{s}) = f(\mathbf{W}\mathbf{s} + \mathbf{b})$ for all $\mathbf{s} \in \mathbb{R}^D$, where $\mathbf{W} \in \mathbb{R}^{D\times D}$ is a weight matrix, $\mathbf{b} \in \mathbb{R}^D$ is the offset vector, and $f : \mathbb{R} \to \mathbb{R}$ represents a non-linear activation function that does not grow super-linearly (*cf.* Appendix 4.3) and it is applied element-wise. While one-step generation may seem restrictive, recent works [17, 23] have achieved it for the diffusion models.

- Suppose that we have a dataset generated from distribution $Q_0 \in \mathbb{R}^D$. Further assuming that there exist ground-truth scaling factors $\boldsymbol{\gamma}^* \in \mathbb{R}^D$ such that each entry of any data point in a relatively small dataset generated from distribution $P_0 \in \mathbb{R}^D$ can be written as $\mathbf{x}^T\boldsymbol{\gamma}^*$ for some $\mathbf{x}$ sampled from Q_0. We denote this transition from Q_0 to P_0 as $P_0 = \boldsymbol{f}_{\boldsymbol{\gamma}^*}\#Q_0$ for brevity.

[5] Here we assume that in the forward process, the conditional distribution $q_{0t}(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$. See Section 3.1 in the main document.

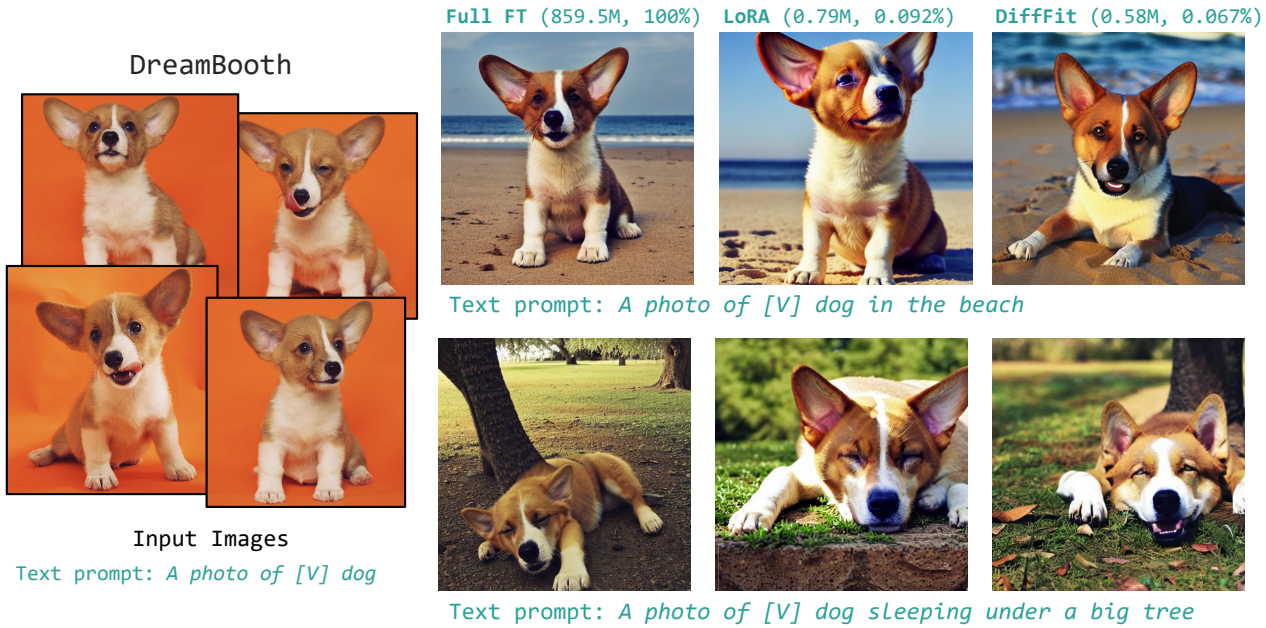

Figure 3: Visualization of original DreamBooth, with LoRA and DiffFit.

Figure 4: Visualization of original DreamBooth, with LoRA and DiffFit.

- Recent theoretical results for diffusion models [7, 6, 14, 4, 3] show that under appropriate conditions, diffusion models can generate samples that approximately follow the original data distribution. Based on these results, we assume that $G(\boldsymbol{\epsilon}) \sim \hat{Q}_0$ for $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\hat{Q}_0$ is close to Q_0 in some probability distance.

Under the above settings, in the training process for the diffusion model using the relatively small dataset, if we only

Instruction: Tell me about alpacas.
Alpaca: Alpacas are small, fluffy animals related to camels and llamas. They are native to Peru and Bolivia, and were first domesticated around 5,000 years ago. They are kept mainly for their fine, soft fleece, which is used to make knitwear and other garments. Alpacas are herd animals and live in small family groups, led by an older male. They are highly social animals and can form strong bonds with their owners.
Alpaca-LoRA: Alpacas are members of the camelid family and are native to the Andes Mountains of South America. They are known for their soft, luxurious fleece, which is used to make clothing, blankets, and other items. Alpacas are herbivores and graze on grasses and other plants. They are social animals and live in herds of up to 20 individuals.
LLaMA-DiffFit: Alpacas are members of the camelid family and are native to the Andes Mountains in South America. They are domesticated for their soft, luxurious fleece, which is used to make clothing, blankets, and other textiles. Alpacas are herbivores and graze on grasses and other plants. They can live up to 20 years in captivity and are known for their gentle nature.

Instruction: List all Canadian provinces in alphabetical order.
Alpaca: Alberta, British Columbia, Manitoba, New Brunswick, Newfoundland and Labrador, Nova Scotia, Ontario, Prince Edward Island, Quebec, Saskatchewan, Northwest Territories, Nunavut, Yukon.
Alpaca-LoRA: Alberta, British Columbia, Manitoba, New Brunswick, Newfoundland and Labrador, Nova Scotia, Ontario, Prince Edward Island, Quebec, Saskatchewan.
LLaMA-DiffFit: Alberta, British Columbia, Manitoba, New Brunswick, Newfoundland and Labrador, Northwest Territories, Nova Scotia, Nunavut, Ontario, Prince Edward Island, Quebec, Saskatchewan, Yukon.

Instruction: Tell me five words that rhyme with 'shock'.
Alpaca: Five words that rhyme with shock are: rock, pop, shock, cook, and snock.
Alpaca-LoRA: Flock, lock, rock, stock, and sock.
LLaMA-DiffFit: Shock, sock, rock, cook, and book.

Instruction: Write a Python program that prints the first 10 Fibonacci numbers.
Alpaca:

```
numbers = [0, 1]

for i in range(2, 11):
   numbers.append(numbers[i-2] + numbers[i-1])

print(numbers)
# Output: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

Alpaca-LoRA:

```
def fibonacci(n):
   if n == 0:
      return 0
   elif n == 1:
      return 1
   else:
      return fibonacci(n-1) + fibonacci(n-2)

print(fibonacci(10))
```

LLaMA-DiffFit:

```
def fib(n):
   if n == 0:
      return 0
   elif n == 1:
      return 1
   else:
      return fib(n-1) + fib(n-2)

print(fib(10))
```

Table 2: **Instruction-following Comparison** between Alpaca, Alpaca-LoRA and LLaMA-DiffFit.

fine-tune the scaling factors, the objective function can be expressed as:

$$\min_{\boldsymbol{\gamma}\in\mathbb{R}^D} \sum_{i=1}^{m} \|G(\boldsymbol{\epsilon}_i)^T\boldsymbol{\gamma} - \mathbf{x}_i^T\boldsymbol{\gamma}^*\|_2^2, \tag{1}$$

Algorithm 1 Adding the trainable scale factor γ in the attention block.

```
import torch
import torch.nn as nn

class Attention(nn.Module):
 def __init__(self, dim, num_heads=8, qkv_bias=False, proj_bias=True, attn_drop=0., proj_drop=0., eta=None):
   super().__init__()
   self.num_heads = num_heads
   head_dim = dim // num_heads
   self.scale = head_dim ** -0.5

   self.qkv = nn.Linear(dim, dim * 3, bias=qkv_bias)
   self.attn_drop = nn.Dropout(attn_drop)
   self.proj = nn.Linear(dim, dim, bias=proj_bias)
   self.proj_drop = nn.Dropout(proj_drop)

   # Initilize gamma to 1.0
   self.gamma1 = nn.Parameter(torch.ones(dim * 3))
   self.gamma2 = nn.Parameter(torch.ones(dim))


 def forward(self, x):
   B, N, C = x.shape
   # Apply gamma
   qkv = (self.gamma1 * self.qkv(x)).reshape(B, N, 3, self.num_heads, C//self.num_heads).permute(2,0,3,1,4)
   q, k, v = qkv.unbind(0)

   attn = (q @ k.transpose(-2, -1)) * self.scale
   attn = attn.softmax(dim=-1)
   attn = self.attn_drop(attn)

   x = (attn @ v).transpose(1, 2).reshape(B, N, C)
   # Apply gamma
   x = self.gamma2 * self.proj(x)
   x = self.proj_drop(x)
   return x
```

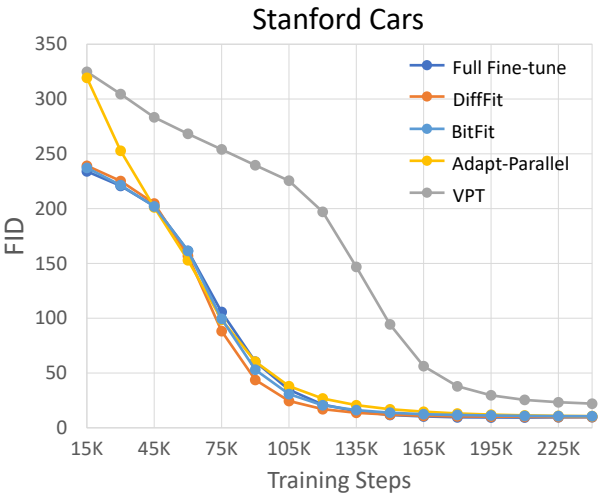

Figure 5: FID of five methods every 15K iterations on Stanford Cars dataset.

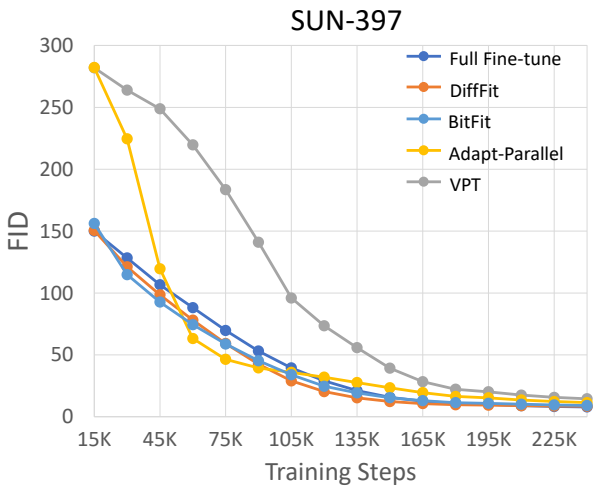

Figure 6: FID of five methods every 15K iterations on SUN 397 dataset.

where $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{x}_i \sim Q_0$ (then $\mathbf{x}_i^T \boldsymbol{\gamma}^*$ corresponds to an entry of data point generated from P_0), and m is the number of training samples. As $G(\boldsymbol{\epsilon}_i)$ is sub-Gaussian (*cf.* Appendix 4.2) and $G(\boldsymbol{\epsilon}_i) \sim \hat{Q}_0$ with $\hat{Q}_0$ being close to Q_0, we assume that $\mathbf{x}_i$ can be represented as $\mathbf{x}_i = G(\boldsymbol{\epsilon}_i) + \mathbf{z}_i$, where each entry of $\mathbf{z}_i$ is zero-mean sub-Gaussian with the sub-Gaussian norm (*cf.* Appendix 4.2 for the definition of sub-Gaussian norm) being upper bounded by some small constant $\eta > 0$.

Let $\mathbf{a}_i = G(\boldsymbol{\epsilon}_i) = f(\mathbf{W}\boldsymbol{\epsilon}_i + \mathbf{b}) \in \mathbb{R}^D$. We write $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_m]^T \in \mathbb{R}^{m \times D}$, $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_m]^T \in \mathbb{R}^{m \times D}$, and $\mathbf{y} = [y_1, \ldots, y_m]^T$ with $y_i = \mathbf{x}_i^T \boldsymbol{\gamma}^*$. Then, we have $\mathbf{y} = (\mathbf{A} + \mathbf{Z})\boldsymbol{\gamma}^*$ and the objective function (1) can be re-written as

$$\min_{\boldsymbol{\gamma} \in \mathbb{R}^D} \|\mathbf{A}\boldsymbol{\gamma} - \mathbf{y}\|_2^2. \tag{2}$$

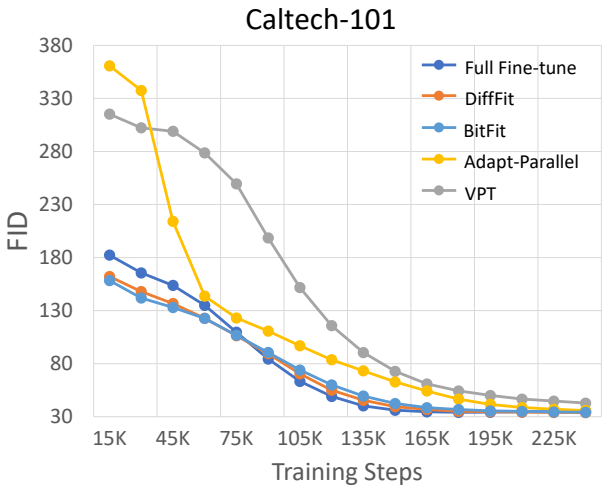

Figure 7: FID of five methods every 15K iterations on Caltech 101 dataset.

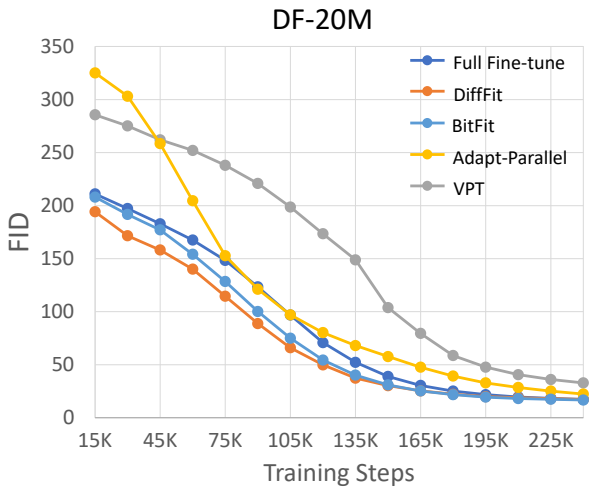

Figure 8: FID of five methods every 15K iterations on DF20M dataset.

Let $V_{\min} := \min_{i\in\{1,\ldots,D\}} \mathrm{Var}[f(X_i)]$ with $X_i \sim \mathcal{N}(b_i, \|\mathbf{w}_i\|_2^2)$, where b_i is the i-th entry of the offset vector $\mathbf{b}$ and $\mathbf{w}_i^T \in \mathbb{R}^{1\times D}$ is the i-th row of the weight matrix $\mathbf{W}$. Then, we have the following theorem concerning the optimal solution to (2). The proof of Theorem 1 is deferred to Section 4.4.

Theorem 1. *Under the above settings, if $\hat{\boldsymbol{\gamma}}$ is an optimal solution to (2) and $m = \Omega(D^2 \log D)$, with probability $1 - e^{-\Omega(D\log D)}$, it holds that*

$$\|\hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}^*\|_2 < \frac{4\sqrt{2}\eta}{\sqrt{V_{\min}}} \cdot \|\boldsymbol{\gamma}^*\|_2. \tag{3}$$

Note that $\eta > 0$ is considered to be small and $V_{\min}$ is a fixed positive constant. In addition, the gradient descent algorithm aims to minimize (2). Therefore, *Theorem 1 essentially says that when the number of training samples is sufficiently large, the simple gradient descent algorithm finds an estimate $\hat{\boldsymbol{\gamma}}$ that is close to $\boldsymbol{\gamma}^*$ with high probability* (in the sense that the relative distance $\|\hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}^*\|_2/\|\boldsymbol{\gamma}^*\|_2$ is small). Furthermore, *with this $\hat{\boldsymbol{\gamma}}$, the diffusion model generates distribution $\hat{P}_0 = \boldsymbol{f}_{\hat{\boldsymbol{\gamma}}}\#\hat{Q}_0$, which is naturally considered to be close to $P_0 = \boldsymbol{f}_{\boldsymbol{\gamma}^*}\#Q_0$ since $\hat{\boldsymbol{\gamma}}$ is close to $\boldsymbol{\gamma}^*$ and $\hat{Q}_0$ is close to Q_0.*

Before presenting the proof of Theorem 1 in Section 4.4, we provide some auxiliary results.

4.1. Notation

We write $[N] = \{1, 2, \ldots, N\}$ for a positive integer N. The unit sphere in $\mathbb{R}^D$ is denoted by $\mathcal{S}^{D-1} := \{\mathbf{x} \in \mathbb{R}^D : \|\mathbf{x}\|_2 = 1\}$. We use $\|\mathbf{X}\|_2$ to denote the spectral norm of a matrix $\mathbf{X}$. We use the symbols C, C', c to denote absolute constants, whose values may differ from line to line.

4.2. General Auxiliary Results

First, the widely-used notion of an ϵ-net is presented as follows, see, e.g., [18, Definition 3].

Definition 1. *Let $(\mathcal{X}, \mathrm{d})$ be a metric space, and fix $\epsilon > 0$. A subset $S \subseteq \mathcal{X}$ is said be an ϵ-net of $\mathcal{X}$ if, for all $\mathbf{x} \in \mathcal{X}$, there exists some $\mathbf{s} \in S$ such that $\mathrm{d}(\mathbf{s}, \mathbf{x}) \le \epsilon$. The minimal cardinality of an ϵ-net of $\mathcal{X}$, if finite, is denoted $C(\mathcal{X}, \epsilon)$ and is called the covering number of $\mathcal{X}$ (at scale ϵ).*

The following lemma provides a useful bound for the covering number of the unit sphere.

Lemma 1. [26, Lemma 5.2] *The unit Euclidean sphere $\mathcal{S}^{D-1}$ equipped with the Euclidean metric satisfies for every $\epsilon > 0$ that*

$$\mathcal{C}(\mathcal{S}^{D-1}, \epsilon) \le \left(1 + \frac{2}{\epsilon}\right)^D. \tag{4}$$

In addition, we have the following lemma concerning the spectral norm of a matrix.

Lemma 2. [26, Lemma 5.3] *Let* $\mathbf{A}$ *be an* $m \times D$ *matrix, and let* $\mathcal{N}_\epsilon$ *be an* ϵ*-net of* $\mathcal{S}^{D-1}$ *for some* $\epsilon \in (0,1)$*. Then*

$$\|\mathbf{A}\|_2 \leq (1-\epsilon)^{-1} \max_{\mathbf{x}\in\mathcal{N}_\epsilon} \|\mathbf{A}\mathbf{x}\|_2. \tag{5}$$

Standard definitions for sub-Gaussian and sub-exponential random variables are presented as follows.

Definition 2. *A random variable* X *is said to be sub-Gaussian if there exists a positive constant* C *such that* $\left(\mathbb{E}\left[|X|^p\right]\right)^{1/p} \leq C\sqrt{p}$ *for all* $p \geq 1$*, and the corresponding sub-Gaussian norm is defined as* $\|X\|_{\psi_2} := \sup_{p\geq 1} p^{-1/2} \left(\mathbb{E}\left[|X|^p\right]\right)^{1/p}$.

Definition 3. *A random variable* X *is said to be sub-exponential if there exists a positive constant* C *such that* $\left(\mathbb{E}\left[|X|^p\right]\right)^{1/p} \leq Cp$ *for all* $p \geq 1$*, and the corresponding sub-exponential norm is defined as* $\|X\|_{\psi_1} := \sup_{p\geq 1} p^{-1} \left(\mathbb{E}\left[|X|^p\right]\right)^{1/p}$.

The following lemma concerns the relation between sub-Gaussian and sub-exponential random variables.

Lemma 3. [26, Lemma 5.14] *A random variable* X *is sub-Gaussian if and only if* X^2 *is sub-exponential. Moreover,*

$$\|X\|_{\psi_2}^2 \leq \|X\|_{\psi_1}^2 \leq 2\|X\|_{\psi_2}^2. \tag{6}$$

The following lemma provides a useful concentration inequality for the sum of independent sub-exponential random variables.

Lemma 4. [26, Proposition 5.16] *Let* $X_1, \ldots, X_N$ *be independent zero-mean sub-exponential random variables, and* $K = \max_i \|X_i\|_{\psi_1}$*. Then for every* $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^T \in \mathbb{R}^N$ *and* $\epsilon \geq 0$*, it holds that*

$$\mathbb{P}\left(\left|\sum_{i=1}^N \alpha_i X_i\right| \geq \epsilon\right) \leq 2\exp\left(-c\cdot\min\left(\frac{\epsilon^2}{K^2\|\boldsymbol{\alpha}\|_2^2}, \frac{\epsilon}{K\|\boldsymbol{\alpha}\|_\infty}\right)\right), \tag{7}$$

where $c > 0$ *is an absolute constant. In particular, with* $\boldsymbol{\alpha} = \left[\frac{1}{N}, \ldots, \frac{1}{N}\right]^T$*, we have*

$$\mathbb{P}\left(\left|\frac{1}{N}\sum_{i=1}^N X_i\right| \geq \epsilon\right) \leq 2\exp\left(-c\cdot\min\left(\frac{N\epsilon^2}{K^2}, \frac{N\epsilon}{K}\right)\right). \tag{8}$$

4.3. Useful Lemmas

Throughout the following, we use $f(\cdot)$ to denote some non-linear activation function that does not grow faster than linear, i.e., there exist scalars a and b such that $f(x) \leq a|x| + b$ for all $x \in \mathbb{R}$. Then, if $X \sim \mathcal{N}(\mu, \sigma^2)$ is a random Gaussian variable, $f(X)$ will be sub-Gaussian [19]. Note that the condition that $f(\cdot)$ does not grow super-linearly is satisfied by popular activation functions such as ReLU, Sigmoid, and Hyperbolic tangent function.

Lemma 5. *For* $i \in [N]$*, let* $X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ *be a random Gaussian variable. Then for every* $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_N]^T \in \mathbb{R}^N$*, we have*

$$V_{\min}\|\boldsymbol{\gamma}\|_2^2 \leq \mathbb{E}\left[\left(\sum_{i=1}^N \gamma_i f(X_i)\right)^2\right] \leq (N+1)U_{\max}\|\boldsymbol{\gamma}\|_2^2, \tag{9}$$

where $V_{\min} = \min_{i\in[N]} \mathrm{Var}\left[f(X_i)\right]$ *and* $U_{\max} = \max_{i\in[N]} \mathbb{E}\left[f(X_i)^2\right]$ *are dependent on* $\{\mu_i, \sigma_i^2\}_{i\in[N]}$ *and the non-linear activation function* $f(\cdot)$*.*

Proof. We have

$$\begin{aligned}
\mathbb{E}\left[\left(\sum_{i=1}^N \gamma_i f(X_i)\right)^2\right] &= \sum_{i=1}^N \gamma_i^2 \mathbb{E}\left[f(X_i)^2\right] + \sum_{i\neq j} \gamma_i\gamma_j \mathbb{E}[f(X_i)]\cdot\mathbb{E}[f(X_j)] && (10)\\
&= \sum_{i=1}^N \gamma_i^2 \mathrm{Var}\left[f(X_i)\right] + \left(\sum_{i=1}^N \gamma_i \mathbb{E}[f(X_i)]\right)^2 && (11)\\
&\geq \sum_{i=1}^N \gamma_i^2 \mathrm{Var}\left[f(X_i)\right] && (12)\\
&\geq V_{\min}\|\boldsymbol{\gamma}\|_2^2. && (13)
\end{aligned}$$

In addition, from (11), by the Cauchy-Schwarz Inequality, we obtain

$$\mathbb{E}\left[\left(\sum_{i=1}^{N}\gamma_i f(X_i)\right)^2\right] \leq U_{\max}\|\boldsymbol{\gamma}\|_2^2 + \left(\sum_{i=1}^{N}\mathbb{E}[f(X_i)]^2\right)\|\boldsymbol{\gamma}\|_2^2 \tag{14}$$

$$\leq (N+1)U_{\max}\|\boldsymbol{\gamma}\|_2^2. \tag{15}$$

This completes the proof. □

Lemma 6. *Let $\mathbf{E} \in \mathbb{R}^{m\times D}$ be a standard Gaussian matrix, i.e., each entry of $\mathbf{E}$ is sampled from standard Gaussian distribution, and let $\mathbf{W} \in \mathbb{R}^{D\times D}$ be a fixed matrix that has no zero rows. In addition, for a fixed vector $\mathbf{b} \in \mathbb{R}^D$, let $\mathbf{B} = [\mathbf{b}, \mathbf{b}, \ldots, \mathbf{b}]^T \in \mathbb{R}^{m\times D}$. Then, when $m = \Omega(D)$, with probability $1 - e^{-\Omega(m)}$, it holds that*

$$\frac{1}{\sqrt{m}}\|f(\mathbf{E}\mathbf{W}^T + \mathbf{B})\|_2 \leq C\sqrt{D}, \tag{16}$$

where C is an absolute constant and the non-linear activation function $f(\cdot)$ is applied element-wise.

Proof. Let $\mathbf{w}_i^T \in \mathbb{R}^{1\times D}$ be the i-th row of $\mathbf{W}$. By the assumption that $\mathbf{W}$ has no zero rows, we have $\|\mathbf{w}_i\|_2 > 0$ for all $i \in [N]$. Let $\mathbf{H} = \mathbf{E}\mathbf{W}^T + \mathbf{B} \in \mathbb{R}^{m\times D}$. Then the (i,j)-th entry of $\mathbf{H}$, denoted h_{ij}, follows the $\mathcal{N}\big(b_j, \|\mathbf{w}_j\|_2^2\big)$ distribution. For any $\boldsymbol{\gamma} \in \mathcal{S}^{D-1}$ and any fixed i, from Lemma 3, we obtain that $\big(\sum_{j=1}^{D} f(h_{ij})\gamma_j\big)^2$ is sub-exponential with the sub-exponential norm being upper bounded by C, where C is some absolute constant. Let $E_{\boldsymbol{\gamma}}$ be the expectation of $\big(\sum_{j=1}^{D} f(h_{ij})\gamma_j\big)^2$. From Lemma 5, we obtain that it holds uniformly for all $\boldsymbol{\gamma} \in \mathcal{S}^{D-1}$ that

$$E_{\boldsymbol{\gamma}} \leq C'D, \tag{17}$$

where C' is an absolute constant. In addition, from Lemma 4, for any $\epsilon \in (0,1)$, we obtain

$$\mathbb{P}\left(\left|\frac{1}{m}\sum_{i=1}^{m}\left(\sum_{j=1}^{D} f(h_{ij})\gamma_j\right)^2 - E_{\boldsymbol{\gamma}}\right| \geq \epsilon\right) \leq 2\exp(-\Omega(m\epsilon^2)). \tag{18}$$

From Lemma 1, there exists an ϵ-net $\mathcal{N}_\epsilon$ of $\mathcal{S}^{D-1}$ with $\big|\mathcal{N}_\epsilon\big| \leq (1+2/\epsilon)^D$. Taking a union bound over $\mathcal{N}_\epsilon$, we have that when $m = \Omega\big(\frac{D}{\epsilon^2}\log\frac{1}{\epsilon}\big)$, with probability $1 - e^{-\Omega(m\epsilon^2)}$, it holds for all $\boldsymbol{\gamma} \in \mathcal{N}_\epsilon$ that

$$E_{\boldsymbol{\gamma}} - \epsilon \leq \frac{1}{m}\|f(\mathbf{E}\mathbf{W}^T + \mathbf{B})\boldsymbol{\gamma}\|_2^2 = \frac{1}{m}\sum_{i=1}^{m}\left(\sum_{j=1}^{D} f(h_{ij})\gamma_j\right)^2 \leq E_{\boldsymbol{\gamma}} + \epsilon. \tag{19}$$

Then, since (17) holds uniformly for all $\boldsymbol{\gamma} \in \mathcal{S}^{D-1}$, setting $\epsilon = \frac{1}{2}$, Lemma 2 implies that when $m = \Omega(D)$, with probability $1 - e^{-\Omega(m)}$,

$$\frac{1}{\sqrt{m}}\|f(\mathbf{E}\mathbf{W}^T + \mathbf{B})\|_2 \leq 2\max_{\boldsymbol{\gamma}\in\mathcal{N}_{1/2}}\frac{1}{\sqrt{m}}\|f(\mathbf{E}\mathbf{W}^T + \mathbf{B})\boldsymbol{\gamma}\|_2 \tag{20}$$

$$\leq 2\max_{\boldsymbol{\gamma}\in\mathcal{N}_{1/2}}\sqrt{E_{\boldsymbol{\gamma}} + \frac{1}{2}} \tag{21}$$

$$\leq C\sqrt{D}. \tag{22}$$

□

Lemma 7. *Let us use the same notation as in Lemma 6. When $m = \Omega(D^2\log D)$, with probability $1 - e^{-\Omega(D\log D)}$, it holds for* all *$\boldsymbol{\gamma} \in \mathbb{R}^D$ that*

$$\frac{1}{\sqrt{m}}\|f(\mathbf{E}\mathbf{W}^T + \mathbf{B})\boldsymbol{\gamma}\|_2 > \sqrt{\frac{V_{\min}}{2}}\|\boldsymbol{\gamma}\|_2, \tag{23}$$

where $V_{\min} = \min_{i\in[D]}\mathrm{Var}[f(X_i)]$ with $X_i \sim \mathcal{N}(b_i, \|\mathbf{w}_i\|_2^2)$.

Proof. It suffices to consider the case that $\boldsymbol{\gamma} \in \mathcal{S}^{D-1}$. From (19), we have that when $m = \Omega\big(\frac{D}{\epsilon^2}\log\frac{1}{\epsilon}\big)$, with probability $1 - e^{-\Omega(m\epsilon^2)}$, it holds for all $\boldsymbol{\gamma} \in \mathcal{N}_\epsilon$ that

$$\frac{1}{m}\|f(\mathbf{E}\mathbf{W}^T + \mathbf{B})\boldsymbol{\gamma}\|_2^2 \geq E_{\boldsymbol{\gamma}} - \epsilon. \tag{24}$$

In addition, from Lemma 5, we have that it hold uniformly for all $\boldsymbol{\gamma} \in \mathcal{S}^{D-1}$ that

$$E_{\boldsymbol{\gamma}} \geq V_{\min}. \tag{25}$$

Then, if setting $\epsilon = \frac{c}{\sqrt{D}}$ for some small absolute constant c, we have that when $m = \Omega\big(D^2\log D\big)$, with probability $1 - e^{-\Omega(D\log D)}$, for all $\boldsymbol{\gamma} \in \mathcal{N}_\epsilon$,

$$\frac{1}{\sqrt{m}}\|f(\mathbf{E}\mathbf{W}^T + \mathbf{B})\boldsymbol{\gamma}\|_2 \geq \sqrt{E_{\boldsymbol{\gamma}} - \epsilon}. \tag{26}$$

For any $\boldsymbol{\gamma} \in \mathcal{S}^{D-1}$, there exists an $\mathbf{s} \in \mathcal{N}_\epsilon$ such that $\|\mathbf{s} - \boldsymbol{\gamma}\|_2 \leq \epsilon$, and thus

$$\frac{1}{\sqrt{m}}\|f(\mathbf{E}\mathbf{W}^T + \mathbf{B})\boldsymbol{\gamma}\|_2 \geq \frac{1}{\sqrt{m}}\|f(\mathbf{E}\mathbf{W}^T + \mathbf{B})\mathbf{s}\|_2 - \frac{1}{\sqrt{m}}\|f(\mathbf{E}\mathbf{W}^T + \mathbf{B})(\boldsymbol{\gamma} - \mathbf{s})\|_2 \tag{27}$$

$$\geq \sqrt{E_{\boldsymbol{\gamma}} - \epsilon} - \frac{1}{\sqrt{m}}\|f(\mathbf{E}\mathbf{W}^T + \mathbf{B})\|_2 \cdot \epsilon \tag{28}$$

$$\geq \sqrt{E_{\boldsymbol{\gamma}} - \epsilon} - cC, \tag{29}$$

where (29) follows from Lemma 6 and the setting $\epsilon = \frac{c}{\sqrt{D}}$. Then, if c is set to be sufficiently small, from (25), we have for all $\boldsymbol{\gamma} \in \mathcal{S}^{D-1}$ that

$$\frac{1}{\sqrt{m}}\|f(\mathbf{E}\mathbf{W}^T + \mathbf{B})\boldsymbol{\gamma}\|_2 > \sqrt{\frac{V_{\min}}{2}}. \tag{30}$$

□

Lemma 8. *Let $\mathbf{Z} \in \mathbb{R}^{m\times D}$ be a matrix with independent zero-mean sub-Gaussian entries. Suppose that the sub-Gaussian norm of all entries of $\mathbf{Z}$ is upper bounded by some $\eta > 0$. Then, for any $\boldsymbol{\gamma} \in \mathbb{R}^D$, we have that with probability $1 - e^{-\Omega(m)}$,*

$$\frac{1}{\sqrt{m}}\|\mathbf{Z}\boldsymbol{\gamma}\|_2 \leq 2\eta\|\boldsymbol{\gamma}\|_2. \tag{31}$$

Proof. Let $\mathbf{z}_i^T \in \mathbb{R}^{1\times D}$ be the i-th row of $\mathbf{Z} \in \mathbb{R}^{m\times D}$. From the definition of sub-Gaussian norm, we have that if a random variable X is zero-mean sub-Gaussian with $\|X\|_{\psi_2} \leq \eta$, then

$$\eta = \sup_{p\geq 1} p^{-1/2}(\mathbb{E}[|X|^p])^{1/p} \geq 2^{-1/2}(\mathbb{E}[|X|^2])^{1/2}, \tag{32}$$

which implies

$$\mathbb{E}[X^2] \leq 2\eta^2. \tag{33}$$

Then, we have

$$\mathbb{E}[(\mathbf{z}_i^T\boldsymbol{\gamma})^2] \leq 2\eta^2\|\boldsymbol{\gamma}\|_2^2. \tag{34}$$

In addition, from Lemma 3, we also have that $(\mathbf{z}_i^T\boldsymbol{\gamma})^2$ is sub-exponential with the sub-exponential norm being upper bounded by $C\eta^2\|\boldsymbol{\gamma}\|_2^2$. Then, from Lemma 4, we obtain that for any $\epsilon \in (0,1)$, with probability $1 - e^{-\Omega(m\epsilon^2)}$,

$$\left|\frac{1}{m}\sum_{i=1}^m\left((\mathbf{z}_i^T\boldsymbol{\gamma})^2 - \mathbb{E}\left[(\mathbf{z}_i^T\boldsymbol{\gamma})^2\right]\right)\right| \leq C\epsilon\eta^2\|\boldsymbol{\gamma}\|_2^2, \tag{35}$$

which implies

$$\frac{1}{m}\|\mathbf{Z}\boldsymbol{\gamma}\|_2^2 = \frac{1}{m}\sum_{i=1}^m(\mathbf{z}_i^T\boldsymbol{\gamma})^2 \leq \frac{1}{m}\sum_{i=1}^m\mathbb{E}\left[(\mathbf{z}_i^T\boldsymbol{\gamma})^2\right] + C\epsilon\eta^2\|\boldsymbol{\gamma}\|_2^2 \leq 2\eta^2\|\boldsymbol{\gamma}\|_2^2 + C\epsilon\eta^2\|\boldsymbol{\gamma}\|_2^2, \tag{36}$$

where the last inequality follows from (33). Setting $\epsilon = \min\{\frac{2}{C}, 1\}$, we obtain the desired result. □

4.4. Proof of Theorem 1

Let $\mathbf{E} = [\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_m]^T \in \mathbb{R}^{m \times D}$. Then $\mathbf{A}$ can be written as $\mathbf{A} = f(\mathbf{E}\mathbf{W}^T + \mathbf{B})$, where $\mathbf{B} = [\mathbf{b}, \ldots, \mathbf{b}]^T \in \mathbb{R}^{m \times D}$. Then, we have

$$\sqrt{m}\|\hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}^*\|_2 < \sqrt{\frac{2}{V_{\min}}} \left\| f(\mathbf{E}\mathbf{W}^T + \mathbf{B})(\hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}^*) \right\|_2 \tag{37}$$

$$= \sqrt{\frac{2}{V_{\min}}} \|\mathbf{A}(\hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}^*)\|_2 \tag{38}$$

$$\leq \sqrt{\frac{2}{V_{\min}}} (\|\mathbf{A}\hat{\boldsymbol{\gamma}} - \mathbf{y}\|_2 + \|\mathbf{y} - \mathbf{A}\boldsymbol{\gamma}^*\|_2) \tag{39}$$

$$\leq \sqrt{\frac{8}{V_{\min}}} \|\mathbf{y} - \mathbf{A}\boldsymbol{\gamma}^*\|_2 \tag{40}$$

$$= \sqrt{\frac{8}{V_{\min}}} \|\mathbf{Z}\boldsymbol{\gamma}^*\|_2 \tag{41}$$

$$\leq \frac{4\sqrt{2m\eta}}{\sqrt{V_{\min}}} \cdot \|\boldsymbol{\gamma}^*\|_2, \tag{42}$$

where (37) follows from Lemma 7, (39) follows from the triangle inequality, (40) follows from the condition that $\hat{\boldsymbol{\gamma}}$ minimizes (2), (41) follows from $\mathbf{y} = (\mathbf{A} + \mathbf{Z})\boldsymbol{\gamma}^*$. Finally, we use Lemma 8 to obtain (42).

5. More Visualization Results

In this section, we present additional samples generated by the DiT-XL/2 + DiffFit fine-tuning. Specifically, we demonstrate its efficacy by generating high-quality images with a resolution of 512×512 on the ImageNet dataset, as illustrated in Figures 9, 10, and 11. Additionally, we showcase the model's capability on 8 downstream tasks by displaying its image generation results with a resolution of 256×256, as depicted in Figures 12 through 19.

Correspondence to {xie.enze, li.zhenguo}@huawei.com

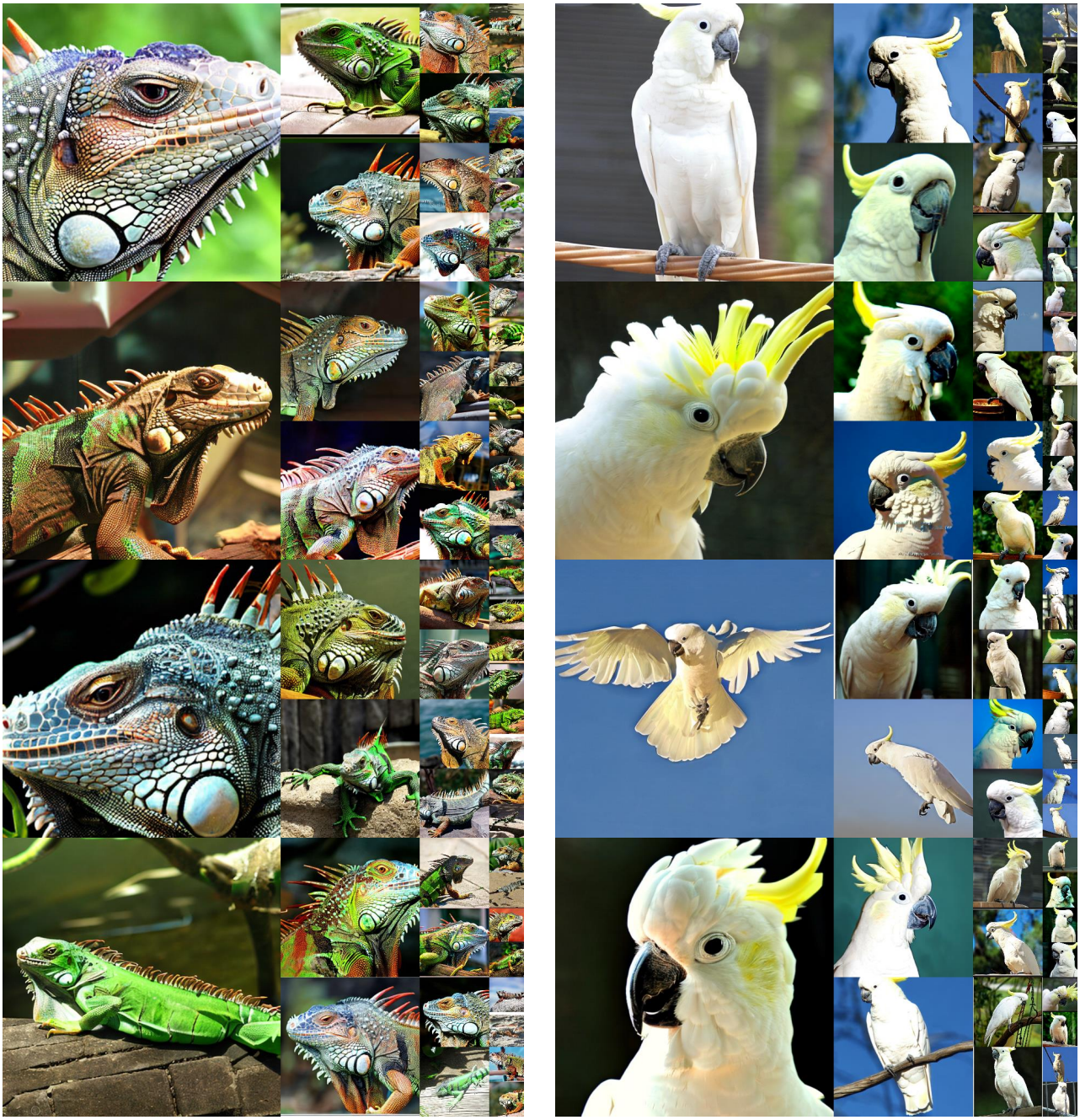

Figure 9: Visualization of DiffFit on ImageNet 512×512. Classifier-free guidance scale = 4.0, sampling steps = 250.

Figure 10: Visualization of DiffFit on ImageNet 512×512. Classifier-free guidance scale = 4.0, sampling steps = 250.

Figure 11: Visualization of DiffFit on ImageNet 512×512. Classifier-free guidance scale = 4.0, sampling steps = 250.

Figure 12: Visualization of DiffFit on Food 101.
Classifier-free guidance scale = 4.0, sampling steps = 250.

Figure 13: Visualization of DiffFit on ArtBench 10.
Classifier-free guidance scale = 4.0, sampling steps = 250.

Figure 14: Visualization of DiffFit on Stanford Cars. Classifier-free guidance scale = 4.0, sampling steps = 250.

Figure 15: Visualization of DiffFit on Caltech 101. Classifier-free guidance scale = 4.0, sampling steps = 250.

Figure 16: Visualization of DiffFit on DF20M. Classifier-free guidance scale = 4.0, sampling steps = 250.

Figure 17: Visualization of DiffFit on Flowers 102. Classifier-free guidance scale = 4.0, sampling steps = 250.

Figure 18: Visualization of DiffFit on CUB-200-2011. Classifier-free guidance scale = 4.0, sampling steps = 250.

Figure 19: Visualization of DiffFit on SUN 397. Classifier-free guidance scale = 4.0, sampling steps = 250.

References

[1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, 2014. 2

[2] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, 2018. 1

[3] Minshuo Chen, Kaixuan Huang, Tuo Zhao, and Mengdi Wang. Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data. *arxiv*, 2023. 5

[4] Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *ICLR*, 2023. 5

[5] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *ICLR*, 2023. 4

[6] Valentin De Bortoli. Convergence of denoising diffusion models under the manifold hypothesis. *TMLR*, 2022. 5

[7] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. In *NeurIPS*, 2021. 5

[8] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *ICLR*, 2019. 4

[9] Weihao Gao, Ashok V Makkuva, Sewoong Oh, and Pramod Viswanath. Learning one-hidden-layer neural networks under general input distributions. In *AISTATS*, 2019. 4

[10] Rong Ge, Jason D Lee, and Tengyu Ma. Learning one-hidden-layer neural networks with landscape design. In *ICLR*, 2018. 4

[11] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007. 2

[12] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *NeurIPS*, 2021. 4

[13] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV workshops*, 2013. 3

[14] Holden Lee, Jianfeng Lu, and Yixin Tan. Convergence for score-based generative modeling with polynomial complexity. In *NeurIPS*, 2022. 5

[15] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 1

[16] Peiyuan Liao, Xiuyu Li, Xihui Liu, and Kurt Keutzer. The artbench dataset: Benchmarking generative models with artworks. *arXiv*, 2022. 3

[17] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023. 4

[18] Zhaoqiang Liu, Jiulong Liu, Subhroshekhar Ghosh, Jun Han, and Jonathan Scarlett. Generative principal component analysis. In *ICLR*, 2022. 8

[19] Zhaoqiang Liu and Jonathan Scarlett. The generalized lasso with nonlinear observations and generative priors. In *NeurIPS*, 2020. 9

[20] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, 2008. 3

[21] Lukáš Picek, Milan Šulc, Jiří Matas, Thomas S Jeppesen, Jacob Heilmann-Clausen, Thomas Læssøe, and Tobias Frøslev. Danish fungi 2020-not just another image recognition dataset. In *WACV*, 2022. 2

[22] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv*, 2022. 1

[23] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arxiv*, 2023. 4

[24] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv*, 2023. 2

[25] Luca Venturi, Afonso S Bandeira, and Joan Bruna. Spurious valleys in one-hidden-layer neural network optimization landscapes. *JMLR*, 2019. 4

[26] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arxiv*, 2010. 8, 9

[27] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge J. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 3

[28] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, 2010. 2

[29] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv*, 2023. 1

[30] Xiao Zhang, Yaodong Yu, Lingxiao Wang, and Quanquan Gu. Learning one-hidden-layer relu networks via gradient descent. In *AISTATS*, 2019. 4

[31] Kai Zhong, Zhao Song, Prateek Jain, Peter L Bartlett, and Inderjit S Dhillon. Recovery guarantees for one-hidden-layer neural networks. In *ICML*, 2017. 4