FouRA: Fourier Low Rank Adaptation

Shubhankar Borse[*§] **Shreya Kadambi**[*§] **Nilesh Prasad Pandey**[†] **Kartikeya Bhardwaj**
Viswanath Ganapathy[†] **Sweta Priyadarshi**[†] **Risheek Garrepalli** **Rafael Esteves**
Munawar Hayat[§] **Fatih Porikli**[§]
Qualcomm AI Research[‡]
[§]{sborse, skadambi, mhayat, fporikli}@qti.qualcomm.com

Abstract

While Low-Rank Adaptation (LoRA) has proven beneficial for efficiently fine-tuning large models, LoRA fine-tuned text-to-image diffusion models lack diversity in the generated images, as the model tends to copy data from the observed training samples. This effect becomes more pronounced at higher values of adapter strength and for adapters with higher ranks which are fine-tuned on smaller datasets. To address these challenges, we present FouRA, a novel low-rank method that learns projections in the Fourier domain along with learning a flexible input-dependent adapter rank selection strategy. Through extensive experiments and analysis, we show that FouRA successfully solves the problems related to data copying and distribution collapse while significantly improving the generated image quality. We demonstrate that FouRA enhances the generalization of fine-tuned models thanks to its adaptive rank selection. We further show that the learned projections in the frequency domain are decorrelated and prove effective when merging multiple adapters. While FouRA is motivated for vision tasks, we also demonstrate its merits for language tasks on commonsense reasoning and GLUE benchmarks.

1 Introduction

Figure 1: **Distribution collapse with LoRA**. Visual results generated by the Realistic Vision 3.0 model trained with LoRA and FouRA, for "*Blue Fire*" and "*Origami*" style adapters **across four seeds**. While LoRA images suffer from distribution collapse and lack diversity, we observe diverse images generated by FouRA.

Parameter-Efficient FineTuning (PEFT) [27] methods such as Low-Rank Adaptation [17] provide a promising solution to quickly adapt large foundation models, including large vision models (LVMs) and large language models (LLMs) to new tasks [26, 22, 3]. The LoRA module has an elegant design, allowing quick adaptation to new styles or concepts without changing the underlying base model, thus effectively retaining previous knowledge and preventing catastrophic forgetting.

[*]These authors contributed equally to this work.
[†]Work done while employed at Qualcomm AI Research.
[‡]Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-2284

While LoRAs are highly effective in quickly adapt to new styles, they exhibit multiple challenges, with the rank of LoRA modules being a highly sensitive parameter. As LoRA is built for adapting to new tasks using a small training set, it tends to overfit to the distribution of small training set when the rank is high. Recent works [39, 40] observed that when diffusion models overfit to a small training set, they demonstrate a tendency to repeatedly "copy" few samples from the training set. LoRAs trained on smaller data therefore tend to generate data copying artifacts, also known as distribution collapse. The generated images lack diversity, and the phenomenon is very similar to mode collapse observed in GANs. We illustrate this tendency in Fig. 1, specially at high values of adapter strength α across different seeds. Additionally, as the rank reduces, the strength of the adapter reduces, and LoRA has a reduced ability to generate diverse images due to underfitting. Hence, the rank is a very sensitive parameter.

Gating mechanisms have been proposed [3] to produce a dynamic rank at every layer, to provide flexibility to the adapter in LLM tasks. However, we argue that dynamic rank reduction is still not flexible for vision tasks as the rank is computed during training and does not vary at inference. We observe that text-to-image diffusion models greatly benefit from a rank adaptation mechanism which can also vary during inference, along the diffusion time steps. Furthermore, while all the previous works apply low-rank adaptation in the feature space, we argue that there is a transform domain over which fine-tuning low-rank adaptation modules generates much richer representations. We provide theoretical and analytical evidence to show that low-rank adaptation in the frequency domain produces a highly compact representation, effectively reducing the generalization error. Hence, this can potentially push the adaptive rank selection mechanism to generalize better, not only reducing the risk of underfitting when rank reduces, but also overfitting at higher ranks. Additionally, there have been attempts to merge multiple LoRA concepts and/or styles as a linear weighted combination of multiple LoRAs [34]. Recent works [45, 12, 23] empirically show that this approach is prone to noisy and inaccurate outputs, and propose **joint** finetuning the adapters with learnable gates in the low rank subspace. However, we argue that jointly training multiple LoRA modules is highly restrictive and equally tedious for practical use-cases requiring flexibility in combining multiple different LoRAs. Our developed approach of gating in frequency domain enables flexible mixing of multiple adapters.

In this paper, we propose FouRA (Fourier Low Rank Adaptation), a PEFT technique to address the aforementioned challenges of LoRA. We transform the input features to the frequency domain, and apply both the down-projection (to a lower rank) and the up-projection (back to the higher rank) in this frequency domain. During inference, we fold the adapter strength α into the low rank subspace. FouRA learns an adaptive mask inside the low-rank subspace to dynamically drop certain frequency transformed basis, effectively varying the rank for each layer. The adaptive mask selection is input dependant, and varies during the diffusion process. Through rigorous analysis, we show that FouRA provides clear benefits over LoRA (and other adaptive gating methods), and generates high quality diverse images.We show for lower ranks increasing the effect of adapter weights in FouRA does not deteriorate the representation power of original model. Additionally, we show that FouRA provides a rich disentangled orthogonal basis to Low Rank Adapters in the frequency domain, making it beneficial for merging multiple styles. Our contributions are summarized as:

- We introduce FouRA, the first low-rank adapter module that performs the low rank transforms in the frequency domain along pixel or channel dimensions of the feature space.
- We propose an adaptive learnable masking strategy in the frequency domain that flexibly varies the effective rank for every FouRA layer in the network, thus enabling the model to generalize well, even when the size of training set is very small.
- We demonstrate that FouRA successfully provides a decorrelated orthonormal basis to Low Rank Adapters in the frequency domain, making it highly beneficial for merging two styles or concepts, without the need for joint training.
- Through extensive experiments and theoretical analysis, we demonstrate how FouRA consistently produces a diverse set of aesthetically improved images compared to LoRA, and is equally effective for LLM tasks.

2 Related Work

Text-to-Image Diffusion Models: Multiple diffusion based image generative models have been proposed recently [33, 31, 6], [32, 29, 36, 30]. These models exhibit excellent text-to-image generation ability and can be adapted to new styles using LoRA [17].

Fourier Transforms in Generative Literature: Recent work [21] shows that the latents of the denoising models trained on sufficient data lie on adaptive basis with oscillating patterns. Other works have shown that we can use fourier operators for non parametric regression tasks and cast self attention as a kernel regression problem. [28] shows that it offers smoother representations over the input and better captures the correlations between query and keys. [24] has shown that Fourier spectral filters operate in the continuous domain and work well in representing images as continuous functions. Further convolutions in spatial domain can be represented as multiplications in the Fourier space thus spectral filters can act as global convolution operator. A concurrent work on language models [10] has proposed parameter-efficient fine-tuning in the Fourier Domain.

Many works have analysed the eigen spread of signal transformed to harmonic basis. [1], analysed the effect of applying these transforms on a signal sampled from a Markovian process and show that Fourier transforms decorrelates such as signal in least mean square setting.

Low Rank Adaptation: LoRAs [17] suffer from a tradeoff between fidelity and diversity of generated images. [3] tried to alleviate this problem by sparse regularization. SVDiff [14] explicitly only updates the singular values while retaining the subspaces. In a high rank setting this method is acceptable. However, in FouRA we are learning in a low rank subspace. Other works like AdaLORA [48], [46] applied to language models, further parameterized the weight matrices using SVD and jointly optimized for eigen vectors and the singular values through importance scoring metric. O-lora [42] computes orthogonal gradient spaces between different tasks letting the model sequentially adapt to new tasks without catastrophic forgetting. [3] applies proximal gradient gating in the loss function to learn important subspaces and mask out the remaining ones. While all these papers directly operate by constraining the subspace of the weight matrices, we show in our paper that the Fourier domain implicitly enforces these properties without any constraints in the optimization. We show that applying gating in the frequency domain provides a more compact representation with stable generalization error bounds. In addition results in lower effective rank for each layer. We also show that the learnt spaces across different adapters also have decorrelated basis. MoLE [45], ZipLoRA[37] and Mix of Show [12, 50] explore various strategies to merge LoRAs. This is done using either supervised or self-supervised objectives for jointly training weights corresponding to both adapters. As the number of adapters grow, we argue that the two-stage method to merge adapters is not flexible and quite tedious. FouRA on the other hand does not require any fine-tuning, and is truly a training-free approach to merge multiple adapters.

Disentangled spaces for editing [43] [13] have explored diffusion models for disentangled interpretable latent representation. While LoRAs have been proposed for personalization, [9] proposed a way to do fine-grained editing of images while still preserving the features of the original image. They identify semantic directions and traverse on the latent space on these directions. Concept sliders have been applied to real applications such as fixing distortions in diffusion generated images. We show in our work that our method identifies more compact disentangled representations over LoRA, thus providing more performance improvements over fine-grained edits.

3 Proposed Approach

3.1 Formulation of Low Rank Adaptation

We illustrate the base LoRA module in Fig. 2. Consider the original set of pre-trained weights $\mathbf{W_0} \in \mathbb{R}^{k_1 \times k_2}$ where k_1 and k_2 represent the input and output embedding dimensions respectively. LoRA modules consist of the down layer $\mathbf{A} \in \mathbb{R}^{k_1 \times r}$ and the up layer $\mathbf{B} \in \mathbb{R}^{r \times k_2}$, projecting the input features to and from the low-rank subspace of rank r. Consider an input feature $\mathbf{z_{in}} \in \mathbb{R}^{d \times k_1}$, where d is the number of input tokens, the output after the low-rank adaptation $\mathbf{z_{out}} \in \mathbb{R}^{d \times k_2}$ is given as $\mathbf{z_{out}} = \mathbf{z_{og}} + \alpha \mathbf{z_{lora}} = \mathbf{W_0 z_{in}} + \alpha \mathbf{B A z_{in}}$. Here, $\mathbf{z_{og}}$ and $\mathbf{z_{lora}}$ are the outputs from the original and low-rank branches respectively, and α is a scalar to blend the two branches. We denote the learned adapter matrices as $\Delta \mathbf{W_{lora}} = \mathbf{BA}$ as in [17].

3.2 Low Rank Adaptation in the Frequency Domain

The projection to and from a low-rank subspace is prone to information loss. To mitigate this, we propose to transform the inputs to a domain which contains an inherently compact representation, i.e. the frequency domain. We are motivated by the fact that transforming to the frequency domain

 https://doi.org/10.52202/079017-2284

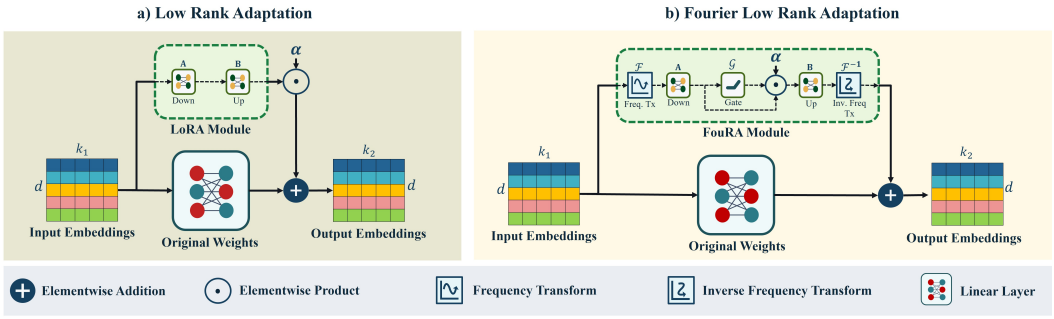


Figure 2: **LoRA v/s FouRA**. For FouRA, we transform feature maps to frequency domain, where we learn up and down adapter projections along-with our proposed adaptive rank gating module.

preserves valuable information, due to its inherent de-correlation capabilities [11, 16]. We validate this further by analyzing the effects of the frequency transform on the model weights in Sec. 4.1.

Given the pre-trained weight matrix $\mathbf{W_0}$, we apply the low rank transforms $\mathbf{B}$ and $\mathbf{A}$ in the frequency domain. Inspired by [38], we fold the blending parameter α inside the low-rank subspace, effectively acting as a scaling factor in the frequency domain. We apply the frequency transforms as follows.

$$\mathbf{z_{out}} = \mathbf{z_{og}} + \mathbf{z_{foura}} = \mathbf{W_0 z_{in}} + \mathcal{F}^{-1}(\mathbf{B}\alpha\mathbf{A}\mathcal{F}(\mathbf{z_{in}})) \tag{1}$$

Here, $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ are the normalized forward and inverse frequency transforms respectively.

3.3 Frequency Transforms

We investigate the properties of Discrete Fourier Transform (DFT) and Discrete Cosine Transform (DCT) in the low rank space. We apply 1D DFT to the embedding dimension $k_1 \in (0, K)$ before the subspace decomposition. Given input $z_{in} \in \mathbb{R}^{d \times k_1}$ to the adapter branch , we expand $\mathcal{F}$ in Eq. (5) as,

$$\mathbf{Z_{k_1}}(f) = \mathcal{F}(\mathbf{z_{in}})_{d \times k_1} = \frac{1}{\sqrt{k_1}} \sum_{k=0}^{k_1-1} e^{-j\frac{2\pi f_r k}{k_1}} \mathbf{z_{in}}(k), f_r : \forall r \in (0, 1...k_1 - 1). \tag{2}$$

Where f_r is the frequency of the basis represented by DFT. As we do not apply any padding, the dimension of the transform preserves the dimension of $\mathbf{z_{in}}$. In our experiments, we apply the 1-D transform on the embedding dimension k_1 for each token on both self and cross attention layers.

To motivate the idea of generalizing FouRA across tasks such as targeted editing [9], where disentangled latent space is required to gain control over generated images, we further explored Discrete Cosine Transform (DCT) with compact subspaces (eigen spread), which leads to less overfitting. We later show in App. B.1 and Fig. 4 that the subspaces of FouRA are more uncorrelated from each other. We observe that for certain tasks, DCT provides a smoother representation as the implicit window is twice that of DFT signals. For a given a finite length signal $\mathbf{z_{in}} \in \mathrm{R}^{d \times k_1}$, we compute DCT as follows. We first construct a double length even signal by

$$\tilde{\mathbf{z_{in}}}(d, k_1) = \begin{cases} \mathbf{z_{in}}(d, k_1), & 0 \leq k_1 \leq K \\ \mathbf{z_{in}}(d, 2K - k_1 - 1), & K \leq k_1 \leq 2K - 1, \end{cases} \tag{3}$$

The DCT is then computed as the DFT of $\tilde{\mathbf{z_{in}}}$.

3.4 Adaptive Rank Gating Method

LoRA methods pre-define the rank for all layers. Recent method [3] has an adaptive rank during training, which is however fixed at inference time, thus lacking flexibility. In our approach, we propose a learned adaptive gating mechanism, which can vary each layers rank during training and inference, dependent upon the inputs. We introduce our learnable gating mechanism $\mathcal{G}(\cdot)$ inside the low-rank subspace within the frequency domain. Consider the low-rank representation denoted as $\mathbf{z_{lr}} \leftarrow \mathbf{A}\mathcal{F}(\mathbf{z_{in}}) \in \mathbb{R}^{d \times r}$, our gating operation is defined as,

$$\mathcal{G}(\mathbf{z_{lr}}) = \begin{cases} 1, & \text{if } \mathcal{S}(\mathcal{H}(\mathbf{G z_{lr}})) == 1 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

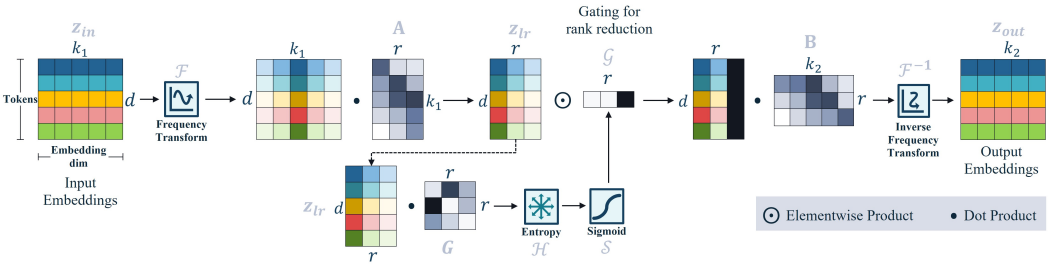


Figure 3: **Operational diagram of FouRA**. Illustrating the components of Eq. 5.

Here, $\mathcal{H}(\cdot)$ and $\mathcal{S}(\cdot)$ represent entropy and sigmoid functions respectively, $\mathbf{G}$ represents the weights of a learnable multi-layer perceptron (MLP), $\mathcal{G}$ is a function to learn a weighting for every singular value in the low-rank subspace. The FouRA output, illustrated in Fig. 3, is then given by,

$$\mathbf{z_{out}} = \mathbf{z_{og}} + \mathbf{z_{foura}} = \mathbf{W_0 z_{in}} + \mathcal{F}^{-1}(\mathbf{B}\alpha\mathcal{G}(\mathbf{z_{lr}}) \cdot \mathbf{A}\mathcal{F}(\mathbf{z_{in}})) \tag{5}$$

The learned FouRA adapter weights are $\Delta\mathbf{W_{foura}} = \mathcal{F}^{-1}(\mathbf{B}\mathcal{G}(\mathbf{z_{lr}})\mathcal{F}(\mathbf{A}))$, as per notation in Sec. 3.1.

We conduct further analysis of our proposed gating function in Sec. 4.2, analyzing its behaviour across diffusion time-steps and various resolutions. Further, we demonstrate its efficacy over both fixed LoRA and recent Adaptive Rank selection methods which are fixed at inference (SoRA [3]).

3.5 Combining multiple adapters

Merging of LoRA adapters has multiple practical use-cases [34]. The method we use to merge two adapters varies according to the task.

Text-to-Image Style Transfer: Following the standard method, we merge two FouRA style based adapters using a linear combination of the outputs of adapter $\Delta W_1.\mathbf{z_{in}}$ and $\Delta W_2.\mathbf{z_{in}}$ during inference.

Image editing using Concept Sliders: Similar to [9], we perform concept slider evaluations for text based editing using FouRA in Sec. 5.3. Given n concept sliders, we define $c_{n,j}$ concept for n^{th} slider (e.g "very old") and $\tilde{c}_{n,i}$ as the negative concept (e.g " very young"). We composite the adapters in the epsilon ϵ space, with composed score function $\hat{\epsilon}$, and sample from the factorized distribution $p(\mathbf{x}/(\tilde{c}_i, c_j))$

$$\hat{\epsilon}(\mathbf{x}) = \epsilon_\theta(\mathbf{x}) + \sum_n w_n(\epsilon_\theta(\mathbf{x}, c_{n,j}) - \epsilon_\theta(\mathbf{x}, c_{n,i})) \tag{6}$$

For merging of two styles, as well as composition of two concept adapters across different strengths α, we notice that the feature spaces of FouRA adapters are less entangled as compared to LoRA. Further analysis is present in Appendix B.4 and B.2.

4 Theoretical Analysis

4.1 Frequency Domain Fine Tuning

Frequency domain transforms decorrelate input representations, minimize spectral redundancy [47], and are effective in compression since they concentrate most of the energy in a few coefficients [16]. Learning in the spectral domain is shown to enable faster convergence and sparser weight matrices [11]. Motivated by these advantages, we propose to fine-tune adapters in the frequency domain.

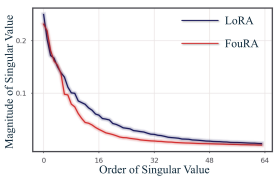
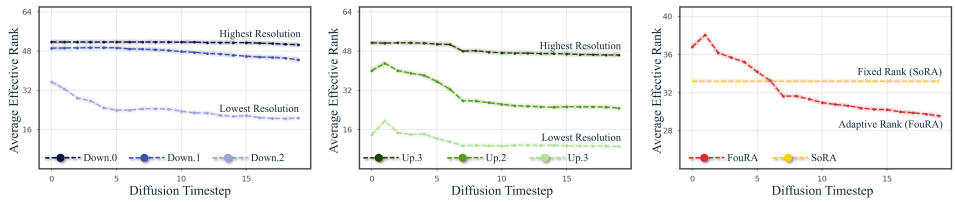


Figure 4: Singular value spread for FouRA v/s LoRA.

Singular Value Distribution Analysis: Consider a weight matrix $\mathbf{W}$. The singular value decomposition of this matrix is represented as $\mathbf{UDV^T}$, where $\mathbf{U} \in \mathbb{R}^{k_1 \times k_1}$, $\mathbf{V} \in \mathbb{R}^{k_2 \times k_2}$ are orthonormal matrices and $\mathbf{D} \in \mathbb{R}^{k_1 \times k_2}$ is a matrix, containing the singular values of $\mathbf{W}$, $\sigma_i \forall i \in \{\mathbb{N}^{min(k_1,k_2)}\}$. Considering an r rank approximation of $\mathbf{W}$, we denote the singular values as $\{\sigma_1, \sigma_2 ... \sigma_r\}$, arranged in descending order, and the corresponding diagonal matrix as $\mathbf{D_r}$. The r-rank approximation of $\mathbf{W}$ is hence computed as $LR_r(\mathbf{W}) = \mathbf{UD_rV^T}$.

https://doi.org/10.52202/079017-2284



a) FouRA UNet Downsampling Blocks **b) FouRA UNet Upsampling Blocks** **c) FouRA Unet v/s SoRA UNet**

Figure 5: **Average Effective Rank of FouRA**. Figure a. and b. shows plots for the average effective rank for various layers of the FouRA U-Net (Darker lines correspond to higher resolutions) and Figure c. compares the average effective rank of FouRA to SoRA. FouRA's effective rank reduces with the feature resolution, and it also reduces as the diffusion process proceeds, owing to lesser changes required towards the end.

Lemma 4.1. *Considering two adapters* $\Delta\mathbf{W_1}$ *and* $\Delta\mathbf{W_2}$ *and their corresponding sets of singular values* $\{\sigma_{1,i}\}$ *and* $\{\sigma_{2,i}\}$*. The adapter* $\Delta\mathbf{W_1}$*, will admit* r *rank approximation with lower error than* $\Delta\mathbf{W_2}$ *if* $\sigma_{1,i} < \sigma_{2,i}$ *for all* $i \geq r$*.*

We provide a proof for the above lemma in Appendix B.1. We empirically analyze the distribution of singular values for r rank approximations of $\Delta\mathbf{W_{lora}}$ and $\Delta\mathbf{W_{foura}}$ (without adaptive masking) for the last layer of our trained UNet model in Fig. 4. FouRA has a more compact spread of singular values as compared to LoRA. Hence, using Lemma 4.1, we can say that the accumulated error for a LoRA adapter with a low-rank approximation will be greater than the a FouRA adapter with the same rank.

4.2 Gated Frequency Domain Fine Tuning

Motivated by observations in [3, 25], our proposed rank gating mechanism intends to vary the effective rank of each low-rank adapter in the network. We describe effective rank per layer as the number of singular values which are not masked out by the learned gating function. Using observations from [7, 25], we propose the following Lemma:

Lemma 4.2. *Consider an adapter* $\Delta\mathbf{W}$ *with a rank higher than the required rank to fit a training data distribution. The upper-bound of generalization error* $\mathcal{R}$ *for fine-tuning this adapter reduces as the effective rank of the adapter reduces. After reducing to a certain value of effective rank, the upper-bound of generalization error will increase as rank reduces further.*

Corollary 4.2.1. *Additionally, the generalization bound is more stable when the singular value distribution of adapter weights* $\Delta\mathbf{W}$ *is more compact.*

We provide a proof in Appendix B.2. The effectiveness of variable rank selection can be justified using Lemma 4.2. As LoRA rank reduces, the model tends to underfit. However, increasing the rank above the required rank to fit a training distribution leads to overfitting, which reduces the models performance. Dynamically determining the effective rank in every layer produces promising results, as it provides a learnable trade-off between generalization and overfitting.

In Fig. 5, we plot FouRA average effective ranks for a denoising UNet over 20 iterations of the reverse diffusion process. Our analysis reveals that the effective rank learnt for high-resolution layers is higher than low-resolution layers. Furthermore, the effective rank reduces as the denoising process continues. This essentially means that noisy inputs require more singular values to update. We further observe in Fig. 9 that our proposed adaptive masking (which varies in inference time) significantly outperforms methods such as SoRA (which freezes its masks after training).

Furthermore, from Corollary 4.2.1 and a consequence of the property observed in Fig. 4, as FouRA obtains compact spread of singular values, we can determine that the generalization bound is more stable in the frequency domain for lower effective ranks, as compared to the feature space. We verify this in Fig. 9 as FouRA outperforms SoRA and LoRA with our proposed adaptive masking. The **data copying artifacts** observed for LoRA model in Fig. 1 are a consequence of overfitting. This was observed by recent works targeting Digital Forgery [39, 40]. As FouRA significantly reduces the generalization error, it can generate a diverse set of images. Additionally, we also observe in App. E.1.1 that FouRA is able to generalize better on unseen concepts, as compared to LoRA.

https://doi.org/10.52202/079017-2284

Figure 6: **FouRA v/s LoRA:** The prompt on the left is "a football in a field" and on the right is "man in a mythical forest". While staying more faithful to the adapter style, FouRA outputs look aesthetically better than LoRA, which have obvious artifacts at high values of α. Additional results are in Appendix E.

4.3 Subspace Learning

In App. B, we provide a subspace perspective to verify empirically and theoretically that FouRA learns subspaces which are more decorrelated from the base model weights, as compared to LoRA. A higher emphasis on the set of learnt subsapces enables FouRA to learn new tasks without catastrophic forgetting. Additionally, we attribute the strong merging capabilities of different FouRA adapters to their disentangled and decorrelated subspaces learned by respecitve FouRAs.

5 Experiments

5.1 Experimental setup

Datasets: For style transfer, we evaluate FouRA on four datasets collected from public domains, including *Bluefire* , *Paintings*, *3D* and *Origami* styles, see Appendix C.1.3 for details. Our results are averaged over 30 random seeds, and a total of 1530 images. For evaluations on composite sliders, similar to [9], we train 3 sliders *"Age"*, "*Hair*" *"Surprised'* and composite experiments combining both *"Age"* and "*Hair*" . While our approach is motivated for vision tasks, we also evaluate FouRA on language tasks and assess the performance of our adapter on MNLI, CoLA, SST2, STSB, MRPC and QNLI tasks from the GLUE benchmarks. We also evaluate on Commonsense Reasoning benchmarks BoolQ, PIQA, SIQA, HellaSwag, WinoGrande, ARC and OBQA. See App. C.1 for details.

Models: For text-to-image generation experiments, we employ Stable Diffusion-v1.5 [33], using both the base model weights and RealisticVision-v3.0 checkpoints for style transfer tasks. For concept editing, we train on Stable Diffusion-v1.5 [33] base weights. We use DeBERTAv3-Base [15] for General Language Understanging tasks and Llama3-8B [4] for Commonsense Reasoning tasks. See App. C for additional implementation details.

Metrics: For quantifying the quality of images generated by FouRA and LoRA finetuned diffusion models, we report HPSv2.1 [44] and LPIPS diversity [49] scores. The HPSv2 metric evaluates the measure of the image quality, and alignment with the prompt/style. LPIPS diversity score captures the diversity within all possible pairs of generated images across seeds. We provide an in-depth analysis of these metrics in Appendix D. For the image editing task, we compare edited images using LPIPS similarity (compared to the base image). For language models, we report on the General Language Understanding Evaluation (GLUE) benchmarks [41], see details in App. C.2. On commonsense reasoning tasks, we report Accuracy.

5.2 Text-to-Image Stylized Generation

In Fig. 6, we show visual results of LoRA and FouRA on the *Paintings* and *Bluefire* style tasks. FouRA is able to generate high quality images as compared to LoRA over a range of adapter strengths α. We observe that LoRA suffers from artifacts at high values of α in case of the Paintings adapter. Tab. 2 compares LPIPS Diversity and HPSv2 scores for all models, showing that FouRA significantly outperforms LoRA on both the metrics. Our analysis in App. D shows that this gap in LPIPS-diversity and HPS scores is quite significant, specially for higher α values, FouRA shows significant gains compared to LoRA. This is likely because at lower α values, the adapter effect would be reduced and

https://doi.org/10.52202/079017-2284

Dataset	Base Model	Adapter	LPIPS Diversity(↑) $\alpha = 1$	$\alpha = 0.8$	$\alpha = 0.6$	HPSv2 score(↑) $\alpha = 1$	$\alpha = 0.8$	$\alpha = 0.6$
Paintings (630 Images)	Stable Diffusion-v1.5	LoRA	38.3 ± 3.6	43.0 ± 3.2	43.6 ± 3.6	22.3 ± 1.7	25.3 ± 1.9	27.2 ± 2.9
		FouRA	$\mathbf{43.9 \pm 3.7}$	$\mathbf{44.1 \pm 3.8}$	$\mathbf{45.7 \pm 3.8}$	$\mathbf{25.2 \pm 1.6}$	$\mathbf{27.1 \pm 1.8}$	$\mathbf{28.0 \pm 2.4}$
	Realistic Vision-v3.0	LoRA	38.3 ± 3.5	37.8 ± 3.6	39.2 ± 3.7	24.6 ± 1.8	27.7 ± 1.8	30.3 ± 1.7
		FouRA	$\mathbf{44.2 \pm 3.7}$	$\mathbf{44.5 \pm 4.0}$	$\mathbf{44.6 \pm 3.9}$	$\mathbf{28.4 \pm 1.8}$	$\mathbf{30.6 \pm 1.5}$	$\mathbf{32.0 \pm 1.4}$
Blue-Fire (900 Images)	Stable Diffusion-v1.5	LoRA	47.8 ± 3.7	48.4 ± 3.9	49.5 ± 4.2	28.6 ± 2.1	30.4 ± 2.0	30.6 ± 2.2
		FouRA	$\mathbf{50.3 \pm 3.0}$	$\mathbf{50.8 \pm 3.2}$	$\mathbf{51.5 \pm 3.6}$	$\mathbf{29.7 \pm 1.9}$	$\mathbf{30.9 \pm 1.9}$	$\mathbf{30.9 \pm 2.2}$
	Realistic Vision-v3.0	LoRA	46.8 ± 4.0	48.5 ± 4.0	49.8 ± 4.2	32.7 ± 1.6	33.8 ± 1.4	34.0 ± 1.5
		FouRA	$\mathbf{50.4 \pm 3.0}$	$\mathbf{51.6 \pm 3.3}$	$\mathbf{52.2 \pm 3.5}$	$\mathbf{33.6 \pm 1.5}$	$\mathbf{34.1 \pm 1.2}$	$\mathbf{34.0 \pm 1.4}$

Table 2: Evaluation of LoRAs on Text-to-Image tasks. Adapters are rank 64. Results are averaged over 30 seeds.

Figure 7: **Multi-Adapter Fusion in LoRA v/s FouRA**. Sample images for style transfer on various prompts (e.g., bird, car, fox) for *Paintings*, *Bluefire*, *3D* and Merged adapters. Observe the highlighted merged images. FouRA does a much better job in preserving both styles, compared to LoRA.

thus both images look more realistic. These results demonstrate that FouRA images are both diverse (even at high adapter strengths) as well as aesthetically coherent. See App. E for more experiments.

Multi-Adapter: Fig. 7 shows images for style transfer merging for various prompts (e.g., bird, car, fox) for three styles: *Paintings*, *Blue-fire* and *3D*. We also provide the outputs of the linear combination of LoRA and FouRA for both these tasks. We see that merged LoRA images sometimes lose one of the concepts (e.g., the blue fire is lost for Panda and Dog) or have severe artifacts (e.g., the fox with multiple tails and the bird without a head). In comparison, FouRA images for merged adapters preserve the concepts and do not display any distortions. This property of FouRA is a direct consequence of our analysis in App. B.3 and is also evident from the HPSv2 reported in Tab. 1, where for higher adapter strengths, FouRA shows gains upto 3% over LoRA.

Adapter	α_b	α_p	HPSv2 score
LoRA	0.4	0.4	33.4
FouRA	0.4	0.4	**33.5**
LoRA	0.6	0.6	32.7
FouRA	0.6	0.6	**33.5**
LoRA	0.8	0.8	31.2
FouRA	0.8	0.8	**33.6**
LoRA	1.0	1.0	30.3
FouRA	1.0	1.0	**33.1**

Table 1: Merging two adapters for Blue Fire and Paintings with strengths α_b and α_p.

5.3 Text-to-Image Concept Editing

We establish the performance of our approach on nuanced editing tasks for specific target images by training FouRA using the disentangled objective proposed in concept sliders [9]. We train LoRA and FouRA modules using pairs of prompts describing the editing concepts. Fig. 8 shows results of editing the *Age* and *Hair* concepts. As observed, although the Age adapters are trained using a disentangled objective, LoRA changes the gender of the subject, and produces artifacts at high scales. FouRA is elegantly able to age them while retaining their original features. Similarly, the *Hair* FouRA produces a smoother representation. We provide quantitative evaluations in App. 5.3, and observe that at higher strengths, FouRA consistently outperforms LoRA in terms of the LPIPS score.

Composite Sliders: We qualitatively evaluate the composite *'hair'* and *'age'* adapter between LoRA and FouRA in Appendix 5.3. We show the results on two target prompts "A female Indian person" and

https://doi.org/10.52202/079017-2284

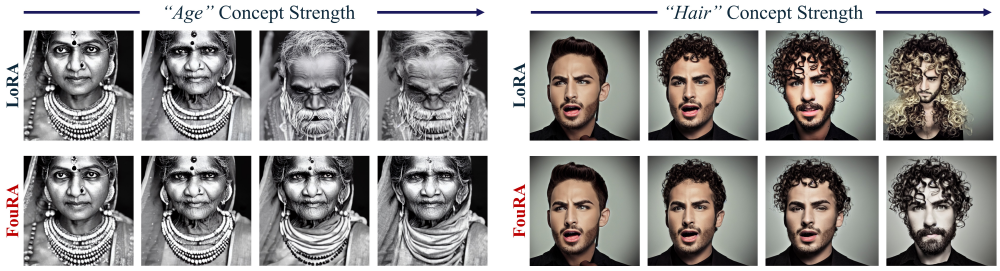


Figure 8: **LoRA v/s FouRA** . *Age* (Left) and *Hair* (right) concept slider examples where as the scale increases the effect of disentanglement in FouRA is more prominent. For larger scales the gender of the person changes in *Age* LoRA, and the structure of the face changes in *Hair* LoRA.

"A male white person" respectively. Overall, we observe that FouRA does a better job at compositing both sliders, as it produces a smooth transition between the concepts. In comparison, LoRA distorts the subjects faces at high adapter scales, and interferes with other facial features. We also show that the LPIPS diversity is much lower for generated images between different strength for FouRA F.4 at higher scales of the adapter.

5.4 Commonsense Reasoning Tasks

While our design choices for FouRA are primarily motivated for vision tasks, we evaluate its efficacy on eight commonsense reasoning tasks using the split from [18] in Tab. 3. We trained LoRA and FouRA adapters over a LLaMA3-8B [4] model. Our analysis shows that employing FouRA at rank 16 and 32 both outperform LoRA at the rank 32 setting.

Adapter	Rank	Trainable Params	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Average
LoRA	32	56.60 M	71.3	87.1	**79.9**	92.7	84.5	87.9	77.2	82.4	82.9
FouRA	16	28.31 M	74.4	**89.1**	79.8	94.9	**86.7**	90.2	80.1	85.2	85.1
FouRA	32	56.63 M	**74.8**	89.0	**79.9**	**95.3**	85.9	**90.9**	**80.8**	**85.6**	**85.3**

Table 3: **Performance on Commonsense Reasoning benchmarks:** Evaluation on eight Commonsense Reasoning benchmarks with the Llama-3(8B) model.

5.5 Computational Analysis

Table 4 provides the computational analysis for FouRA, as compared to LoRA. We provide the #parameters during inference along with the training time for FouRA. Along with this, we show the HPS-v2.1 scores on the *Blue Fire* validation set. Additionally, we provide the results for a FouRA variant with a fixed gating strategy during inference. FouRA layers with inference-adaptive masking produce an overhead of 0.02% more than LoRA, as compared to base model weights. However, FouRA with frozen masking can essentially reduce the computational overhead by a factor of 2, and still retain a higher performance than the base LoRA.

Adapter	Training Time	Epoch Time	GPU Memory	Inference Time	HPS (*Paintings*) (↑)
LoRA	1.87 sec/iter	22.0 sec	53.69 GB	14.9 step/sec	27.7
FouRA (Inference-Adaptive Mask)	2.09 sec/iter	24.5 sec	53.89 GB	11.1 step/sec	**30.6**
FouRA (Frozen Mask)	2.07 sec/iter	24.3 sec	53.81 GB	14.9 step/sec	30.3

Table 4: **Computational and Runtime Complexity.** The training measurements are performed on Tesla A-100 GPU with a batch-size of 8. The adapters are all rank=64, and HPS-v2 is computed at $\alpha = 0.8$.

5.6 Ablation Studies

Individual gain of every component

We show individual contributions from FouRA modules in Table 5. We fix rank=64 and α=0.8, and provide results on the paintings validation set. As evident from LPIPS-Diversity and HPS scores, the adaptive mask selection strategy performs better than the dynamic fixed mask selection strategy. For the case without frequency transform, Inference-Adaptive masking improves the HPS score from 28.2 to 28.7. When accompanied with Frequency transform, the HPS increases from 30.3 for frozen dynamic masking to 30.6 for inference-adaptive masking.

https://doi.org/10.52202/079017-2284

Adapter	Fourier	Frozen Dynamic Mask	Inf-Adaptive Mask	HPS (↑)	LPIPS-Diversity (↑)
LoRA				27.7	37.8
Frozen Mask		✓		28.2	38.9
Inference-Adaptive Mask			✓	28.7	39.7
FouRA (No Mask)	✓			30.0	43.2
FouRA (Frozen Mask)	✓	✓		30.3	44.0
FouRA (Inference-Adaptive Mask)	✓		✓	**30.6**	**44.5**

Table 5: **Individual gain with FouRA components.** Gains from each individual component of FouRA. All results are with rank 64 and $\alpha = 0.8$ on the paintings adapter.

Varying the Adaptive Rank Selection Strategy in Text-to-Image Stylized Generation

Fig. 9 shows the HPS-v2.1 curves obtained for evaluating LoRA, SoRA [3] and FouRA on the Paintings validation set for different adapter strength α. Additionally, we also show the performance of our inference-adaptive rank selection method directly on LoRA. All the numbers are for base rank=64 adapters. As observed, SoRA outperforms LoRA at higher ranks. However, our inference-adaptive rank selection strategy improves performance over SoRA, indicating that in vision models, varying the effective-rank across time steps of the diffusion process is ideal. FouRA outperforms all methods, indicating the benefits of training our proposed rank selection strategy in the frequency domain.

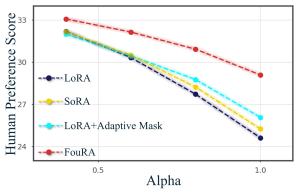


Figure 9: Comparison of different rank selection methods.

Varying the Rank in Text-to-Image Stylized Generation

In Fig. 10, we investigate the impact of FouRA over varying values of input rank, and compare with LoRA. We observe that rank is a highly sensitive parameter for LoRA. However, the HPS scores across ranks for FouRA are higher than the highest HPS score acheived at any rank by LoRA, highlighting the effect of gating in frequency domain. This helps FouRA to avoid underfitting as the rank reduces and overfitting as it increases. Furthermore, FouRA generates a diverse set of images across all ranks.

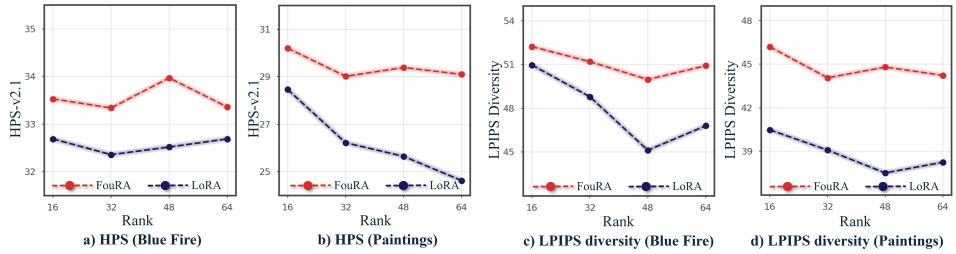


Figure 10: **HPS-v2.1 scores for each adapter across ranks.** FouRA continues to outperform LoRA as the rank increases for both Paintings and Blue Fire datasets.

6 Conclusion

In this paper, we proposed FouRA, a parameter efficient fine-tuning method within the frequency domain. Through extensive experiments and rigorous analysis, we showed that FouRA successfully solves the problems related to data copying and distribution collapse while significantly improving the generated image quality over LoRA. We also present an intensive study on the impact of compact representation of Low rank subspaces in transformed domain. Further, we showed that FouRA can leverage our proposed adaptive mask ranking approach and further push the generalization capabilities of PEFT models without under-fitting. Additionally, we demonstrated the efficacy of FouRA in merging two concepts, as the frequency domain acts as a decorrelated subspace for multiple adapters. Assessing the performance of FouRA, we feel encouraged to think that frequency domain fine-tuning of adapters will potentially be a popular research direction in the coming years.

https://doi.org/10.52202/079017-2284

References

[1] Françoise Beaufays and Bernard Widrow. Simple, alc, o rithms for fast adaptive filtering. 1993.

[2] Marc Peter Deisenroth, A Aldo Faisal, and Cheng Soon Ong. *Mathematics for machine learning*. Cambridge University Press, 2020.

[3] Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. Sparse low-rank adaptation of pre-trained language models. *arXiv preprint arXiv:2311.11696*, 2023.

[4] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[5] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[6] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.03206*, 2024.

[7] Zihao Fu, Haoran Yang, Anthony Man-Cho So, Wai Lam, Lidong Bing, and Nigel Collier. On the effectiveness of parameter-efficient fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12799–12807, 2023.

[8] Rohit Gandikota. Concept slider. `https://github.com/rohitgandikota/sliders/`, 2023.

[9] Rohit Gandikota, Joanna Materzynska, Tingrui Zhou, Antonio Torralba, and David Bau. Concept sliders: Lora adaptors for precise control in diffusion models. *arXiv preprint arXiv:2311.12092*, 2023.

[10] Ziqi Gao, Qichao Wang, Aochuan Chen, Zijing Liu, Bingzhe Wu, Liang Chen, and Jia Li. Parameter-efficient fine-tuning with discrete fourier transform. *arXiv preprint arXiv:2405.03003*, 2024.

[11] Arthita Ghosh and Rama Chellappa. Deep feature extraction in the dct domain. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 3536–3541, 2016.

[12] Yuchao Gu, Xintao Wang, Jay Zhangjie Wu, Yujun Shi, Yunpeng Chen, Zihan Fan, Wuyou Xiao, Rui Zhao, Shuning Chang, Weijia Wu, et al. Mix-of-show: Decentralized low-rank adaptation for multi-concept customization of diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

[13] René Haas, Inbar Huberman-Spiegelglas, Rotem Mulayoff, and Tomer Michaeli. Discovering interpretable directions in the semantic latent space of diffusion models. *arXiv preprint arXiv:2303.11073*, 2023.

[14] Ligong Han, Yinxiao Li, Han Zhang, Peyman Milanfar, Dimitris Metaxas, and Feng Yang. Svdiff: Compact parameter space for diffusion fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7323–7334, 2023.

[15] Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*, 2021.

[16] Xuanhua He, Keyu Yan, Rui Li, Chengjun Xie, Jie Zhang, and Man Zhou. Frequency-adaptive pan-sharpening with mixture of experts, 2024.

[17] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

 https://doi.org/10.52202/079017-2284

[18] Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2304.01933*, 2023.

[19] Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2304.01933*, 2023.

[20] Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering, 2019.

[21] Zahra Kadkhodaie, Florentin Guth, Eero P Simoncelli, and Stéphane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representation. *arXiv preprint arXiv:2310.02557*, 2023.

[22] Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with lora. *arXiv preprint arXiv:2312.03732*, 2023.

[23] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1931–1941, 2023.

[24] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[25] Yang Lin, Xinyu Ma, Xu Chu, Yujie Jin, Zhibang Yang, Yasha Wang, and Hong Mei. Lora dropout as a sparsity regularizer for overfitting control. *arXiv preprint arXiv:2404.09610*, 2024.

[26] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*, 2024.

[27] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`, 2022.

[28] Tan Nguyen, Minh Pham, Tam Nguyen, Khai Nguyen, Stanley J Osher, and Nhat Ho. Transformer with fourier integral attentions. *arXiv preprint arXiv:2206.00206*, 2022.

[29] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, pages 16784–16804. PMLR, 2022.

[30] Pablo Pernias, Dominic Rampas, Mats Leon Richter, Christopher Pal, and Marc Aubreville. Würstchen: An efficient architecture for large-scale text-to-image diffusion models. In *The Twelfth International Conference on Learning Representations*, 2023.

[31] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[32] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[34] Simo Ryu. Low-rank adaptation for fast text-to-image diffusion fine-tuning, 2021.

[35] Levent Sagun, Leon Bottou, and Yann LeCun. Eigenvalues of the hessian in deep learning: Singularity and beyond. *arXiv preprint arXiv:1611.07476*, 2016.

[36] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

[37] Viraj Shah, Nataniel Ruiz, Forrester Cole, Erika Lu, Svetlana Lazebnik, Yuanzhen Li, and Varun Jampani. Ziplora: Any subject in any style by effectively merging loras. *arXiv preprint arXiv:2311.13600*, 2023.

[38] Chenyang Si, Ziqi Huang, Yuming Jiang, and Ziwei Liu. Freeu: Free lunch in diffusion u-net. *arXiv preprint arXiv:2309.11497*, 2023.

[39] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in stable diffusion. 2023.

[40] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. *Advances in Neural Information Processing Systems*, 36:47783–47803, 2023.

[41] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[42] Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. *arXiv preprint arXiv:2310.14152*, 2023.

[43] Qiucheng Wu, Yujian Liu, Handong Zhao, Ajinkya Kale, Trung Bui, Tong Yu, Zhe Lin, Yang Zhang, and Shiyu Chang. Uncovering the disentanglement capability in text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1900–1910, 2023.

[44] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

[45] Xun Wu, Shaohan Huang, and Furu Wei. Mole: Mixture of lora experts. In *The Twelfth International Conference on Learning Representations*, 2023.

[46] Xilie Xu, Jingfeng Zhang, and Mohan Kankanhalli. Autolora: A parameter-free automated robust fine-tuning framework. *arXiv preprint arXiv:2310.01818*, 2023.

[47] Jun Zhang, Yixin Liao, Xinshan Zhu, Hongquan Wang, and Jie Ding. A deep learning approach in the discrete cosine transform domain to median filtering forensics. *IEEE Signal Processing Letters*, 27:276–280, 2020.

[48] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2023.

[49] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[50] Ming Zhong, Yelong Shen, Shuohang Wang, Yadong Lu, Yizhu Jiao, Siru Ouyang, Donghan Yu, Jiawei Han, and Weizhu Chen. Multi-lora composition for image generation. *arXiv preprint arXiv:2402.16843*, 2024.

 https://doi.org/10.52202/079017-2284

Appendices

A Contents

As part of the supplementary materials for this paper, we share our Implementation details, show extended qualitative and quantitative results and provide additional theoretical analysis for our proposed approach. The supplementary materials contain:

- Extended Theoretical Analysis
  - Proof of Singular Value Decomposition Analysis Lemma 4.1
  - Proof of Sparsity Lemma 4.2
  - Subspace Analysis
  - Merging of Adapters
  - Learning disentangled representations
- Implementation details and hyperparameters for all experiments
  - Datasets
  - Hyperparameters
- Interpretations for learnt metrics (HPS-v2.1 and LPIPS diversity)
- Additional experiments for text-to-image stylization.
  - Performance on Unseen Concepts for Text-to-Image Stylization
  - Effect of varying the frequency transform
  - Comparisons: 2D FFT on the tokens vs 1D FFT on token embeddings
  - Plots for quantiative metrics in Text-to-Image Stylization
  - Effect on data-copying artifacts after early stopping LoRA training
  - Additional Computational Analysis
  - Additional Visual Results on Text-to-Image Stylization
- Additional Experiments for Text-to-Image Editing using Concept Sliders
- Societal Impacts

B Theoretical Analysis

B.1 Proof for Lemma 4.1

In this section, we provide the proof for Lemma 4.1 of the main text.

Lemma 4.1. *Considering two adapters $\Delta\mathbf{W_1}$ and $\Delta\mathbf{W_2}$ and their corresponding sets of singular values $\{\sigma_{1,i}\}$ and $\{\sigma_{2,i}\}$. The adapter $\Delta\mathbf{W_1}$, will admit r rank approximation with lower error than $\Delta\mathbf{W_2}$ if $\sigma_{1,i} < \sigma_{2,i}$ for all $i \geq r$.*

Proof. Let $\mathbf{D_{1,r}}$ and $\mathbf{D_{2,r}}$ be diagonal matrices corresponding a rank r approximation of $\Delta\mathbf{W_1}$ and $\Delta\mathbf{W_2}$ respectively. The reconstruction errors $\mathbf{E_{1,r}}$ and $\mathbf{E_{2,r}}$ for these approximations are computes as follows:

$$\mathbf{E_{1,r}} = \Delta\mathbf{W_1} - LR_r(\Delta\mathbf{W_1}) = \mathbf{U_1 D_1 V_1^T} - \mathbf{U_1 D_{1,r} V_1^T} \tag{7}$$

$$\mathbf{E_{2,r}} = \Delta\mathbf{W_2} - LR_r(\Delta\mathbf{W_2}) = \mathbf{U_2 D_2 V_2^T} - \mathbf{U_2 D_{2,r} V_2^T} \tag{8}$$

A matrix $\Delta\mathbf{W}$ can be written as the sum of it's right and left 1-D singular vectors $\mathbf{u}$ and $\mathbf{v}$ as follows:

$$\Delta\mathbf{W} = \mathbf{U D V^T} = \sum_{i=1}^{min(k_1,k_2)} \sigma_i \mathbf{u v^T} \tag{9}$$

Hence, we rewrite the reconstruction errors $\mathbf{E_{1,r}}$ and $\mathbf{E_{2,r}}$ as a sum of the product of their 1-D singular vectors.

$$\mathbf{E_1} = \sum_{i=1}^{min(k_1,k_2)} \sigma_{1,i} \mathbf{u_1 v_1^T} - \sum_{i=1}^{r} \sigma_{1,i} \mathbf{u_1 v_1^T} = \sum_{i=r+1}^{min(k_1,k_2)} \sigma_{1,i} \mathbf{u_1 v_1^T} \tag{10}$$

$$\therefore \mathbf{E_2} = \sum_{i=r+1}^{min(k_1,k_2)} \sigma_{2,i} \mathbf{u_2 v_2^T} \tag{11}$$

Following the Eckart-Young theorem [5] and theorem 4.95 in Mathematics for Machine Learning [2], the value of the norm of reconstruction error is given as:

$$\|\mathbf{E_1}\| = \left\| \sum_{i=r+1}^{min(k_1,k_2)} \sigma_{1,i} \mathbf{u_1 v_1^T} \right\| = \sigma_{1,r+1} \tag{12}$$

Hence the difference of reconstruction errors is computed as follows:

$$\|\mathbf{E_{2,r}}\| - \|\mathbf{E_{1,r}}\| = \sigma_{2,r+1} - \sigma_{1,r+1} \tag{13}$$

We know $\sigma_{2,r+1} > \sigma_{1,r+1}$. Hence, we prove that $\|\mathbf{E_{2,r}}\| > \|\mathbf{E_{1,r}}\|$.

□

Here it is important to note an adapter with lesser eigenvalue spread there will exist an r rank approximation such it has a lower approximation error than adapter with wider eigenvalue spread. However, the rank r should follow in lemma above. Further, it is important note the low rank adapter with a lower approximation error would estimate the noise closer to optimal estimate and will converge to de-noised image with improved perception scores.

B.2 Proof for Lemma 4.2

In this section, we provide a proof for Lemma 4.2 and Corollary 4.2.1 of the main text.

Lemma 4.2. *Consider an adapter $\Delta\mathbf{W}$ with a rank higher than the required rank to fit a training data distribution. The upper-bound of generalization error $\mathcal{R}$ for fine-tuning this adapter reduces as the effective rank of the adapter reduces. After reducing to a certain value of effective rank, the upper-bound of generalization error will increase as rank reduces further.*

Corollary 4.2.1. *Additionally, the generalization bound is more stable when the singular value distribution of adapter weights $\Delta\mathbf{W}$ is more compact.*

Proof. Consider $\mathcal{A}$ as a learning algorithm for finetuning our adaptation weights $\Delta\mathbf{W}$, and $\mathbf{S}$ is our training set of length n. Additionally, consider the ratio of effective rank to original rank as p (where $1-p$ is a sparsity parameter). The LoRA Generalization error upper-bound for $\mathcal{A}$ can be computed from Pointwise Hypothesis Stability equations (Theorem 2 of [7]). We have for a constant C with a probability $1-\delta$,

https://doi.org/10.52202/079017-2284

$$\mathcal{R}(\mathcal{A}, S) < \hat{\mathcal{R}}(\mathcal{A}) + \sqrt{\frac{C^2 + \frac{24C\rho^2}{\lambda_{min}+2(1-p)}}{2n\delta}} \tag{14}$$

Here, $\hat{\mathcal{R}}(\mathcal{A}, S)$ represents the emperical error, and λ_{min} represents the minimum eign-value of the loss Hermitian matrix. For finetuning tasks, $\lambda_{min} \approx 0$ for a loss Hermitian matrix which is well behaved as the model has been trained, as observed by [35].

Based on the observations of [25, 7], and the above equation, we can observe that the generalization error reduces as the sparsity increases when the effective rank ratio p is low, and sparsity $(1 - p)$ is relatively high.

As effective rank increases and sparsity$(1 - p)$ reduces, if the length of data distribution is small, there is a high risk of overfitting.

However, as effective rank reduces and sparsity increases, there will come a point when the number of trainable parameters are much lower than what is required for representing the training data distribution, leading to underfitting. Hence, there exists an **optimal** effective rank, proving Lemma 4.2.

The optimal effective rank is driven by the generalization error. For highly sparse representations, the empirical error $\hat{\mathcal{R}}(\mathcal{A}, S)$ dominates over the second term, as it increases significantly.

From Lemma 4.1, we know that if the singular value spread of $LR_r(\Delta\mathbf{W})$ contains a more compact representation, the reconstruction error from the r-rank subspace is reduced. Hence, the training objective $\hat{\mathcal{R}}(\mathcal{A}, S)$ reduces.

A consequence of this reduction in error signifies that the weights can potentially achieve higher generalization capability by even further sparsification, before $\hat{\mathcal{R}}(\mathcal{A}, S)$ starts dominating the generalization error bound.

Hence, model weights which can be represented in compact singular value representations can achieve a lower generalization error by further increasing sparsity, proving Corollary 4.2.1.

□

B.3 Subspace analysis

In Section 5, we demonstrate that the fine tuned FouRA adapter performs significantly better than LoRA. In this Section, we attempt to analyze the performance of adapters in terms of the correlation of the subspaces of the base model and that of the adapter. The analysis follows the approach discussed in [17]. We project the base model weights $\mathbf{W_0}$ onto the r-dimensional subspace of our finetuned adapters $\Delta\mathbf{W}$. The projection of base matrix $\mathbf{W_0}$ on to the subspace of the adapter is $\mathbf{U}^T\mathbf{W_0}\mathbf{V}^T$, where $\mathbf{U}/\mathbf{V}$ are the left and right top-r singular vectors of $\Delta\mathbf{W}$. As defined in [17], $\frac{\|\Delta\mathbf{W}\|_F}{\|\mathbf{U}^T\mathbf{W_0}\mathbf{V}^T\|_F}$ is the amplification factor, a measure of the subspaces emphasised in the adapter $\Delta\mathbf{W}$ when compared with base weights $\mathbf{W_0}$. Between two adapters of the same rank, a higher amplification factor effectively corresponds to the amount of information learned by the adapter, which is orthogonal to the model weights. In table B.1, we analyze the amplification factors of FouRA and LoRA at rank=32. This is an average over all the adaptors of finetuned UNet model. Observe that FouRA Amplifies the learnt subspaces by factor >2x as compared to LoRA. Hence, FouRA weights are more de-correlated from the pretrained base model weights. Additionally, higher emphasis on the set of learnt subsapces enables the learning of new tasks without catastrophic forgetting. Figure B.1 shows further analysis of learnt subspaces over multiple ranks.

	$\|\| \Delta w \|\|_F$	$\|\| U^T W V^T \|\|_F$ (↓)	$\frac{\|\|\Delta w\|\|_F}{\|\|U^T W V^T\|\|_F}$ (↑)
LoRA	1.07	0.95	1.2
FouRA	0.32	0.81	2.8

Table B.1: **Amplification Factor Analysis.** Average amplification factor components over all layers of the diffusion UNet with Rank=32 LoRA and FouRA.

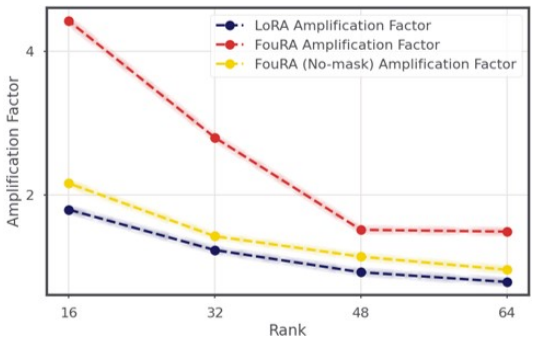


Figure B.1: **Amplification Factor of FouRA v/s LoRA:** As the computed Amplification Factor referred to in B.3 is higher in case of FouRA, we justify the learnt representations are more de-correlated from the base weights.

B.3.1 Merging adapters

Recent works [37] demonstrate joint-adapter training for effectively merging multiple low-rank adapters. In Section 5, we demonstrate the ability of the FouRA module to merge multiple adaptors in a way which retains both their capabilities with high fidelity.

Proposition 1. *Considering two adapters* $\Delta\mathbf{W_1}$ *and* $\Delta\mathbf{W_2}$*. The linear combination of both these adaptors tends to generate results which retain the capabilities of both the adapters, if the norm of the projection of* $\Delta\mathbf{W_1}$ *on the subspace of* $\Delta\mathbf{W_2}$*, computed as* $\|\mathbf{U_2}^T\Delta\mathbf{W_1}\mathbf{V_2}^T\|$ *is lower. Here,* $\mathbf{U_2}/\mathbf{V_2}$ *are the singular vectors of* $\Delta\mathbf{W_2}$*.*

We provide analysis in Table B.2 complementing Proposition 1, and demonstrating how FouRA has a greater tendency to disentangle two adapters, making it highly effective for multi-adaptor fusion without joint training. We computed the Norm of the projections FouRA adapter weights trained on one subtask, onto the weights trained on another subtask, and compared it to LoRA projection norms. We analyzed the correlation between weights of three tasks: *BlueFire*, *Paintings* and *3D*. As observed from the numbers, FouRA projection norms are much lower, suggesting a higher number of orthogonal subspaces for FouRA projections. This aligns with Table 1 and Figure 7 of the main text, where we observe that FouRA is successfully able to retain the capabilities of both adapters after the merge.

Dataset 1	Dataset 2	LoRA Projection Norm(↓)	FouRA Projection Norm (↓)
BlueFire	*Paintings*	0.40	**0.25**
BlueFire	*3D*	0.39	**0.27**
3D	*Paintings*	0.47	**0.32**

Table B.2: Norm of projection of adapter weights trained on task 1, over adapter weights trained on task 2, calculated as $\|\mathbf{U_2}^T\Delta\mathbf{W_1}\mathbf{V_2}^T\|$. Observe that FouRA has a lower Projection Norm,

B.4 Learning disentangled representations

Given $z_{in}, z_{out} \in \mathcal{R}^{d\times k_1}$ from (5), and let the input have three attributes that can be represented as $z_{in} = [z_{race}, z_{age}, z_{gender}]$, the autocorrelation matrix at the output of FouRA layer can be written as

$$\begin{aligned}\mathbf{R}_{d\times d} = \mathbf{z}_{out}\mathbf{z}_{out}^T = \mathbf{z_{in}}(\mathbf{W_0}+\Delta W)(\mathbf{W_0}+\Delta W)^T\mathbf{z_{in}^T} \\ = \mathbf{z_{in}}\mathbf{W_0}\mathbf{W_0}^T\mathbf{z_{in}^T} + \mathbf{z_{in}}\Delta W\Delta W^T\mathbf{z_{in}^T} + \mathcal{F}(\mathbf{W_0}\Delta W^T, \mathbf{z_{in}})\end{aligned} \tag{15}$$

From B.1, we established that the overlap of subspaces between low rank in transform domain ΔW and base matrix $\mathbf{W}$ is smaller at lower rank. In addition, in frequency domain, the term in the middle (in blue) computes the autocorrelation between the subspaces. From [1], this term is almost diagonal making the dot product $< z_{out}^{race}, z_{out}^{gender} > \approx 0$ or $< z_{out}^{race}, z_{out}^{age} > \approx 0$. Thus the weights for each attribute is poised to be learned independently. To verify this, In the experiments section, we motivate the idea of using foura to edit concepts while preserving the attributes of an image using concept sliders [9]

https://doi.org/10.52202/079017-2284

C Implementation Details

C.1 Datasets

C.1.1 Commonsense Reasoning

We use the commonsense reasoning datasets which comprise of 8 sub-tasks, each with a predefined training and testing set as shown in table C.1. We follow the setting of [19] for training. The common sense reasoning training dataset is a combination of the training datasets provided by [20], while we evaluate each evaluation dataset separately.

Dataset	#Train	#Val	Test
PiQA	16K	2K	3K
BoolQ	9.4K	2.4K	2.4K
SIQA	33.4K	1.9K	1.9K
OBQA	4.9K	0.5K	0.5K
Winogrande	9.2K	1.3K	1.8K
HellaSwag	39.9K	10K	10K
Arc_easy	2.25K	570	2.36K
Arc_challenge	1.12K	299	1.12K

Table C.1: Commonsense Benchmark

C.1.2 GLEU

We have performed the LLM study on six of the GLUE benchmarks - CoLA, SST-2, MRPC, STS-B, MNLI, and QNLI. GLEU benchamrk has been widely used for natural language understanding. All the dataset and task described in the Table C.2 is being utilized from Huggingface Datasets and each task has its own respective evaluation metric. We have described the train and test split of each of the task along with the respective evaluation metric in Table C.2.

Dataset	#Train	#Val	Metric
CoLA	8.5K	1043	Mcc
SST-2	67K	872	Acc
MRPC	3.7K	408	Acc
STS-B	5.7K	1.5K	Corr
MNLI	393K	9.8K	Acc(m/mm)
QNLI	105K	5.5K	Acc

Table C.2: GLUE Benchmark

C.1.3 Style Transfer Datasets

In this section, we provide more details on the four style transfer datasets we use for vision adaptation experiments. We followed the licensing terms for every dataset which was curated.

BlueFire (Training): The *BlueFire* dataset is created by collecting images from open public domain and consist of 6 concepts - car, dragon, bird, fox, man and castle. The dataset has a total of 54 images covering all the concepts.

BlueFire (Validation): The *Bluefire* validation set consists of 30 curated text prompts, of which 9 prompts contain one of 6 categories on which the model was trained, and the remaining 21 prompts correspond to categories which the low-rank adapter has not been fine-tuned on. These contain categories such as: (football, monster, sword, chess rook, lion, tiger, dog, cat, koala, panda).

For all training experiments validating on this dataset, we produce 30 images per prompt, varying the input seed. Hence, the HPS analysis is over 900 image and LPIPS-diversity analysis is over 14500 image pairs.

Paintings: On similar lines, the *Paintings* dataset is also a collection of images from public domain (CC0 license). The dataset has a total of 90 images cover 9 concepts - fire, bird, elephants, ship, horse, flower, woman, man and tiger.

Paintings (Validation): The *Paintings* validation set consists of 21 curated text prompts, of which 9 prompts contain one of 9 categories on which the model was trained, and the remaining 12 prompts

correspond to categories which the low-rank adapter has not been fine-tuned on. These contain categories such as: (lion, tiger, dog, cat, koala, panda, and other landscapes)

Paintings merged with BlueFire (Validation): The evauation set for merging *Paintings* and *Bluefire* consists of 18 curated text prompts. These contain categories such as: (fox, bird, lion, tiger, dog, cat, koala, panda, and other landscapes)

For all training experiments validating on this dataset, we produce 30 images per prompt, varying the input seed. Hence, the HPS analysis is over 440 image and LPIPS-diversity analysis is over 8750 image pairs.

Origami: The *Origami* dataset is also a collection of origami images from public domains. The dataset has a total of 52 images covering 7 concepts - bird, boat, flower, cat, dog, fox and house.

3D: The *3D* dataset is also a collection of images from public domains. These images are animated images showing 3D concepts. The dataset has a total of 30 images covering 6 concepts - boy, girl, astronaut, cat, dog, elephant, dog and building.

Concept Sliders: For concept sliders, we train and evaluate on three different concepts as shown in Table C.3. The evaluation set for each concept consists of 400 examples, over 10 seeds, essentially validating over 4000 images per concept. We follow the method in [8]

Concept	Positive prompt	Negative prompt	# Training Attributes	# Val. Attributes
Age	very old, wrinkly, gray hair, aged skin	very young, smooth skin, youthful	20	400
Surprise	looking surprised, wide eyes, open mouth	looking calm, neutral expression	20	400
Hair	curly hair, wavy hair	straight hair	20	400

Table C.3: Dataset statistics for Concept Slider Experiments

C.2 Hyper-parameters and Implementation details for all experiments

Text-to-image style transfer

We used the kohya-ss[4] repository for finetuning models for the text-to-image stylization task. For the masking we follow the approach for soft gating in [5]. For each task, we trained both LoRA and FouRA adapters with the same set of hyperparameters. We trained using 4 NVIDIA A100 GPUs, for 100 epochs at at batch size of 8. Our initial learning rate was set to $1e^{-4}$ for UNet and $5e^{-5}$ for the text encoder. LoRA and FouRA modules are applied in the default places for stable-diffusion-v1.5 backbone, same as in HuggingFace Diffusers. We trained using two sets of weights, the base sd-1.5[6] from runwayML, and RealisticVision3.0[7]. For some ablation studies, we varied the rank between 16, 32, 48, 64. In all the remaining experiments, we set the rank at 64 unless stated otherwise. Additionally, we set the Realistic Vision weights as our default for all experiments.

For quantitative evaluation, we observed the HPS-v2.1 and LPIPS-Diversity metrics at a range of values between $[0, 1]$ for adapter strength α. In all quantitative evaluations, we averaged over the same set of 30 seeds $\{0, 1, 2, ...29\}$.

Image editing using Concept Sliders

Single slider: The training data used in these experiments were curated from [9] . We used the repository [8] for finetuning the adapters. We train across 20 different attributes spanning different genders and races and other person attributes for each concept. The learning rate and other hyper-parameters are re-used from the repository. For all the experiments we fix a rank of 8 and with 50 denoising steps. For evaluations, we tested across 400 different examples for 10 seeds on each prompt including unseen categories such as 'doctor' , 'barista', 'cowboy' . For qualitative analysis, we compare across strengths $\in [-6, 6]$). We also evaluated the inference across different 3 different edit times [750, 800, 850].

[4]https://github.com/kohya-ss/sd-scripts

[5]https://github.com/prachigarg23/Memorisation-and-Generalisation-in-Deep-CNNs-Using-Soft-Gating-Mechanisms

[6]https://huggingface.co/runwayml/stable-diffusion-v1-5

[7]https://huggingface.co/spaces/Thafx/sdrv30

[8]https://github.com/rohitgandikota/sliders

https://doi.org/10.52202/079017-2284

Composite slider: For compositing we use similar setup as in the single slider. We compose the score functions using additive guidance. Specifically we weight each score function based on the relative strengths of the adapter during inference.

GLUE benchmark experiments We trained the LoRA and SoRA [3] baselines on the GLUE benchmark using the code and default set of hyper-parameters provided by the authors[9]. For training FouRA, we used the same set of hyper-parameters as the LoRA baseline. These are provided in this issue in their repository. For all the experiments, we trained using 1 NVIDIA A100 GPU.

For each task, and each baseline, we evaluated on all the samples of the validation set, the size of which is mentioned in Appendix C.2. This is slightly different from the evaluation in [3], as the authors originally ran inference only on a subset of the validation set, indicated here. Additionally, we used the set of three seeds $\{100, 81, 20\}$, chosen at random, to run all experiments.

D Interpretations for Metrics

In the main text, we used two metrics to validate style transfer on text-to-image diffusion models. Both are learnt metrics, i.e. HPS-v2.1 [44] and LPIPS-Diversity [49]. In this section, we provide reference ranges for both metrics, and how they can be interpreted.

D.1 LPIPS Diversity

We compute the LPIPS diversity δ_{lpips} of a dataset of n images as the average of the LPIPS pairwise distance between ${}^nC_2$ image pairs. In Figure D.1, we provide reference ranges for LPIPS distance between pairs of images. Notice the images in D.1a. are very similer. Hence, they generate a low LPIPS score (0.35). Hence in Table 2, we observe for high values of α, as the average LPIPS scores reflect that LoRA produces close to identical images in many case, but FouRA successfully gets rid of this data copying problem. Figures D.1b. and c. are lesser correlated from each other and hence produce a higher distance. Figures D.1d.-f. and g.-i. similarly vary from one another with in ascending order of LPIPS diversity scores, which is reflected in the image (The pose of the fox and variations in the fire in car images). The scores in Table 2 reflect a gain of 2-6 points in LPIPS diversity between LoRA and FouRA. These are significant improvements in the diversity of generated samples as observed from Figure D.1.

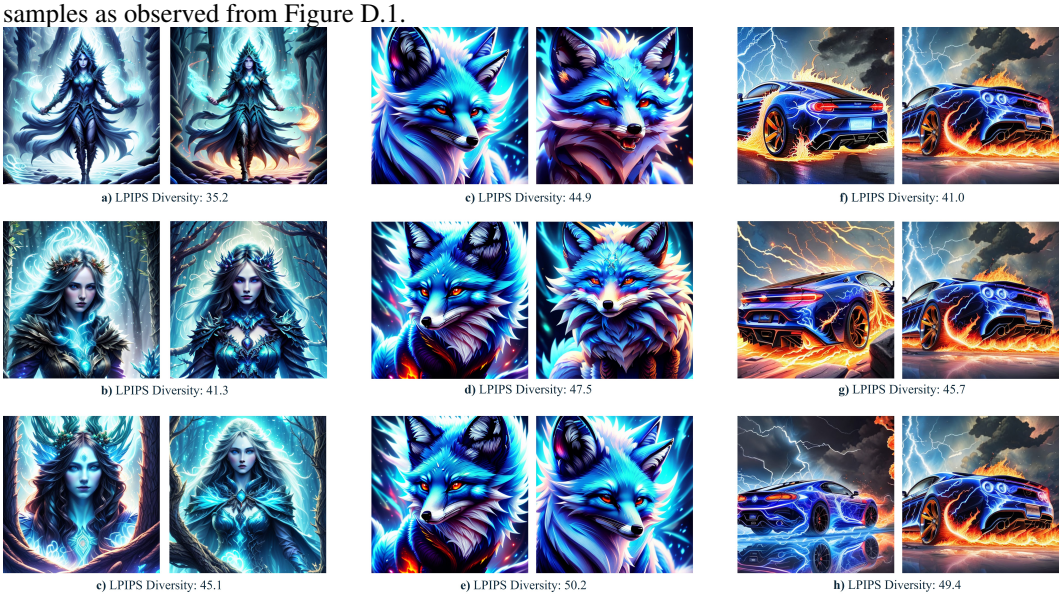


Figure D.1: **Interpretation of the LPIPS Diversity metric**. This figure illustrates the interpretation of LPIPS Diversity, which we used to detect mode collapse. Images which look similar (i.e. sharing the same pose or similar characteristics) tend to generate a lower LPIPS distance.

[9]https://github.com/TsinghuaC3I/SoRA

D.2 Human Preference Scores

For computing Human Preference Score, we utilized to the v2.1 HPS model provided by the authors [44]. Please refer to Figure D.2 for reference HPS-v2.1 values. **Please note that in the Figure D.2 the "prompt" corresponds to the input prompt to HPS model, and may or may not be the prompt used to generate the image.**

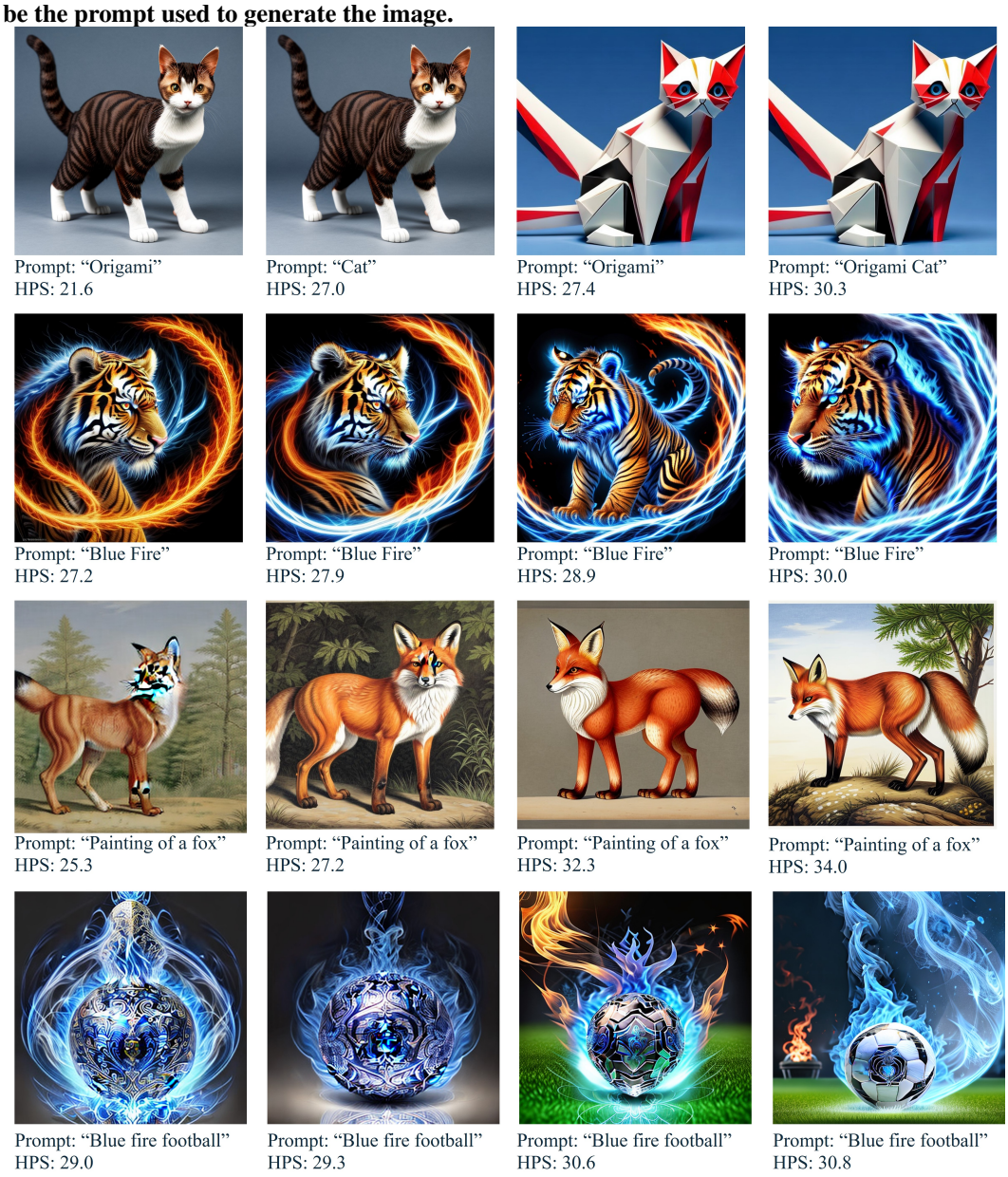


Figure D.2: **Interpretation of the HPS-v2.1 metric**. This figure illustrates the interpretation of HPS scores, which we used to track three key aspects of generated images: 1.Alignment with the prompt, 2.Alignment with the adapter style and 3.Aesthetic quality. Observe that the HPS-v2.1 metric is able to effectively quantify these key aspects of generated images. **The "Prompt" in this figure corresponds to the input prompt to HPS model for text and image alignment, and may or may not be the prompt used to generate the image**

We used HPS as a metric to track a combination of three key aspects of generated images. **Alignment with the Prompt:** Observe the first row in Figure D.2. For the wrong prompt (e.g. "Origami" for a cat image), the model produces a low HPS score (21.6). However, this score increases as the prompt and image alignment improves.

https://doi.org/10.52202/079017-2284

Strength of the adapter: Observe the second row in Figure D.2. The prompt we fed into HPS is the name of the adapter(*blue fire*). Notice how the HPS values increase for increase in the adapter strength.

Image Quality: Observe the third row in Figure D.2. HPS scores can successfully differentiate between images with high and low aesthetic quality.

Thus the, HPS provides us with a quantifiable metric for all the three aspects over we wish to evaluate our finetuned adapters. Moreover, the fourth row in Figure D.2 shows how the HPS can effectively track all these three aspects at once. Hence, the prompt we feed to the HPS model to evaluate an image is a combination of the name of the adapter and the prompt used for generating the image. E.g. the prompt used to evaluate image generated by "dog in space" with the adapter *BlueFire*, is "blue fire dog in space."

This method also works well for evaluating the merging of two adapters. We simply add both the adapter names in the prompts while evaluating their HPS scores.

E Additional Experiments on Text-to-Image stylization

E.1 Additional Ablation Studies

E.1.1 Performance on Unseen Concepts for Text-to-Image Stylization

Section C.1.3 details the distribution of both our validation sets, *Bluefire* and *Paintings*. We split the validation set in seen and unseen concepts during training of the adapter. *Bluefire* contains 21 unseen categories (630 generated images), and *Paintings* contains 12 unseen categories (360 generated images). From Table E.1, we can observe that FouRA has a better generalization capability on unseen classes, as compared to LoRA. This result supplements our Proof for Corollary 4.2.1, essentially confirming that FouRA is able to reduce the upper bound of generalization error.

Adapter	Dataset	HPSv2 score(↑) $\alpha = 1.0$	$\alpha = 0.8$	$\alpha = 0.6$
LoRA	*Paintings (Unseen)*	24.1	27.0	29.7
FouRA	*Paintings (Unseen)*	**28.5**	**30.4**	**31.7**
LoRA	*Bluefire (Unseen)*	32.5	33.6	33.8
FouRA	*Bluefire (Unseen)*	**33.2**	**34.4**	**34.4**

Table E.1: **Performance on unseen classes**. Shows that on unseen classes FouRA generalizes better on unseen categories.

E.1.2 Effect of varying the frequency transform

Finally, we evaluate the effect of changing the frequency transform between DFT and DCT for our proposed FouRA (see Table E.2). First, we observe that both DFT- and DCT-based FouRA models significantly outperform LoRA. Also, both DFT and DCT achieve comparable scores in terms of HPSv2 which means our approach is robust to the type of frequency transforms being used.

Transform	LPIPS Diversity(↑) $\alpha = 1.0$	$\alpha = 0.8$	$\alpha = 0.6$	HPSv2 score(↑) $\alpha = 1.0$	$\alpha = 0.8$	$\alpha = 0.6$
LoRA	38.3	37.8	39.1	24.6	27.7	30.3
FouRA DFT	44.2	44.7	44.8	**29.1**	**30.9**	**32.2**
FouRA DCT	**46.7**	**45.5**	**45.0**	28.9	30.6	31.9

Table E.2: **Effect of varying the frequency transform in FouRA**

E.1.3 Comparisons: 2D FFT on the tokens vs 1D FFT on token embeddings

As illustrated in Fig. E.1, we proposed two variants of our approach: (1) FouRA$_{emb}$ that computes the frequency transform across the embedding dimension, and (2) FouRA$_{token}$ that computes the frequency transform along the token dimension.

Table E.3, we compare FFT applied on token embeddings with LoRA. We hypothesize that transform done this way might capture variations in local patches of the image. Further as LoRA on vision adaptors generally apply rank reduction in the embedding dimension, applying the same in fourier dimension translates to spectral filtering in the embedding space. For the sake of completeness, we also run experiments to apply transform in the 2D token space, we call this $FouRA_{token}$. In

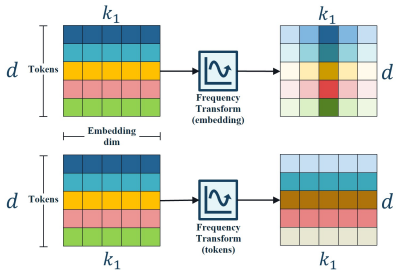


Figure E.1: **Two directions of the proposed Frequency Transform**. FouRA$_{emb}$ computes the frequency transform along the embedding dimension (top), whereas FouRA$_{token}$ computes the frequency transform across all the tokens (bottom).

Table E.3, we empirically observe that FouRA$_{emb}$ performs better than FouRA$_{token}$. Hence, unless stated otherwise, we set FouRA$_{emb}$ as the default variant of FouRA for our experiments.

Style	Base Model	Adapter	LPIPS Diversity(↑) $\alpha = 1$	$\alpha = 0.8$	$\alpha = 0.6$	HPSv2 score(↑) $\alpha = 1$	$\alpha = 0.8$	$\alpha = 0.6$
Painting	RealisticVision	LoRA	38.3 ± 3.5	37.8 ± 3.6	39.2 ± 3.7	24.6 ± 1.8	27.7 ± 1.8	30.3 ± 1.7
		FouRA$_{token}$	$\mathbf{44.2 \pm 3.7}$	44.5 ± 4.0	44.6 ± 3.9	28.4 ± 1.8	30.6 ± 1.5	32.0 ± 1.4
		FouRA$_{emb}$	$\mathbf{44.2 \pm 3.8}$	$\mathbf{44.7 \pm 3.9}$	$\mathbf{44.8 \pm 3.9}$	$\mathbf{29.1 \pm 1.9}$	$\mathbf{30.9 \pm 1.6}$	$\mathbf{32.2 \pm 1.5}$
Blue Fire	RealisticVision	LoRA	46.8 ± 4.0	48.5 ± 4.0	49.8 ± 4.2	32.7 ± 1.6	33.8 ± 1.4	34.0 ± 1.5
		FouRA$_{token}$	50.4 ± 3.0	51.6 ± 3.3	52.2 ± 3.5	$\mathbf{33.6 \pm 1.5}$	34.1 ± 1.2	34.0 ± 1.4
		FouRA$_{emb}$	$\mathbf{50.9 \pm 3.1}$	$\mathbf{52.3 \pm 3.2}$	$\mathbf{53.3 \pm 3.8}$	33.4 ± 1.7	$\mathbf{34.6 \pm 1.3}$	$\mathbf{34.5 \pm 1.2}$

Table E.3: FouRA$_{emb}$ vs FouRA$_{token}$ vs LoRA

E.2 Plots for quantiative metrics in Text-to-Image Stylization

In Fig. E.2, we provide HPS and LPIPS-diversity scores at ranks $\{16, 32, 48, 64\}$ and adapter strengths $\alpha = \{0.2, 0.4, 0.6, 0.8, 1.0\}$ for LoRA and FouRA. These plots are using the base weights of Realistic Vision-3.0. These scores are an extenstion to Table 2 of the main text. Observe FouRA outperforms LoRA on both metrics, at all ranks.

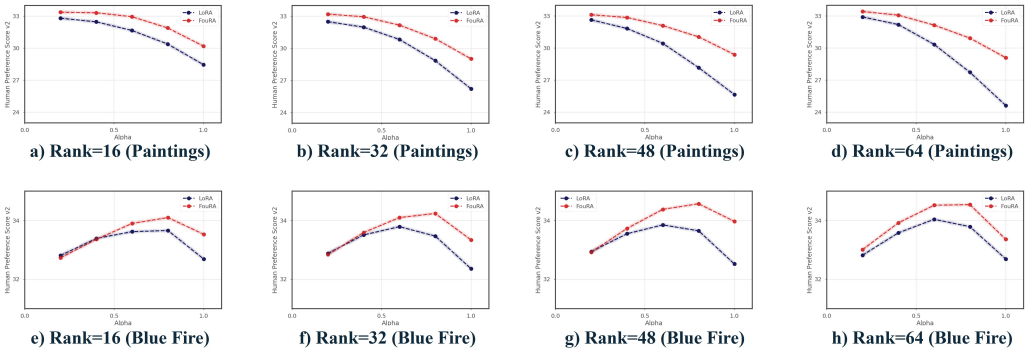


Figure E.2: **Quantitative Evaluations for LoRA v/s FouRA on text-to-image stylization**. We provide plots at ranks $\{16, 32, 48, 64\}$ and adapter strengths $\alpha = \{0.2, 0.4, 0.6, 0.8, 1.0\}$

E.3 Effect on data-copying artifacts after early stopping LoRA training

We study the data-copying(distribution collapse) phenomenon in more detail in Figure E.3. We tracked the LPIPS-diversity as a measure of data-copying and HPS-v2 scores as a measure of adapter quality. We do notice lesser data copying artifacts in the initial phase of training. However, the adapter quality and strength are sub-par due to inadequate training (i.e. the style is not visible in the image). This is visible in HPS-v2 alignment scores. The images produced are similar to those from the base model, and hence lesser artifacts exist. As the training epochs increase, images start to represent the adapter style (represented by HPS scores). Once we reach this point, the number of data-copying artifacts increase significantly in LoRA, as tracked by the LPIPS-diversity. FouRA can achieve the adapter style while being able to produce a diverse range of images, as seen in Fig. 1.

https://doi.org/10.52202/079017-2284

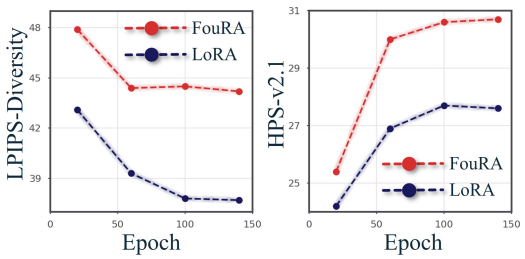


Figure E.3: **Studying the training curves for signs of data-copying artifacts:** We analyzed the effect of early stopping of training by measuring the performance. All results are with rank 64 and $\alpha = 0.8$ on the paintings adapter.

E.4 Additional Computational Analysis

In Section 5.5, we compared LoRA v/s FouRA in terms of training memory and inference time. In this Section, we provide additional computational analysis of our approach. As shown in Figure E.4, we analyzed performance of FouRA v/s LoRA with varying training complexity (training time, memory usage). To vary time, we report HPS scores of FouRA v/s LoRA at intermediate epochs. To vary the memory, we use rank. We observe that FouRA consistently achieves better performance v/s compute operating points compared to LoRA.

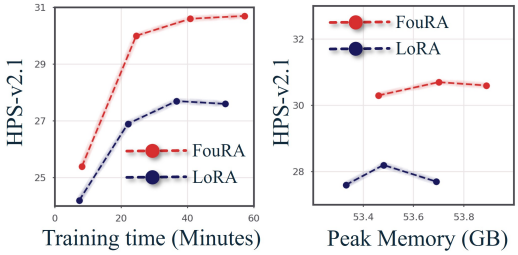


Figure E.4: **Training complexity v/s performance:** We perform an analysis of training complexity v/s performance. This follows two settings: Varying the training epoch (left) to measure training time and Varying the rank (right) to measure peak training GPU memory. We measure HPS as the performance metric. All results are with $\alpha = 0.8$ on the paintings validation set.

Additionally, we showed how the training memory overhead scales with batch-size in Table E.4. We observe that the FouRA memory overhead during training time is negligible and only 0.3-0.4% over LoRA.

Batch Size	8	6	4	2
LoRA	53687 MB	40872 MB	28151 MB	15499 MB
FouRA	53894 MB	41020 MB	28255 MB	15448 MB

Table E.4: **Memory Overhead/Scaling with batch size:** We report the scaling of training memory based on batch size.

E.5 Additional Visual Results on Text-to-Image Stylization

In Figure E.5, we provide additional visual results for FouRA and LoRA finetuning on the *Bluefire* dataset at varying adapter strengths. Within the generated images, the concepts 'Football' and 'Dog' are unseen. As observed, FouRA produces aesthetically appealing images as compared to LoRA in all cases. This is more evident in the 'Football' example. As observed, FouRA can generalize better to new concepts, as compared to LoRA.

In Figure E.6, we show additional results obtained by finetuning the Realistic Vision Model with FouRA adapters on our curated style datasets, *3d*, *Origami* and *Paintings*. As observed, FouRA is capable of generating a diverse set of aesthetically appealing images.

https://doi.org/10.52202/079017-2284

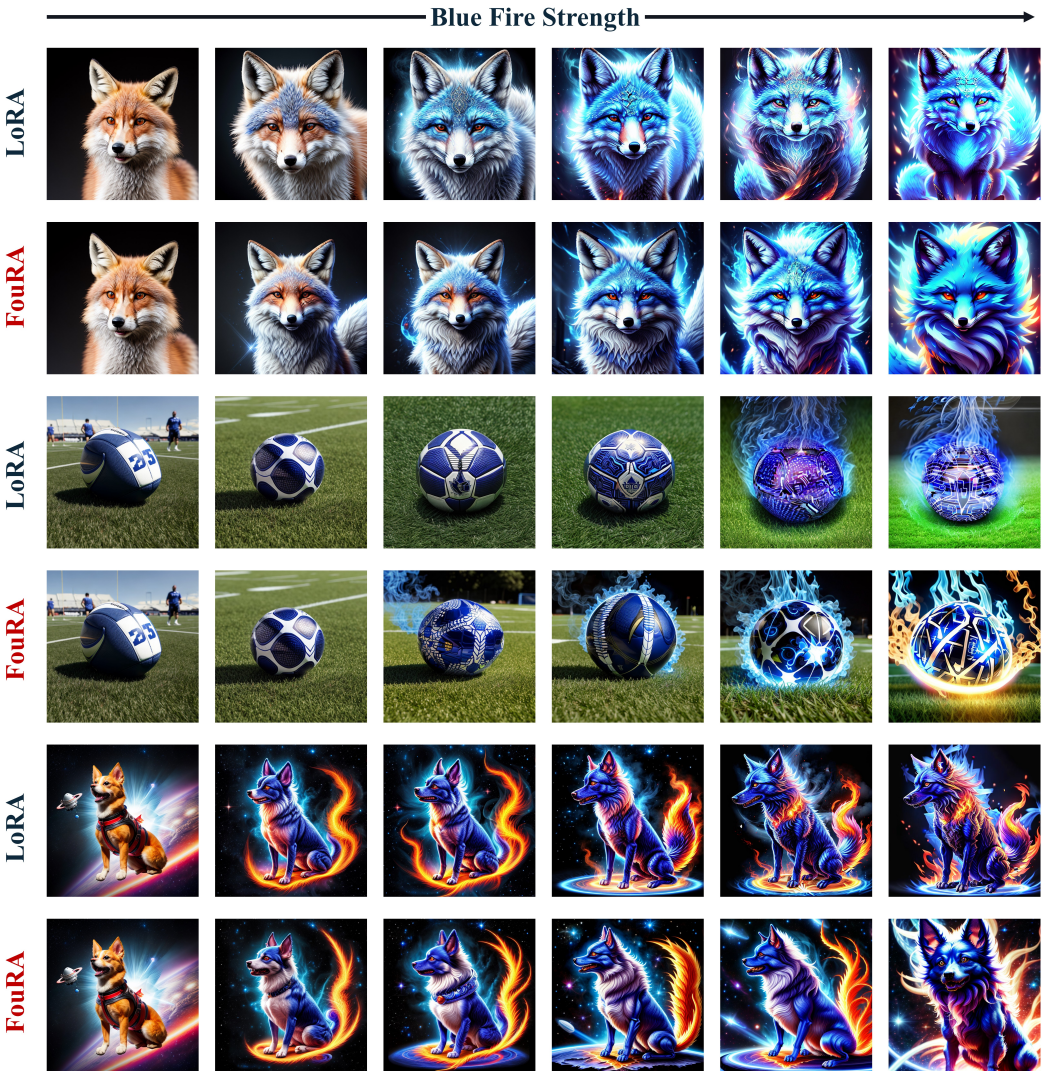


Figure E.5: **Visual Results using *BlueFire*** adapters comparing LoRA and FouRA at varying values of α.

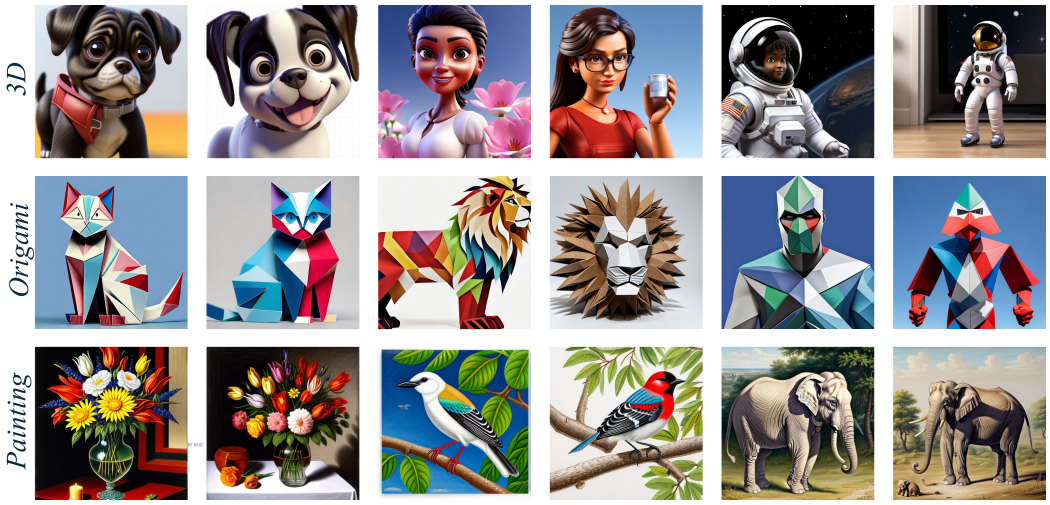


Figure E.6: **Images generated by FouRA** trained on *3D*, *Paintings* and *Origami* datasets.

https://doi.org/10.52202/079017-2284

F Additional Experiments for Text-to-Image Editing using Concept Sliders

Concept sliders provide a framework to train LoRA adapters on single (image, prompt) pair (for example: "very old, wrinkly, gray hair, aged skin") in conjunction with multiple attributes (for example: Male person, very old etc). The disentanglement objective operates on the semantic space of diffusion models constraining the edit to occur only along the direction of the concept without changing the attributes.

From 4 we learnt that ΔW has a small eigen spread leading to more compact representation. Our method favous lower effective rank and the trained model naturally converges to decorrelated subspaces from the base model weights B.3 . In addition in an informal proof B.4 we show that one can leverage the properties of FouRA to learn composition of concepts with less interference with the subspace of other concepts.

We compare the performance of FouRA with LoRA when trained on explicit pairs of prompts across 20 different attributes acting as guidance. We train 3 sliders *"curly hair"*, *"surprise face"* and *"Age slider"* on both the baseline LoRA and our adapter for upto 1000 steps. We trained the model on $rank = 8$. We show that despite explicit training on pairs, low rank adapter space is still prone to changes in gender and race for strong adapter scales especially strength ≥ 4. Below we show results on Single Adapter and Composite adapter.

Single Concept We follow the SDEdit style inference where the adapter kicks in after $\mathcal{T} \in (750, 800, 850)$ timesteps. We notice that the effect of adapter in FouRA-DCT is far less below 800. Refer to figures below for more examples. For our results we fixed the $\mathcal{T} = 800$. We evaluate our results on LPIPS F.4. While our adapter is far more stable compared to LoRA adapter between the strengths $[-6, 6]$. We also note that FouRA on DCT slightly better performance over FFT and for brevity we only show results on DCT. We note that FouRa maintains the balance between prompt and style fidelity and the quality of generated images.

Below are some of the examples of *Age*,

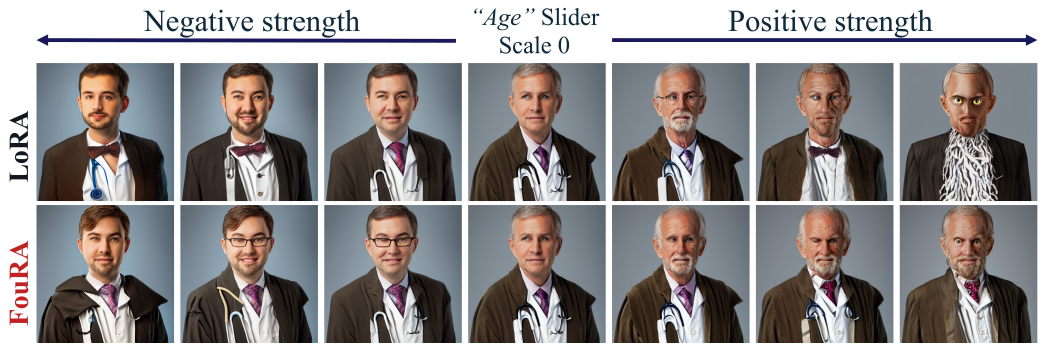


Figure F.1: *Age* Slider, LoRA (top) vs FouRA (bottom). We find that as the strength increases there are more prominent skin tone variations in LoRA.

Figure F.2: *Age* FouRA Slider, "Portrait of a doctor" (top) and "Photo of an Hispanic man" (bottom).

In general Age sliders shows a good improvement on LPIPS score for strength above 3 as shown in figure F.4. We notice that as the strength increases FouRA disentangles from other attributes better.

We also train an adapter to change the strength of curls in hair. Below we show more examples for curly hair. We notice that the both LoRA and FouRA adapters are sensitive to increasing strength. As can be observed LPIPS score are higher for *Hair* than for *Age*. As the strength increases the LoRA adapter tend move in the direction of increased prompt fidelity and removing the face of the person or crunching the face to add more details of hair in LoRA. We show the quanitative results for the same using LPIPS. We observe that across strengths $1 \leq 5$ the FouRA has much smaller LPIPS score. Please refer to the right figure in 8. Below we share more examples of FouRA on other prompts.

Figure F.3: *Hair* Slider: We find that as the strength of the adapter increases the curls increase. In the top image we also see minor variations in the facial details of the person.

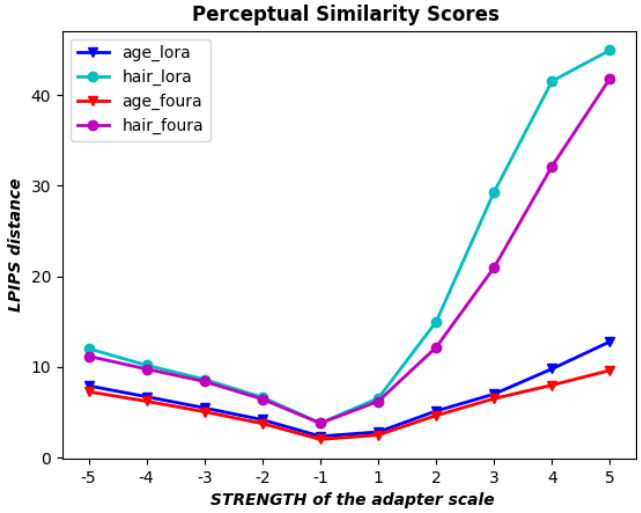


Figure F.4: **Perceptual metric drops for LoRA compared to FouRA for the sliders on "age" and "hair"**. These were tested across 10 scales from (-5, 5). Similarity score was computed across 1000 images and 500 prompts of 10 seeds each.

Composite LoRA : Below we show the results for combining adapters. To combine adapters, we varied the strengths of Adapter 1 between $strengths \in (-8, 8)$ and Adapter 2 between $strengths \in (-8, 8)$. We show some examples of only FouRA F.5 for combined *hair* and *Age* adapter. We show the images for when the adapter strengths are equal i.e increase from $(-6, 6)$ to $(6, 6)$.

Below we show comparison between LoRA and FouRA across different adapter strengths. We emphasize the effect when one slider for e.g *"Age"* has a very high adapter strength on the second slider when the strength is low (bottom left image). We observe that for LoRA the facial distortions when both adapter strengths are high (bottom right) are very evident. The Age adapter in general seems to interfere more with the *Hair* for higher strengths.

https://doi.org/10.52202/079017-2284

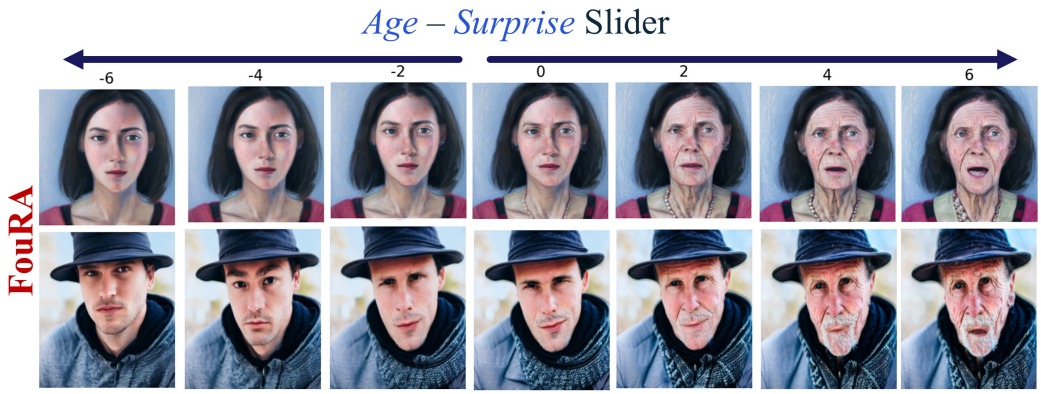


Figure F.5: **Composite FouRA** . Composite *surprised*, *age* slider. Here we show the combined adapter as the strengths of each adapter are jointly incremented in each step in the image. The adapter strengths are (-6 6) for left most image and (6,6) for the right most image. The positive prompt for *surprised face* prompt: **"looking surprised, wide eyes, open mouth"**

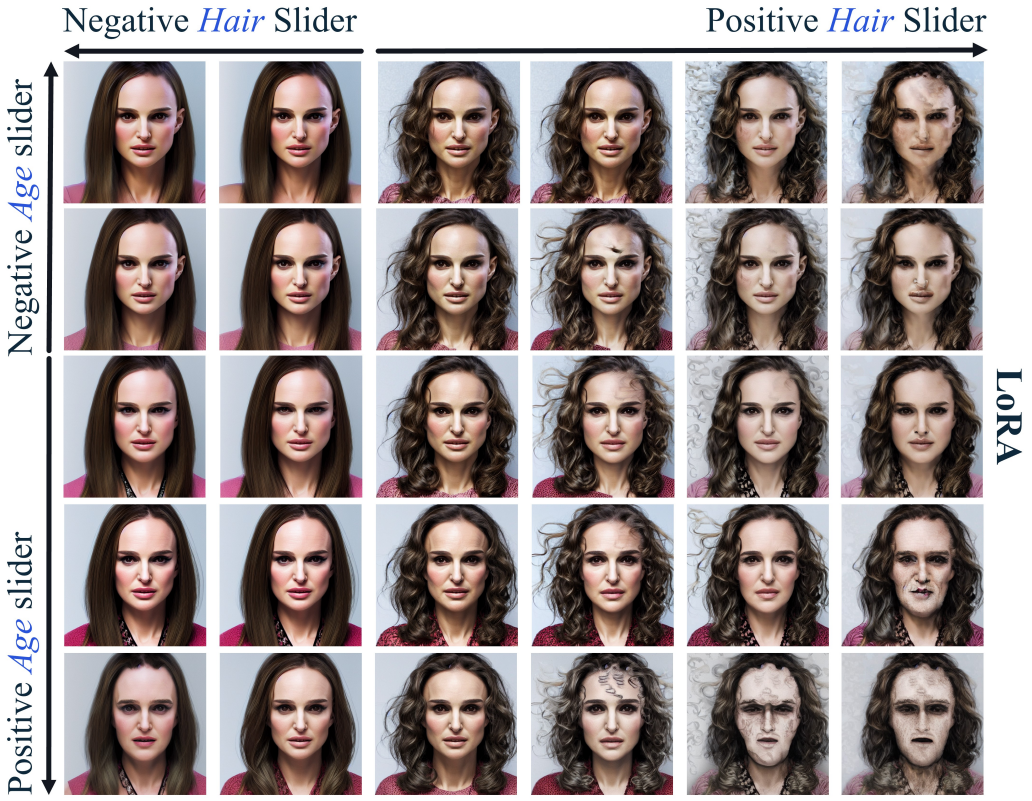


Figure F.6: **Composite LoRA** . Composite *hair*, *age* slider. We find that for higher strength of *Age* adapter as we increase the strength of *Hair*, adapter seems to interfere with the facial features and almost distort the face. However for lower values of *Hair* adapter. Here we show scales between -6 to 8

G FouRA on General Language Understanding Tasks

While our design choices for FouRA are primarily motivated for vision tasks, we evaluate its efficacy on langauge tasks in Tab. G.1, and compare FouRA against another adaptive rank selection approach, SoRA, designed specifically for language tasks [3]. Results show that FouRA's rank selection in frequency domain outperforms SoRA on four out of the six GLUE benchmarks we evaluated on, demonstrating that the feature disentanglement induced by FouRA can be used beyond vision tasks.

https://doi.org/10.52202/079017-2284

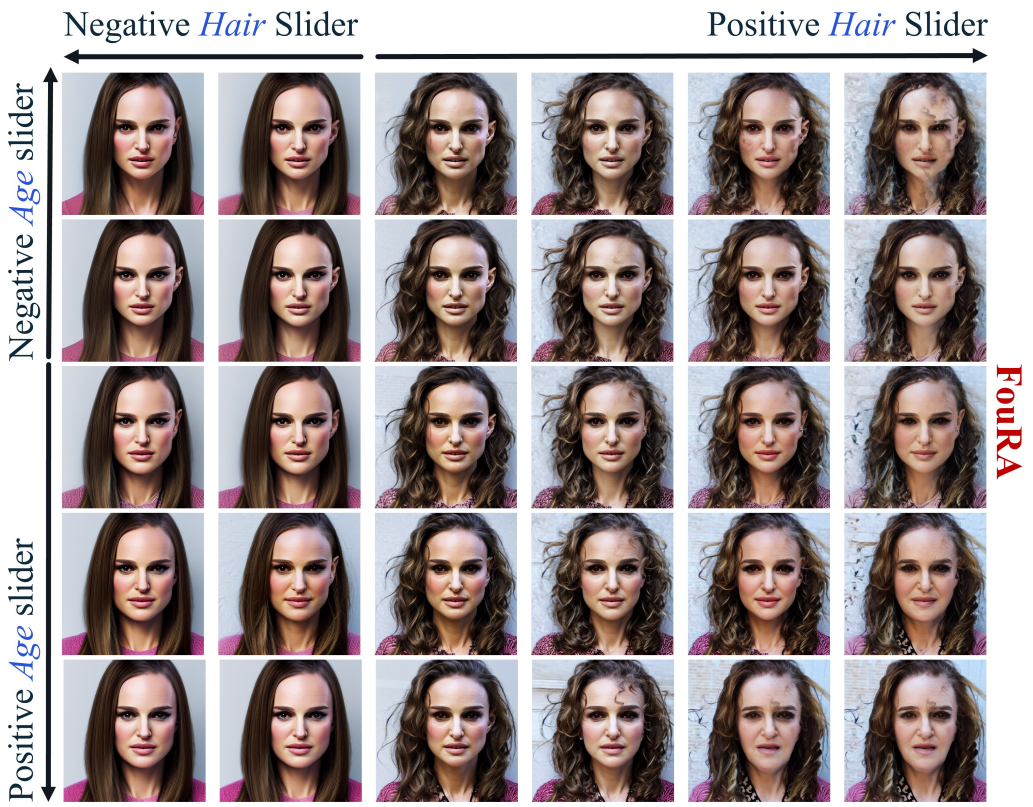


Figure F.7: **Composite FouRA** . Composite *hair*, *age* slider. We note that the adapter is stable for many prompts and seeds upto scale of 8. There are artifacts at large scales strength upto scale=8 of positive slider, however we find that artifacts are fewer and don't distort the facial features.

Adapter	MNLI	CoLA	SST2	STSB	MRPC	QNLI
LoRA	90.2 ± 0.2	67.3 ± 0.8	94.9 ± 0.3	89.9 ± 0.3	90.3 ± 0.6	93.6 ± 0.6
SoRA	$\mathbf{90.5 \pm 0.1}$	69.9 ± 0.8	95.2 ± 0.4	91.4 ± 0.1	$\mathbf{90.6 \pm 0.8}$	93.9 ± 0.3
FouRA	$\mathbf{90.5 \pm 0.1}$	$\mathbf{70.6 \pm 0.7}$	$\mathbf{95.5 \pm 0.4}$	$\mathbf{91.6 \pm 0.1}$	90.4 ± 0.5	$\mathbf{94.2 \pm 0.5}$

Table G.1: Evaluation of DeBERTa-V3 on the GLUE benchmarks, averaged over 3 seeds.

H Societal Impacts

In this section, we discuss the societal impacts of our work. While there are benefits of training FouRA modules as highlighted in the main text, we consider that it can potentially have larger societal impacts. One of the major challenges of text-to-image models is digital forgery, highlighted in previous works [39, 40]. We observed that finetuning low-rank adapters on various tasks in image generation can lead to replication of the input image. This is due to the overfitting of LoRA on a small training set. However, we demonstrate in the paper how FouRA can push the generalization error bound further, hence resolving the data forgery problem to a great extent. Hence, we propose to utilize FouRA in applications where it is imperative to hide the training set, such that it can't be replicated.

I Limitations

FouRA, as demonstrated in the main text, is a highly effective parameter efficient fine-tuning method. However, as it makes use of frequency transforms (dft, dct), one potential limitation is that current Deep Learning hardware systems are not as optimal for frequency transform operations, as they are for matrix multiplies and convolutions. However, with astute recent works such as [38, 24, 28], their

https://doi.org/10.52202/079017-2284

popularity has increased in the field of Deep Learning. Hence, we foresee that it is only a matter of time before DL hardware systems get heavily optimized for frequency transforms.

J Future Work

We have demonstrated that FouRA achieves great performance on tasks such as image generation, Image concept and style editing on Vision tasks in diffusion framework. A good extension of FouRA would be to explore the generalization capabilities to reuse the learnt basis on other adapters trained on different datasets. Additionally, for the FouRA module we would like to explore direct token masking in the frequency domain, as we observed some initial indicators, effectively correlating bands of frequencies and various characteristics of generated images. Seeing the performance of FouRA, we feel encouraged to think that frequency domain fine-tuning of adapters will potentially be a popular research direction in the coming years.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper provides detailed experimentation results and related theory which accuracy reflects the paper's contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Appendix I

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

https://doi.org/10.52202/079017-2284

Justification: Both the provided lemmas are proved in Appendix B

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All implementation details are available in Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Datasets and code will be provided upon request, as we need a legal approval for the same. We are also working on the legal process to provide git access.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All implementation details are available in Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report standard deviation over 30 seeds for the main experiments in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

https://doi.org/10.52202/079017-2284

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All computational analysis is available in Table 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We conform to NeurIPS code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We mention societal impacts in Appendix H.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Not Applicable

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We follow the license terms for every model and dataset we use.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

https://doi.org/10.52202/079017-2284

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: All assets are documented in Appendix C

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Not Applicable

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Not Applicable

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.