DECOUPLING ANGLES AND STRENGTH IN LOW-RANK ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Parameter Efficient FineTuning (PEFT) methods have recently gained extreme popularity thanks to the vast availability of large-scale models, allowing to quickly adapt pretrained models to downstream tasks with minimal computational costs. However, current additive finetuning methods such as LoRA show low robustness to prolonged training and hyperparameter choices, not allowing for optimal out-of-the-box usage. On the other hand, multiplicative and bounded approaches such as ETHER, even if providing higher robustness, only allow for extremely low-rank adaptations and are limited to a fixed-strength transformation, hindering the expressive power of the adaptation. In this work, we propose the DeLoRA finetuning method that first normalizes and then scales the learnable low-rank matrices, thus effectively bounding the transformation strength, which leads to increased hyperparameter robustness at no cost in performance. We show that this proposed approach effectively and consistently improves over popular PEFT methods by evaluating our method on two finetuning tasks, subject-driven image generation and LLM instruction tuning. Code will be released upon acceptance.

1 INTRODUCTION

The rapid advancement of deep learning has led to the development of large-scale pretrained models in various domains, especially in computer vision and natural language processing (Touvron et al., 2023a;b; Radford et al., 2021; Rombach et al., 2022). However, the enormous size of these models, reaching billions of parameters, presents significant challenges when adapting them to specific downstream tasks, particularly in terms of computational cost and efficiency. To address these challenges, Parameter Efficient FineTuning (PEFT) methods have emerged. PEFT methods are characterized by introducing a small set of learnable parameters compared to full finetuning. Notable examples include adapters (Houlsby et al., 2019) and prompt tuning (Lester et al., 2021). In this work, we focus on improving LoRA (Hu et al., 2022), which is a simple and effective finetuning method. Despite its success, LoRA exhibits high sensitivity to hyperparameter choices (Biderman et al., 2024) and exhibits performance degradation during extended finetuning. While robust finetuning approaches such as ETHER and ETHER+ (Bini et al., 2024) address some of these limitations, they are constrained to extremely low-rank adaptations and fixed-strength transformations.

Therefore, we propose DeLoRA, an enhanced version of LoRA that introduces a boundary on the weight updates through normalization. DeLoRA decouples the learning of the LoRA matrices into directional and scale components, enabling greater adaptability to various settings while maintaining the ability for personalization and merging at inference time. We motivate DeLoRA from two different perspectives, as a derivative of LoRA by introducing additional normalization, and as a derivative of ETHER by introducing the possibility of high-rank updates. We ablate the design choices accordingly and show that we improve over both LoRA and ETHER. Furthermore, we validate the benefits of DeLoRA by evaluating it on different tasks in image-generation and LLM adaptation.

In summary, we make the following contributions in this work: (1) We thoroughly review the formulations of LoRA and ETHER and hence derive a novel PEFT method, DeLoRA; (2) We demonstrate that DeLORA is more robust than LoRA in terms of hyperparameter choices and for extended finetuning regimes; (3) We extensively ablate the formulation of DeLoRA by deriving it from both LoRA on the one hand, and from ETHER on the other hand; (4) We evaluate DeLoRA on both vision and language benchmarks, matching or surpassing the performance of competing PEFT methods.

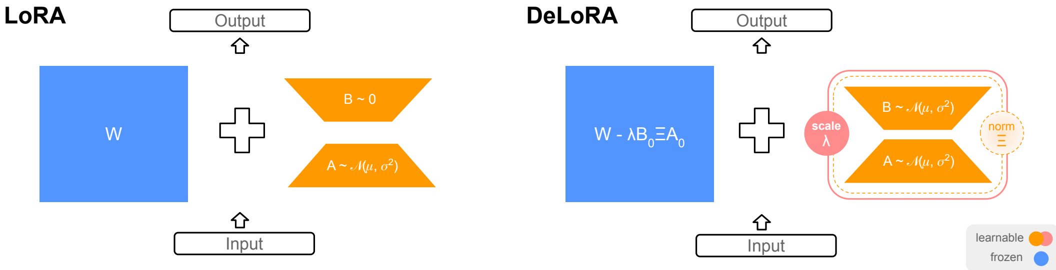

Figure 1: (Left) Visualization of the original LoRA (Hu et al., 2022). (Right) Visualization of our proposed DeLoRA. In addition to the low-rank matrices B, A, we introduce a normalization Ξ and a scaling factor λ, which effectively decouple the angular learning from the transformation strength.

2 Decoupled Low-rank Adaptation (DeLoRA)

Our Decoupled Low-rank Adaptation approach, by introducing learnable boundaries on the weight updates, effectively combines the strengths of LoRA and ETHER methods, allowing for high expressivity and finetuning robustness. In the following sections, we will (i) present an overview of the PEFT methods LoRA and ETHER, focusing on their respective limitations (Section 2.1) and (ii) describe how we derive our proposed DeLoRA method from both perspectives (Section 2.2)

2.1 Preliminaries: LoRA & ETHER, and Their Limitations

Here, we provide a detailed review of LoRA (Hu et al., 2022) and ETHER (Bini et al., 2024), with a particular focus on their limitations.

Low-rank Adaptation (LoRA). Hu et al. (2022) proposed *Low-rank Adaptation* (*LoRA*), which parametrizes the update of pretrained weights $W \in \mathbb{R}^{d \times f}$ during finetuning as

$$\left(W + \frac{\alpha}{r} BA\right)^{\intercal} x + b \tag{1}$$

where $A \in \mathbb{R}^{r \times d}$ and $B \in \mathbb{R}^{f \times r}$ are the learnable matrices, α is a scaling factor, and r is the rank of the final BA matrix. When $r \ll \min(d, f)$, LoRA reduces the required finetuning parameters significantly compared to full finetuning. In addition, BA can be merged into W during inference to avoid additional latency.

However, LoRA is known to be highly sensitive to hyperparameter choices (Biderman et al., 2024), and it is easily affected by over-training (Qiu et al., 2023), thus requiring careful tuning and experimentation to achieve an optimal balance between a sufficiently high learning rate and avoiding catastrophic overwriting of the pretrained weights, also known as catastrophic forgetting. In our proposed DeLoRA, we remove this behavior by introducing a boundary on the weight updates. Thus, DeLoRA achieves good performance for a wide range of learning rates.

Finetuning with Hyperplane Reflections (ETHER). Following efficiency and robustness arguments, Bini et al. (2024) propose to employ bounded transformations for finetuning, namely ETHER and ETHER+. ETHER (left side in Eq. (2)) and ETHER+ (right side) introduce multiplicative transformations H or H^+ respectively, which act on the pretrained weights as follows:

$$(HW)^{\intercal} x + b \quad , \quad \left(H^+ W \tilde{H}^+\right)^{\intercal} x + b. \tag{2}$$

Here, $H = I - 2uu^{\intercal}$, $H^+ = I - uu^{\intercal} + vv^{\intercal}$, $\tilde{H}^+ = I - \tilde{u}\tilde{u}^{\intercal} + \tilde{v}\tilde{v}^{\intercal}$ (where u, v, $\tilde{u}$, $\tilde{v}$ are unit vectors) are bounded in terms of their distance to the identity transformation, as per

$$\|H - I\|_F = 2 \quad , \quad \left\|H^+ - I\right\|_F \leq 2, \tag{3}$$

where the subscript $_F$ denotes the Frobenius norm. This upper bound on the transformation distance prevents weight changes that cause catastrophic forgetting, as shown by Bini et al. (2024).

However, enforcing a constant boundary on the transformation distance can limit the finetuning performance, as the boundary may be too strict to adapt the layer or pretrained model at hand to the respective task. Furthermore, by rewriting the formulations in Eq. (2) in a residual form, we can show that the weight updates are intrinsically limited to be low-rank (see Appendix A), which limits the finetuning capacity of ETHER. With DeLoRA, by introducing a normalization and a scaling factor to LoRA matrices, we allow for controlling both boundary and rank, which effectively leads to better performance.

2.2 DeLoRA

While both LoRA and ETHER demonstrate valuable properties, namely parameter efficiency and robustness, they also exhibit notable limitations. Our proposed PEFT method, DeLoRA, addresses these shortcomings by synthesizing the strengths of both approaches. In this regard, DeLoRA can be thought of as an extension of LoRA that incorporates ETHER's robustness properties or, alternatively, as an enhancement of ETHER that adopts LoRA's more expressive paradigm. In the following, we will present both derivation and finally summarize in a concise way our proposed DeLoRA formulation.

Deriving DeLoRA from LoRA. In order to achieve robustness to learning rates, we first observe that in LoRA's Eq. (1) the strength of the weight updates ΔW is proportional to ΔBA, which in turn is proportional to the learning rate. This means that the update strength at each training step is directly controlled by the learning rate, which can lead to catastrophic forgetting in high learning rate regimes. In order to mitigate this behavior, we decompose the BA matrix into the sum of its rank-1 components, i.e.

$$BA = \sum_{i=1}^{r} b_i a_i^\intercal \tag{4}$$

■ *Controllable Boundary.* Similarly to ETHER, we normalize each rank-1 entry, making the Frobenius norm of each single rank-1 component a constant. This normalization can be written as

$$\sum_{i=1}^{r} \frac{b_i a_i^\intercal}{\|b_i\|\|a_i\|} = B\Xi A \tag{5}$$

where Ξ is a diagonal matrix with entries $\Xi_{i,i} = \frac{1}{\|b_i\|\|a_i\|}$ for $i = 1, \ldots, r$, $\Xi_{i,j} = 0$ for $i, j = 1, \ldots, r, i \neq j$. The final update distance with respect to the pretrained weights thus is bounded as

$$\|B\Xi A\| = \left\| \sum_{i=1}^{r} b_i a_i^\intercal \right\| \leq \sum_{i=1}^{r} \|b_i a_i^\intercal\| = r. \tag{6}$$

Most importantly, the boundary is independent of the used learning rate. To control the boundary and remove its rank dependency we scale $B\Xi A$ by a factor $\frac{\lambda}{r}$, so that the boundary becomes λ, as in

$$\left\| \frac{\lambda}{r} B\Xi A \right\| \leq \lambda. \tag{7}$$

Now, the boundary can be chosen arbitrarily to fit the pretrained network or task at hand. To allow for layer-aware boundaries without the need to manually choose them, we introduce a different learnable λ to each layer. Then, finetuning adapts the value of each λ accordingly. Hence, we effectively decouple the angular learning (the normalized $B\Xi A$ matrices) from the transformation strength, as measured by the boundary λ. Furthermore, introducing a single additional learnable parameter λ to each finetuned matrix creates only negligible overhead in terms of overall trainable parameters and training speed.

■ *Weights-norm Scaling.* Previous works suggest that when finetuning image generative models, multiplicative finetuning methods show stronger performance (Qiu et al., 2023; Liu et al., 2024b) than additive finetuning methods like LoRA. This might be due to the fact that in multiplicative methods, weight updates ΔW are proportional to the pretrained weights W, which makes each update layer-aware. To mimic this approach in our additive proposed method DeLoRA, we introduce a scaling factor equal to the pretrained weights norm. This can be formally stated as

$$\Delta W = \frac{\lambda\|W\|}{r} B\Xi A. \tag{8}$$

This term allows to adapt different layers in a more layer-aware manner. This is especially relevant when adapting a diverse set of layers, which is the case for our image generative models tasks. Our ablation studies demonstrate these performance improvements empirically (see Section 3.2).

■ *Initialization.* To initialize the finetuning process from the pretrained model, since following LoRA's zero initialization would cause instabilities because of the introduced normalization term, we take inspiration from (Bini et al., 2024) and subtract a copy of the kaiming-randomly initialized B and A matrices. With respect to (Bini et al., 2024), we simply freeze these additional parameters and merge them to the pretrained weights, as in

$$W = \bar{W} - \left(\frac{\lambda\|W\|}{r} B\Xi A\right)_0 \tag{9}$$

where $\bar{W}$ is the original pretrained matrix, and $(\frac{\lambda\|W\|}{r} B\Xi A)_0$ is the update matrix at time 0.

Deriving DeLoRA from ETHER So far, we showed how to derive DeLORA from LoRA. Alternatively, it is possible to derive DeLoRA by introducing properties of LoRA to ETHER. We find this to be insightful to understand the impact of each individual component from a theoretical perspective. In addition, we quantitatively ablate all innovations of DeLoRA in Section 3.2.

■ *Controllable Boundary.* One of the main limitations of ETHER is its fixed boundary (see Section 2.1), which is always constant thus cannot be adapted to the pretrained model at hand. We address this limitation by introducing a scaling parameter λ to ETHER as in

$$H = I - \lambda u u^{\intercal} \quad , \quad H^{+} = I - \frac{\lambda}{2} u u^{\intercal} + \frac{\lambda}{2} v v^{\intercal}. \tag{10}$$

Then, the boundaries on the distances of H and H^{+} to the identity matrix become $\|H - I\|_F = \lambda$, and $\|H^{+} - I\|_F \leq \lambda$. In Section 3.2, we show that this modification, i.e. introducing a controllable bound, leads to the largest increase in performance.

■ *Increasing the rank.* In Appendix A, we show that ETHER and ETHER+ are restricted to rank-1 and rank-4 weight updates respectively. In order to arbitrarily control the rank, we extend the H^{+} parameter of ETHER+ to $\hat{H}$, which allows for an arbitrary number of weight reflection operations:

$$\hat{H} = I - \sum_{i=1}^{r/2} u_i u_i^{\intercal} + \sum_{i=1}^{r/2} v_i v_i^{\intercal}. \tag{11}$$

We can rewrite $\hat{H}$ by gathering the u and v unit vectors into two rank-$\frac{r}{2}$ matrices, as in

$$\hat{H} = I - U\Sigma U^{\intercal} + V\Theta V^{\intercal}, \tag{12}$$

where Σ and Θ are diagonal normalization matrices with entries $\Sigma_{i,i} = \frac{1}{\|u_i\|^2}$, $\Theta_{i,i} = \frac{1}{\|v_i\|^2}$, The entries on the diagonals of Σ and Θ are constructed to normalize u and v to unit vectors. Thus, the distance from the identity matrix becomes

$$\|\hat{H} - I\| = \left\| \sum_{i=1}^{r/2} u_i u_i^{\intercal} - \sum_{i=1}^{r/2} v_i v_i^{\intercal} \right\| \leq \sum_{i=1}^{r/2} \|u_i u_i^{\intercal}\| + \sum_{i=1}^{r/2} \|v_i v_i^{\intercal}\| = r. \tag{13}$$

As above, we can control the boundary on the distance by introducing a scaling factor λ as in

$$\hat{H} = I - \frac{\lambda}{r} U\Sigma U^{\intercal} + \frac{\lambda}{r} V\Theta V^{\intercal} \tag{14}$$

■ *U,V Relaxation.* Finally, we relax $U\Sigma U^{\intercal}, V\Theta V^{\intercal}$ and replace them with distinguished trainable matrices $B\Xi A$ and $D\Phi C$ respectively, which leads to $\hat{H} = I - \frac{\lambda}{r}(B\Xi A - D\Phi C)W$. We highlight how such a formulation resembles a multiplicative analogous of our proposed DeLoRA method, and we also introduce this version in our ablation study. We ablate all alternatives in Section 3.2. There, we find that DeLoRA, combined with weights-norm scaled updates, as in multiplicative finetuning, achieves overall stronger performance.

DeLoRA formulation. Summarizing, our proposed DeLoRA finetuning method consists in learning a normalized low-rank matrix $B\Xi A$ and a scale λ, updating the pretrained weights as in

$$\left(W + \frac{\lambda\|\bar{W}\|}{r} B\Xi A\right)^{\intercal} x + b \tag{15}$$

This formulation constrains by construction the learnable finetuning updates with a $\lambda\|\bar{W}\|$-sized boundary, where $\bar{W}$ is the norm of the pretrained weights, effectively decoupling the transformation strength from the angular learning. In more detail, the key components are:

- *Normalization:* Ξ is a r-dimensional diagonal matrix that normalizes LoRA's inner low-dimensional bottleneck (Eq. (5)), bounding the Frobenius norm of $B\Xi A$ to r (Eq. (6)).
- *Scaling Factors:* (i) $1/r$ is used to remove the rank dependency on the boundary dimensionality, (ii) $\|\bar{W}\|$ to make the weight updates proportional to the pretrained weights, and (iii) λ to control the adaptation strength and allow for a layer-specific boundary adaptation (Eq. (7))
- *Initialization:* Pretrained initialization follows by merging to the pretrained weights a frozen copy of the initialized finetuning adaptation matrices (Eq. (9)).

DoRA vs DeLoRA discussion. DoRA (Liu et al., 2024a), similarly to our work, addresses the finetuning process by trying to decouple the learning of magnitudes and angles, by using a formulation that leads to weight updates $W' = m\frac{W+\Delta W}{\|W+\Delta W\|}$. We can summarize the key differences of DoRA with respect to our proposal into two: (i) the normalization and scaling operations happen on the fully finetuned weights, and (ii) these operations happen on the column space of the weight matrices, which draw a significant difference to our proposal. We argue that DeLoRA finetuning (i) by introducing the normalization and scaling operations directly on the weight updates ΔW, it more directly tackles the goal of not diverging from the pretrained model, and (ii) by normalizing the inner low-dimensional space (rather than the column space), it actually results in an implicit Frobenius-distance boundary, which acts as a mathematical guarantee for non-divergence. These eventually lead to (i) peculiar training dynamics (as shown in Fig. 3, whereas DoRA and LoRA show similar behavior), and (ii) better decoupling, supported by the strong robustness results in Fig. 2. In this regard, we notice that even if DeLoRA, by having a learnable boundary, in principle also has an unbounded Frobenius distance, in practice divergence does not happen, as shown in Fig. 2. This demonstrates that during finetuning, DeLoRA's learnable boundary is able to effectively adjust and avoid divergence from the pretrained weights, behavior that does not happen with DoRA.

3 Experiments

In this section, we evaluate our proposed DeLoRA method for image generation and natural language understanding, and instruction tuning tasks. We begin by providing a detailed description of these tasks and their relevance. To justify our design choices, we present a comprehensive ablation study that highlights the key innovations of DeLoRA. Finally, we demonstrate that DeLoRA not only matches or exceeds the performance of LoRA and other state-of-the-art methods but also exhibits superior robustness. This enhanced stability is particularly evident in two aspects: reduced sensitivity to learning rate selection and improved performance retention during extended finetuning periods.

3.1 Tasks

Semantic Map to Image We finetune Stable Diffusion models (Rombach et al., 2022) to generate a realistically-looking image based on a given segmentation map. The image should follow the spatial structure laid out in the segmentation map as closely as possible. Examples of segmentation maps and generated images are in Fig. 7 (right side). As control signal, we use the pretrained encoder from ControlNet (Zhang et al., 2023a). For training and evaluation, we take semantic maps and images from the ADE20K dataset (Zhou et al., 2019). After training, we generate images for 2000 segmentation masks from the ADE20K validation set and report the mean Intersection-over-Union (mIoU) and accuracy of semantic maps as predicted by UperNet-101 (Xiao et al., 2018). Note that we only use the Semantic Map to Image task to ablate our method design decisions.

Subject-driven Image Generation. Following (Qiu et al., 2023; Ruiz et al., 2023), we evaluate the efficacy of our proposed methods in the DreamBooth setting, i.e. adapting Stable Diffusion to recontextualize a subject shown in a set of images based on a given prompt. The data is taken from (Ruiz et al., 2023) and consists of 30 subjects and 25 prompts each. The task is to adapt Stable Diffusion to generate images showing the given subject in the context specified by the prompts. An example is in Fig. 7 (left side). For each combination of image and prompt, after finetuning, we generate four images and measure the subject-fidelity by DINO (Caron et al., 2021) and CLIP (Radford et al., 2021), as proposed by (Ruiz et al., 2023). Here, the score represents the similarity of generated and given images, measuring the faithfulness of generating images of the given subject to the provided real images. Among the two metrics, the DINO score is more significant since it is more sensitive to subject-unique features (Ruiz et al., 2023).

Natural Language Understanding We test how DeLoRA performs in adapting Language Models for Natural Language Understanding on the GLUE benchmark Wang et al. (2018) finetuning a pretrained RoBERTa-base model Liu et al. (2020). GLUE tasks have been extensively used to measure natural language understanding performance, comprising inference tasks (MNLI, QNLI, RTE), sentiment classification (SST-2), and correct identification of English grammatical structures (CoLA). CoLA results refer to Matthews correlation coefficient, MNLI to matched accuracy, and STS-B to average correlation, while all other tasks are evaluated on accuracy.

Instruction Tuning. We evaluate how effectively DeLoRA can adapt LLMs to follow user-given instructions. For this experiment, we finetune LLaMA-2-7B (Touvron et al., 2023b) on the Alpaca dataset (Taori et al., 2023). Following Bini et al. 2024, we finetune on the full dataset for one epoch and evaluate the zero-shot performance of instruction-tuned models on four different tasks, namely (1) Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2021), which features 57 tasks in different categories such as STEM, Humanities, and Social Sciences; (2) AI2 Reasoning Challenge (ARC) (Clark et al., 2018), which contains over 7000 grade-school science questions; (3) TruthfulQA (Lin et al., 2022), which contains 817 questions representing common misconceptions in 38 categories like health, law, finance and politics. TruthfulQA additionally features two separate sub-tasks, namely single-true and multi-true. In single-true, only one of the provided answers is correct, and the model has to select the unique correct answer. In multi-true, several of the provided answers may be correct, and the model has to assign a high probability to correct answers and a low probability to incorrect answers. We report scores for both tasks separately.

3.2 Ablation of DeLoRA design choices

In this section,, we ablate the different design decisions that equip DeLoRA with an advantage with respect to LoRA and ETHER. From the LoRA derivation (top to bottom in Tables 1,2), we show how adding normalization with a controllable boundary and weight scaling to pretrained matrices yields performance improvements. From the ETHER derivation (bottom to top in Tables 1,2), we show how the introduction of a controllable scale, a higher-rank formulation, the learnable matrices relaxation, and the additive finetuning tranformation, lead to incrementally improved performance.

Results for subject-driven image generation are in Table 1. For this ablation we use a small-scale version of the setting proposed by (Ruiz et al., 2023), finetuning 3 subjects over 25 prompts each. Among all modifications, we notice how the introduction of a controllable boundary in ETHER+ (one-sided) has the highest impact, raising the DINO score from 0.624 to 0.678 and the CLIP score from 0.746 to 0.810. This shows how the lack of strength is the hindering factor for ETHER+(one-sided), as already noted by (Bini et al., 2024). Starting from LoRA, we notice how the weight-scaling has the largest impact on performance, raising the DINO score from 0.682 to 0.701 and the CLIP score from 0.809 to 0.825. Additionally, we note that DeLoRA's performance without the weight-norm scaling falls short compared to its multiplicative counterpart.

For the Semantic Map to Image ablation study, we run a small-scale grid search by finetuning Stable Diffusion for 10 epochs on ADE20K in bfloat16 precision. Results are reported in Table Table 2. We note how DeLoRA achieves best controllability among different variations. In addition, we also note the increase in Accuracy when increasing the rank of ETHER+, hinting that it could have been a limiting factor.

Method		ΔW formulation	DINO	CLIP-I
LoRA [rank-r]		BA	0.674	0.785
↓ + normalize w/ controllable boundary		$\frac{\lambda}{r}B\Xi A$	0.682	0.809
· + normalize w/ controllable boundary + weight-scaling · + controllable boundary + high rank + relaxed + additive FT	**(DeLoRA)**	$\frac{\lVert W\rVert\lambda}{r}B\Xi A$	**0.701**	0.825
↑ + controllable scale + high rank + relaxed		$\frac{\lambda}{r}(B\Xi A - D\Phi C)W$	<u>0.696</u>	<u>0.833</u>
| + controllable boundary + high rank		$\frac{\lambda}{r}(U\Sigma U^\intercal - V\Theta V^\intercal)W$	0.685	**0.840**
| + controllable boundary		$\lambda(uu^\intercal - vv^\intercal)W$	0.678	0.810
ETHER+ (one-side) [rank-2, boundary equal to 2]		$(uu^\intercal - vv^\intercal)W$	0.624	0.746

Table 1: Ablation of DeLoRA innovations on the **Subject-driven Image Generation** task. We show how different components affect performance from both LoRA and ETHER derivation.

Method		ΔW Formulation	mIoU ↑	Acc. ↑	FID ↓
LoRA [rank-r]		BA	25.13	64.95	31.35
↓ + normalize w/ controllable boundary		$\frac{\lambda}{r}B\Xi A$	<u>25.66</u>	**65.82**	31.01
· + normalize w/ controllable boundary + weight-scaling · + controllable boundary + high rank + relaxed + additive FT	**(DeLoRA)**	$\frac{\lVert W\rVert\lambda}{r}B\Xi A$	**26.10**	65.08	<u>30.71</u>
↑ + controllable boundary + high rank + relaxed		$\frac{\lambda}{r}(B\Xi A - D\Phi C)W$	25.55	<u>65.16</u>	**29.89**
| + controllable boundary		$\lambda(uu^\intercal - vv^\intercal)W$	24.56	62.70	31.28
ETHER+ (one-side) [rank-2, boundary equal to 2]		$(uu^\intercal - vv^\intercal)W$	23.46	62.26	31.18

Table 2: Ablation of DeLoRA innovations on the **Semantic Map to Image** task. We show how different components from both LoRA and ETHER derivations incrementally improve performance.

3.3 Benchmark Results

Subject-Driven Image Generation Results are in Table 3. For full benchmark performance comparisons, we report low-rank results from Bini et al. (2024), while we run and compare LoRA, DoRA, and DeLoRA methods with same rank. For each method, we run a grid search to find best hyperparameters on the same 3 subjects used for the ablations, then we evaluate the best methods on the full 30 subjects benchmark, evaluating each method on the same 3 different seeds. Best and average results are reported in Table 3. We notice that LoRA, DoRA, and DeLoRA, all achieve comparable average performance in terms of DINO and CLIP-Image, and they all surpass lower-rank baselines. Therefore, DeLoRA effectively brings ETHER+ robustness properties while achieving superior performance.

Natural Language Understanding Results are in Table 4. For proper evaluation on the GLUE validation set, we follow Wu et al. (2024a;b) and split the validation set into two subsets (determined by pre-defined seeds), and use the first subset to tune hyperparameters, and the second subset to evaluate method performance. For fair comparisons we use same seeds as Wu et al. (2024a;b). In addition, in order to compare with LoRA's implementation, we simply apply DeLoRA to Q,V

Method		#param	DINO	CLIP-I
Real Images			0.703	0.864
DreamBooth	(Ruiz et al., 2023)	859.5M	0.644	0.793
$\text{OFT}_{n=4}$	(Qiu et al., 2023)	11.6M	0.652	0.794
ETHER+	(Bini et al., 2024)	0.4M	0.666	0.800
$\text{LoRA}_{r=4}$	(Hu et al., 2022)	0.8M	0.660	0.796
$\text{LoRA}_{r=16}$	(Hu et al., 2022)	3.2M	<u>0.686</u>	0.818
$\text{DoRA}_{r=16}$	(Liu et al., 2024a)	3.2M	**0.687**	<u>0.819</u>
$\text{DeLoRA}_{r=16}$	(ours)	3.2M	<u>0.686</u>	**0.820**
$\text{LoRA}^{\dagger}_{r=16}$	(Hu et al., 2022)	3.2M	0.688	0.818
$\text{DoRA}^{\dagger}_{r=16}$	(Liu et al., 2024a)	3.2M	<u>0.689</u>	<u>0.819</u>
$\text{DeLoRA}^{\dagger}_{r=16}$	(ours)	3.2M	**0.693**	**0.820**

Table 3: Results for evaluating DeLoRA in **subject-driven image generation**. † indicates experiments with tuned hyperparameters.

Method		#param	MNLI	SST-2	MRPC	CoLA	QNLI	QQP	RTE	STS-B	Avg
Full Finet.		125M	87.3	94.4	87.9	62.4	92.5	91.7	78.3	90.6	85.6
BitFit	(Zaken et al. (2022))	0.1M	84.7	**94.0**	88.1	54.0	91.0	87.3	69.8	89.5	82.3
IA3	(Liu et al. (2022))	0.06M	85.4	93.4	86.4	57.8	91.1	88.5	73.5	88.5	83.1
LoReFT	(Wu et al. (2024b))	0.02M	83.1	93.4	**89.2**	60.4	91.2	87.4	**79.0**	90.0	84.2
RED	(Wu et al. (2024a))	0.02M	83.9	93.9	**89.2**	61.0	90.7	87.2	78.0	90.4	84.3
LoRA	(Hu et al. (2022))	0.3M	86.6	93.9	88.7	59.7	**92.6**	90.4	75.3	90.3	84.7
Adapter$^{\mathrm{FFN}}$	(Pfeiffer et al. (2021))	0.3M	**87.1**	93.0	88.8	58.5	92.0	90.2	77.7	90.4	84.7
Adapter	(Houlsby et al. (2019))	0.4M	87.0	93.3	88.4	60.9	92.5	**90.5**	76.5	90.5	85.0
DeLoRA	(ours)	0.3M	86.9	93.7	88.6	**64.7**	**92.6**	90.2	77.3	**90.6**	**85.6**

Table 4: Comparisons of different methods finetuning RoBERTa-base on **GLUE benchmark**. Results of all baselines are taken from Wu et al. (2024a) and Wu et al. (2024b).

Method		#param	MMLU	ARC	Tru-1	Tru-2	Avg
LLaMA-2-7B		-	41.81	42.92	25.21	38.95	37.22
ETHER$_{n=32}$	(Bini et al. (2024))	0.26M	44.57	45.14	27.91	41.83	39.86
ETHER+$_{n=32}$	(Bini et al. (2024))	1.04M	**44.87**	46.50	29.38	43.51	41.07
LoRA$_{r=8}$	(Hu et al. (2022))	4.19M	43.61	46.16	28.76	42.21	40.19
DoRA$_{r=8}$	(Liu et al. (2024a))	4.19M	43.24	47.18	29.01	43.47	40.73
DeLoRA$_{r=8}$	(ours)	4.19M	44.21	**47.70**	**29.62**	**44.14**	**41.42**

Table 5: Results for **Instruction Tuning** on MMLU, ARC, and TruthfulQA benchmarks. Values represent accuracy scores achieved by different finetuning methods. Best scores are highlighted in bold, and second-best scores are underlined.

attention layers with rank 8, which is likely sub-optimal with respect to applying lower-rank modules to a larger set of layers Hu et al. (2022). We notice how DeLoRA achieves better performance on CoLA, QNLI and STS-B, and an overall significantly better average score with respect to all baselines, demonstrating its efficacy in adapting language models for NLU tasks.

Instruction Tuning Results are in Table 5. There, we can see that DeLoRA achieves the best result on three out of four tasks and surpasses all other methods in average scores. In particular, DeLoRA also achieves better results than DoRA, which already compares favorably with LoRA. This confirms the effectiveness of our improvements compared to LoRA and also DoRA, although DoRA is motivated by similar considerations to DeLoRA. However, on the MMLU task DeLoRA, while still achieving better finetuning performance than LoRA or DoRA, does not surpass the performance of ETHER and ETHER+. However, we note that MMLU overall is the hardest task to finetune, as even the best method, ETHER+, only achieves a 3 point advantage over the pretrained LLaMA-2-7B model. Possible improvements are larger on all other tasks. Therefore, we hypothesize that on MMLU, the increased robustness of ETHER has the advantage over the increased flexibility of other methods since, in no case, can large improvements be achieved.

3.4 Insights

In this section we analyze (i) the robustness properties, and (ii) the training dynamics, with a focus on prolonged training setting, of DeLoRA with respect to other finetuning methods. Then, we analyze (iii) how weights' norm differ on a pretrained model, to better understand the weights-norm scaling effect in DeLoRA.

Robustness Analysis. We conducted a comprehensive learning rate robustness analysis in the setting of the Subject-driven Generation task of Section 3. Evaluation is done reporting DINO scores (Fig.2, Left) and Euclidean distance between finetuned and pretrained weights of a projection layer in an attention module (Fig.2, Right) across multiple methods, using a range of learning rates derived from each method's base learning rate. Our analysis shows that DeLoRA is able to achieve the same robustness of ETHER+, while improving performance, while both LoRA and DoRA performance degrade at $4\times$ the base learning rate. We also notice how LoRA updates' distance grows at higher

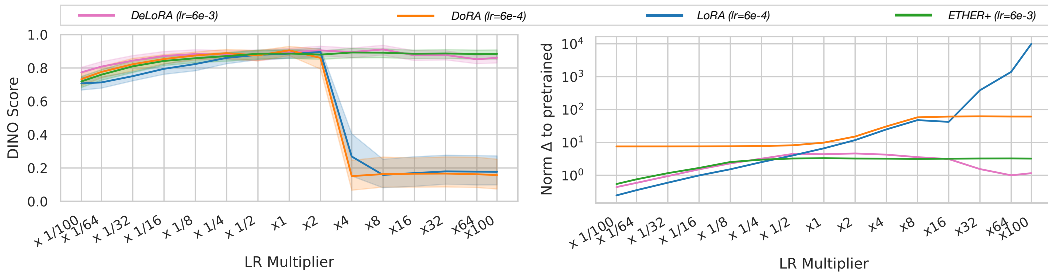

Figure 2: Learning rate robustness plots in Subject-driven generation task in terms of DINO scores (Left) and Euclidean distance between a finetuned vs pretrained projection layer weights (Right). Learning rates used for robustness evaluation were derived by multiplying the base learning rate in a range of factors.

Figure 3: (Left) Euclidean Distance of finetuned weights to pretrained weights as a function of the number of training steps. (Right) Qualitative examples show that LoRA exhibits significant artifacts earlier in the process compared to DeLoRA, which maintains better image quality.

learning rates, while interestingly DoRA, after $8\times$, does not diverge further, likely thanks to its magnitude control. However this does not lead to better performance in these regimes.

Finetuning Regime. We further study the behavior of weight updates, by measuring the Euclidean distance of a finetuned weight matrix (i.e. after merging) to the pretrained weight matrix during finetuning. This gives us a measure of how much and how fast the finetuned weight matrix diverts from the pretrained weights. In Fig. 3, we show this analysis for the out-projection matrix in one of StableDiffusion's Unet self-attention layers. We find that LoRA- and DoRA-trained weights continuously depart from the pretrained weights over the course of training, passing through an optimal regime but eventually overshooting and ending in a diverging regime. In contrast, DeLoRA-trained weights move away from the pretrained weights very quickly during the first stage of training, but are then bounded by DeLoRA's intrinsic boundary. This boundary effectively allows for the usage of high learning rates without the risk of divergence.

Weights Norms Analysis. In Fig. 4, we show the mean of column norms for weight matrices in different attention blocks of the Unet in Stable Diffusion v1.5. By doing so, we highlight the effect of weight-scaling as introduced in Section 2. We find that different modules, as well as different positions in the Unet, show systematic differences w.r.t. weight norms. This points at differences within the pretrained model which finetuning methods should account for. Our proposed scaling is one possibility to accomplish this. Exploring more sophisticated methods to include layer-wise differences is an interesting direction for future research.

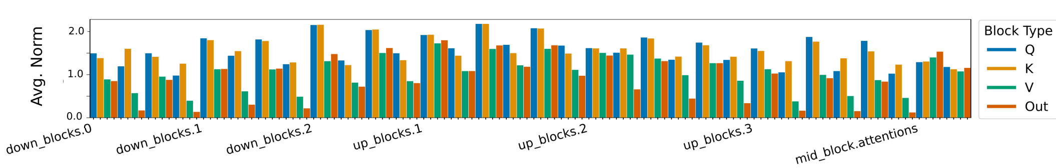

Figure 4: Average column norms of parameters in the attention modules of Stable Diffusion's Unet

4 Related Work

Parameter efficient finetuning (PEFT) is an active field of research, encompassing methods such as adapters (Houlsby et al., 2019), prompt- and prefix-tuning variations (Lester et al., 2021; Li & Liang, 2021; Liu et al., 2023), and more specialized methods such as BitFit (Zaken et al., 2022), FourierFT (Gao et al., 2024), and LayerNorm Tuning (Zhao et al., 2024). In this paper, we propose an improved PEFT method based on low-rank adapters (LoRA) first described by (Hu et al., 2022). Therefore, we focus our review of previous work on LoRA variants and refer to recent surveys (Han et al., 2024; Xin et al., 2024) regarding PEFT methods in general. LoRA is a popular finetuning approach for large models, featuring advantages such as low-memory footprint and no additional inference cost (Hu et al., 2022). Compared to full-finetuning, LoRA is also less prone to catastrophic forgetting (Biderman et al., 2024).

However, beyond falling behind in performance on downstream tasks compared to full finetuning (Biderman et al., 2024), previous work has identified and attempted to address different limitations of the original LoRA method. Lialin et al. (2023); Zi et al. (2023); Xia et al. (2024); Ren et al. (2024) propose methods to overcome the low-rank limitation without sacrificing memory efficiency. Similarly, VeRA (Kopiczko et al., 2024) keeps the original LoRA setup but reduces trainable parameters further by only scaling the randomly initialized matrices, which are shared across layers. To account for differences between layers, (Zhang et al., 2023b; Ding et al., 2023; Zhang et al., 2024; Liu et al., 2024c) describe methods to dynamically adapt the rank of different LoRA adapters. Instead of changing the rank, in this work, we propose to dynamically change the scaling of LoRA matrices for different layers, highlighting the need for layer-adaptive methods. PiSSA (Meng et al., 2024) and MiLoRA (Wang et al., 2024) show how improved initialization of LoRA can lead to better performance and faster convergence. Zhu et al. (2024) and Hayou et al. (2024) show that LoRA matrices behave differently in terms of optimal initialization and learning rate. Our work is complementary to these findings, as we also argue for different treatments of LoRAs, but regarding different layers within a model, not within the same adapter. DoRA (Liu et al., 2024a), like our work, proposes to stabilize LoRA training by normalizing and scaling the weights, however they normalize the full updated weight matrix $W + \Delta W$ on the column space, controlling each singular column of the finetuned matrices, whereas we normalize the inner r-dimensional space of each ΔW update matrix.

5 Conclusions

In this work, we propose a novel parameter efficient finetuning method, DeLoRA, which combines the strength of LoRA and ETHER to address their respective individual limitations. We demonstrate that using DeLoRA for finetuning yields improved results compared to LoRA, DoRA, and ETHER on two tasks in image-generation and LLM adaptation. Beyond showing the strong performance of DeLoRA, we provide detailed insights into its derivation, motivating our method both from the perspective of LoRA and ETHER. Furthermore, we ablate the contribution of individual innovations and find that bounding the update strength is key to improved performance. Finally, we also analyze the hyperparameter robustness of DeLoRA and find that it yields strong results for a wide range of learning rates as well as training steps. These insights provide valuable perspectives on how to improve existing PEFT methods and overcome their limitations.

Reproducibility Statement

To facilitate deployment and further research on DeLoRA, we will release our code upon acceptance. The code also includes the implementation of all benchmarks in this study, as well as ablation studies and hyperparameter choices.

REFERENCES

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John Patrick Cunningham. LoRA learns less and forgets less. In *TMLR*, 2024.

Massimo Bini, Karsten Roth, Zeynep Akata, and Anna Khoreva. ETHER: Efficient finetuning of large-scale models with hyperplane reflections. In *ICML*, 2024.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *CVPR*, 2021.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. In *arXiv*, 2018.

Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. Sparse low-rank adaptation of pre-trained language models. In *EMNLP*, 2023.

Ziqi Gao, Qichao Wang, Aochuan Chen, Zijing Liu, Bingzhe Wu, Liang Chen, and Jia Li. Parameter-efficient fine-tuning with discrete fourier transform. In *ICML*, 2024.

Zeyu Han, Chao Gao, Jinyang Liu, Sai Qian Zhang, et al. Parameter-efficient fine-tuning for large models: A comprehensive survey. In *arXiv*, 2024.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. The impact of initialization on lora finetuning dynamics. In *arXiv*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *ICLR*, 2021.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *ICML*, 2019.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki M Asano. VeRA: Vector-based random matrix adaptation. In *The Twelfth International Conference on Learning Representations*, 2024.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *EMNLP*, 2021.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL*, 2021.

Vladislav Lialin, Sherin Muckatira, Namrata Shivagunde, and Anna Rumshisky. Relora: High-rank training through low-rank updates. In *ICML*, 2023.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *ACL*, 2022.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning, August 2022. URL `http://arxiv.org/abs/2205.05638`. arXiv:2205.05638 [cs].

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. In *ICML*, 2024a.

Weiyang Liu, Zeju Qiu, Yao Feng, Yuliang Xiu, Yuxuan Xue, Longhui Yu, Haiwen Feng, Zhen Liu, Juyeon Heo, Songyou Peng, Yandong Wen, Michael J. Black, Adrian Weller, and Bernhard Schölkopf. Parameter-efficient orthogonal finetuning via butterfly factorization. In *ICLR*, 2024b.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. In *AI Open*, 2023.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Ro{bert}a: A robustly optimized {bert} pre-training approach, 2020. URL https://openreview.net/forum?id=SyxS0T4tvS.

Zequan Liu, Jiawen Lyn, Wei Zhu, and Xing Tian. Alora: Allocating low-rank adaptation for fine-tuning large language models. In *NAACL*, 2024c.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. In *arXiv*, 2024.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-Destructive Task Composition for Transfer Learning. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 487–503, 2021. doi: 10.18653/v1/2021.eacl-main.39. URL https://aclanthology.org/2021.eacl-main.39. Conference Name: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume Place: Online Publisher: Association for Computational Linguistics.

Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. Controlling text-to-image diffusion by orthogonal finetuning. In *NeurIPS*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Pengjie Ren, Chengshun Shi, Shiguang Wu, Mengqi Zhang, Zhaochun Ren, Maarten Rijke, Zhumin Chen, and Jiahuan Pei. Melora: Mini-ensemble low-rank adapters for parameter-efficient fine-tuning. In *ACL*, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, 2023.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. In *arXiv*, 2023b.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2018.

Hanqing Wang, Zeguan Xiao, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. Milora: Harnessing minor singular components for parameter-efficient llm finetuning. In *arXiv*, 2024.

Muling Wu, Wenhao Liu, Xiaohua Wang, Tianlong Li, Changze Lv, Zixuan Ling, Jianhao Zhu, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. Advancing parameter efficiency in fine-tuning via representation editing, 2024a. URL https://arxiv.org/abs/2402.15179.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. Reft: Representation finetuning for language models, 2024b. URL https://arxiv.org/abs/2404.03592.

Wenhan Xia, Chengwei Qin, and Elad Hazan. Chain of lora: Efficient fine-tuning of language models via residual learning. In *arXiv*, 2024.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018.

Yi Xin, Siqi Luo, Haodi Zhou, Junlong Du, Xiaohong Liu, Yue Fan, Qing Li, and Yuntao Du. Parameter-efficient fine-tuning for pre-trained vision models: A survey. In *arXiv*, 2024.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *ACL*, 2022.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *CVPR*, 2023a.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *ICLR*, 2023b.

Ruiyi Zhang, Rushi Qiang, Sai Ashish Somayajula, and Pengtao Xie. Autolora: Automatically tuning matrix ranks in low-rank adaptation based on meta learning. In *NAACL*, 2024.

Bingchen Zhao, Haoqin Tu, Chen Wei, Jieru Mei, and Cihang Xie. Tuning layernorm in attention: Towards efficient multi-modal llm finetuning. In *ICLR*, 2024.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. In *International Journal of Computer Vision*, 2019.

Jiacheng Zhu, Kristjan Greenewald, Kimia Nadjahi, Haitz Sáez de Ocáriz Borde, Rickard Brüel Gabrielsson, Leshem Choshen, Marzyeh Ghassemi, Mikhail Yurochkin, and Justin Solomon. Asymmetry in low-rank adapters of foundation models. In *ICML*, 2024.

Bojia Zi, Xianbiao Qi, Lingzhi Wang, Jianan Wang, Kam-Fai Wong, and Lei Zhang. Delta-lora: Fine-tuning high-rank parameters with the delta of low-rank matrices. In *arXiv*, 2023.

A ETHER AND ETHER+ LOW-RANK LIMITATION

In ETHER and ETHER+, even if the transformation matrices are full-rank, the weight updates to the pretrained model are intrinsically limited to be low-rank. To show this, we can rewrite the applied transformations in a residual form. For ETHER the matrix multiplication can be written as:

$$HW = W - 2uu^\intercal W$$

where the right-hand side, by multiplying the pretrained with a rank-1 transformation, restricts the weight updates to be rank-1.

While for ETHER+:

$$\begin{aligned}
&H^{+}W\tilde{H}^{+} \\
&= (W - uu^\intercal W + vv^\intercal W)\tilde{H}^{+} \\
&= W - uu^\intercal W + vv^\intercal W - (W - uu^\intercal W + vv^\intercal W)\tilde{u}\tilde{u}^\intercal + (W - uu^\intercal W + vv^\intercal W)\tilde{v}\tilde{v}^\intercal
\end{aligned}$$

where the rank-1 residual matrices on the right-hand side will bring the updates to be rank-4.

As a side note, we think that the higher rank of ETHER+ updates further explains the better performance of ETHER+ with respect to ETHER. However, one downside of ETHER and ETHER+ finetuning is the inability to easily control the rank of these updates, differently from LoRA.

B EXPERIMENTAL DETAILS

In this section we report further details about our experiments, along with standard deviation results.

Subject-Driven Generation. We used the first 3 subjects (10% of the data) to select best hyperparameter for each method among LoRA, DoRA and DeLoRA with rank 16. Then, we used best hyperparameters to evaluate each method on all 30 subjects, for 3 different seeds. Results with standard deviations are reported in Table 6.

Method		DINO	CLIP-I
LoRA$_{r=16}$	(Hu et al., 2022)	<u>0.686</u>$_{\pm.0012}$	0.818$_{\pm.0017}$
DoRA$_{r=16}$	(Liu et al., 2024a)	**0.687**$_{\pm.0015}$	<u>0.819</u>$_{\pm.0015}$
DeLoRA$_{r=16}$	(ours)	<u>0.686</u>$_{\pm.0056}$	**0.820**$_{\pm.0027}$

Table 6: Results for evaluating DeLoRA in subject-driven image generation. Best scores are highlighted in bold, and second-best scores are underlined.

GLUE. Following Wu et al. (2024b), for each benchmark task, we split the publicly available validation set in two subsets. We then use one subset to tune the hyperparameters on seed 42. Best hyperparameters are then used to evaluate test performance for seeds 42, 43, 44, 45, 46. We highlight that with respect to Wu et al. (2024b), we don't discard any underperforming seed.

Experiments with standard deviation details are in 7.

	#param	MNLI	SST-2	MRPC	CoLA	QNLI	QQP	RTE	STS-B	Avg
Full Finet.	125M	87.3$_{\pm.34}$	94.4$_{\pm.96}$	87.9$_{\pm.91}$	62.4$_{\pm3.29}$	92.5$_{\pm.22}$	91.7$_{\pm.19}$	78.3$_{\pm3.20}$	90.6$_{\pm.59}$	85.6
BitFit	0.1M	84.7$_{\pm.08}$	**94.0**$_{\pm.87}$	88.1$_{\pm1.57}$	54.0$_{\pm3.07}$	91.0$_{\pm.05}$	87.3$_{\pm.02}$	69.8$_{\pm1.51}$	89.5$_{\pm.35}$	82.3
IA3	0.06M	85.4$_{\pm-}$	93.4$_{\pm-}$	86.4$_{\pm-}$	57.8$_{\pm-}$	91.1$_{\pm-}$	88.5$_{\pm-}$	73.5$_{\pm-}$	88.5$_{\pm-}$	83.1
LoReFT	0.02M	83.1$_{\pm.26}$	93.4$_{\pm.64}$	**89.2**$_{\pm2.62}$	60.4$_{\pm2.60}$	91.2$_{\pm.25}$	87.4$_{\pm.23}$	**79.0**$_{\pm2.76}$	90.0$_{\pm.29}$	84.2
RED	0.02M	83.9$_{\pm.14}$	93.9$_{\pm.31}$	**89.2**$_{\pm.98}$	61.0$_{\pm2.96}$	90.7$_{\pm.35}$	87.2$_{\pm.17}$	78.0$_{\pm2.06}$	90.4$_{\pm.32}$	84.3
LoRA	0.3M	86.6$_{\pm.23}$	93.9$_{\pm.49}$	88.7$_{\pm.76}$	59.7$_{\pm4.36}$	**92.6**$_{\pm.10}$	90.4$_{\pm.08}$	75.3$_{\pm2.79}$	90.3$_{\pm.54}$	84.7
Adapter$^{\mathrm{FFN}}$	0.3M	**87.1**$_{\pm.10}$	93.0$_{\pm.05}$	88.8$_{\pm1.38}$	58.5$_{\pm1.69}$	92.0$_{\pm.28}$	90.2$_{\pm.07}$	77.7$_{\pm1.93}$	90.4$_{\pm.31}$	84.7
Adapter	0.4M	87.0$_{\pm.28}$	93.3$_{\pm.40}$	88.4$_{\pm1.54}$	60.9$_{\pm3.09}$	92.5$_{\pm.02}$	**90.5**$_{\pm.08}$	76.5$_{\pm2.26}$	90.5$_{\pm.35}$	85.0
DeLoRA(ours)	0.3M	86.9$_{\pm.21}$	93.7$_{\pm.79}$	88.6$_{\pm1.49}$	**64.7**$_{\pm2.33}$	**92.6**$_{\pm.53}$	90.2$_{\pm.17}$	77.3$_{\pm1.96}$	**90.6**$_{\pm.38}$	**85.6**

Table 7: **GLUE benchmark.** Comparisons of different methods finetuning RoBERTa-base. Results of all baselines are taken from Wu et al. (2024a) and Wu et al. (2024b).

C FIXING THE MAGNITUDE TERM IN DORA

In the following section we provide preliminaries experiments testing if fixing the magnitude in DoRA could lead to similar robustness properties as DeLoRA.

Performance. We first evaluate if fixing the magnitude term could be detrimental in terms of performance. Following the setting of our small-scale ablation in Section 3.2, we run a small scale experiment comparing DoRA variations, along with DeLoRA.

Method	DINO	CLIP-I
DoRA$_{r=16}$(fixed magnitude)	0.681	0.822
DoRA$_{r=16}$	0.683	0.820
DeLoRA$_{r=16}$	0.701	0.825

Table 8: Subject-driven Image Generation small-scale ablation

We notice how DoRA results without updating the magnitude term seem to lead to only slightly underperforming results with respect to standard DoRA.

Robustness. We then run the same robustness analysis as reported in Fig. 2. We see how fixing the magnitude term does not lead to a behavior similar to DeLoRA, but rather still follows DoRA behavior.

Plots in Fig. 5 show that simply fixing the magnitude term does not alter DoRA robustness properties (Fig. 5, Left), while actually in higher learning rate regimes seems to lead to further divergence (Fig. 5 Right), not allowing the magnitude to counterbalance the divergent trend. This behavior suggests that keeping column norms constant might not be restrictive enough. In this regard, DeLoRA inner normalization in terms of Frobenius distance seems to be a more promising strategy to avoid model divergence.

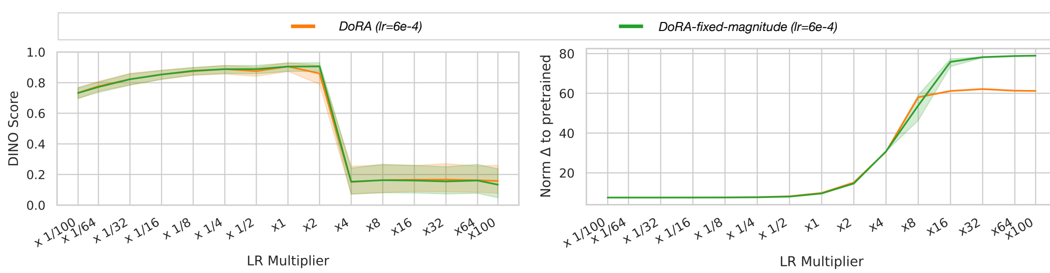

Figure 5: Robustness analysis between DoRA with and without magnitude updates, with respect to learning rate changes from the optimal learning rate.

D Robustness Ablation on DeLoRA's boundary and angles

We additionally conducted an ablation on DeLoRA's setting, where we run the same robustness analysis of Section 3.4 by varying the learning rate of the scaling term λ (affecting the boundary), and the weights BA (angular component). We notice how all methods lead to convergence, additionally demonstrating DeLoRA's robustness properties.

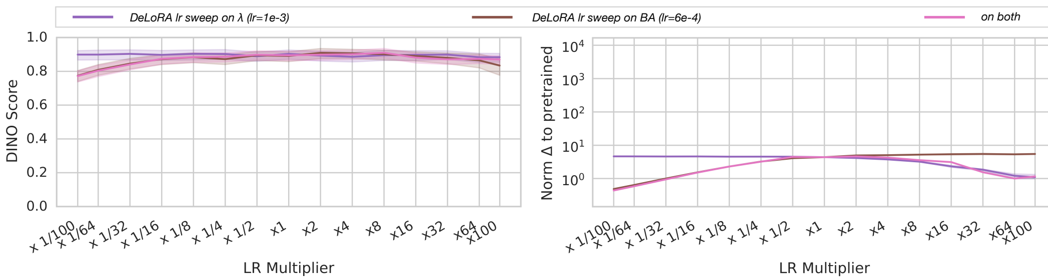

Figure 6: Learning rate robustness plots for DeLoRA in Subject-driven generation task in terms of DINO scores (Left) and Euclidean distance finetuned vs pretrained weights of a projection layer (Right). Ablation testing impact of increasing learning rate for boundary (λ) or angular weights (BA).

E Qualitative Examples

We report in E qualitative examples generated by our propopsed DeLORA finetuning Stable Diffusion for the tasks of Subject-driven Generation and Semantic Map to Image. While in E we report qualitative examples of prolonged genearation with DeLoRA, LoRA and DoRA methods.

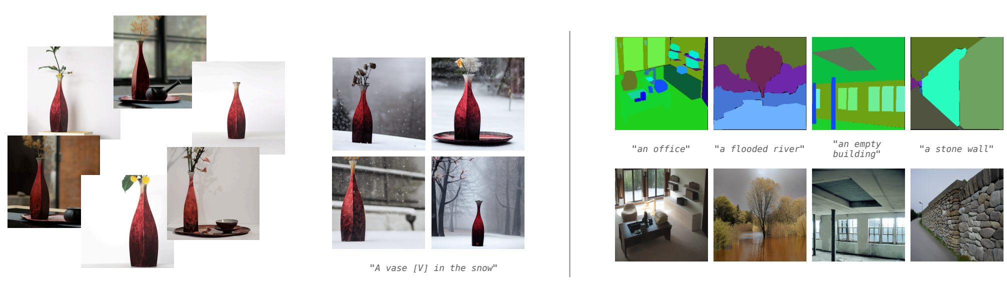

Figure 7: Examples generated by DeLoRA-finetuned Stable Diffusion for personalized generation on a small set of subject-specific images (left), and for semantic map to image on ADE20K (right).

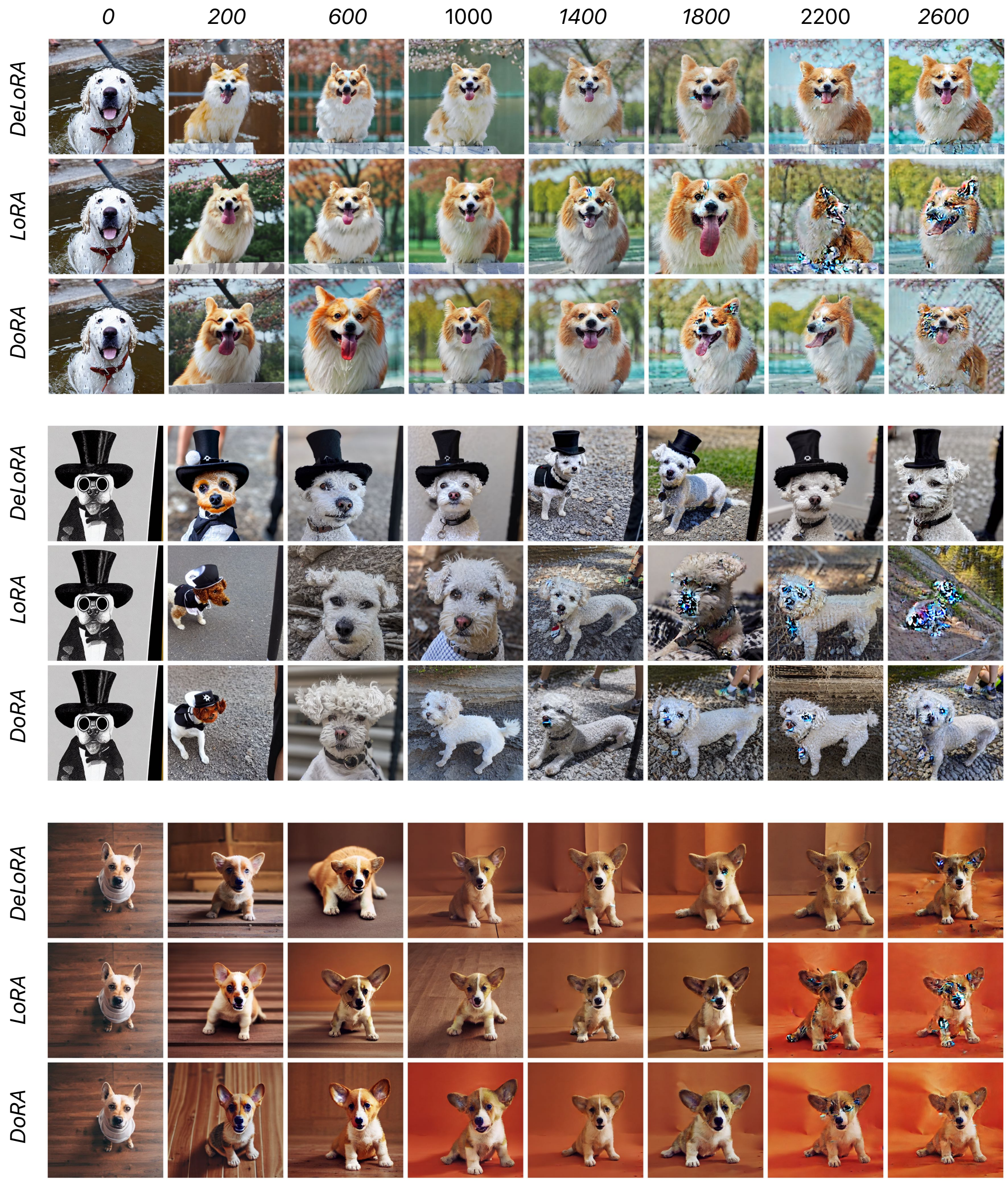

Figure 8: Prolonged finetuning generated examples generated by DeLoRA, LoRA, and DoRA methods, up to time step 2600.