DoRA: Weight-Decomposed Low-Rank Adaptation

Shih-Yang Liu [1 2] **Chien-Yi Wang** [1] **Hongxu Yin** [1] **Pavlo Molchanov** [1] **Yu-Chiang Frank Wang** [1]
Kwang-Ting Cheng [2] **Min-Hung Chen** [1]

Abstract

Among the widely used parameter-efficient fine-tuning (PEFT) methods, LoRA and its variants have gained considerable popularity because of avoiding additional inference costs. However, there still often exists an accuracy gap between these methods and full fine-tuning (FT). In this work, we first introduce a novel weight decomposition analysis to investigate the inherent differences between FT and LoRA. Aiming to resemble the learning capacity of FT from the findings, we propose Weight-**D**ecomposed **L**o**w**-**R**ank **A**daptation (**DoRA**). DoRA decomposes the pre-trained weight into two components, *magnitude* and *direction*, for fine-tuning, specifically employing LoRA for directional updates to efficiently minimize the number of trainable parameters. By employing DoRA, we enhance both the learning capacity and training stability of LoRA while avoiding any additional inference overhead. DoRA consistently outperforms LoRA on fine-tuning LLaMA, LLaVA, and VL-BART on various downstream tasks, such as commonsense reasoning, visual instruction tuning, and image/video-text understanding. Code is available at https://github.com/NVlabs/DoRA.

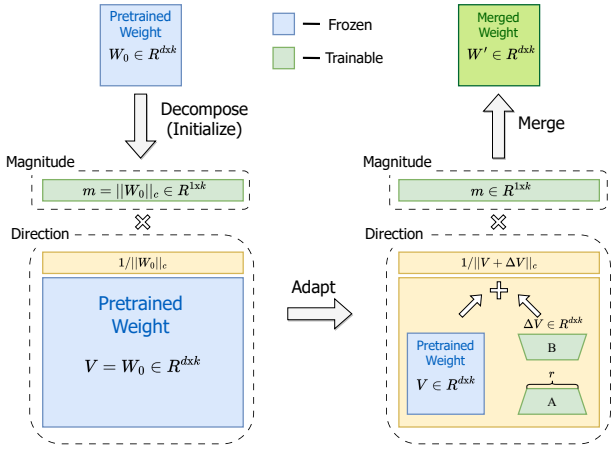

Figure 1. An overview of our proposed DoRA, which decomposes the pre-trained weight into *magnitude* and *direction* components for fine-tuning, especially with LoRA to efficiently update the direction component. Note that $||\cdot||_c$ denotes the vector-wise norm of a matrix across each column vector.

1. Introduction

Models that are pre-trained with extensive general domain datasets have demonstrated remarkable generalization abilities, significantly benefiting a wide array of applications, from natural language processing (NLP) tasks (Qin et al., 2023; Taori et al., 2023) to multi-modal tasks (Li et al., 2022; Liu et al., 2023a). To tailor these general models for specific downstream tasks, *full fine-tuning (FT)* is commonly employed, involving the retraining of all model parameters. Nevertheless, as the size of models and datasets expand in scale, the expense associated with fine-tuning the entire model becomes prohibitively large.

To address this issue, parameter-efficient fine-tuning (PEFT) methods (Houlsby et al., 2019) have been introduced to fine-tune the pre-trained models with only a minimal number of parameters. Among these, LoRA (Hu et al., 2022), which does not change the model architecture, has become notably popular for its simplicity and efficacy. Nevertheless, there is still a capacity gap between LoRA and FT, which is often attributed to the limited number of trainable parameters without further exploration of other underlying causes (Hu et al., 2022; Kopiczko et al., 2024).

Drawing on Weight Normalization (Salimans & Kingma, 2016), which achieves faster convergence via improving the conditioning of the gradient with weight reparameterization, we introduce a novel weight decomposition analysis that initially reparameterizes model weights into magnitude and directional components, subsequently examining the changes

[1]NVIDIA [2]HKUST. Correspondence to: Shih-Yang Liu <shihyangl@nvidia.com, sliuau@connect.ust.hk>, Min-Hung Chen <minhungc@nvidia.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

in magnitude and direction introduced by LoRA and FT. Our analysis reveals that LoRA and FT exhibit markedly distinct patterns of updates, leading us to surmise that these variations mirror the learning capability of each method. Inspired by our findings, we propose Weight-**D**ecomposed **L**ow-**R**ank **A**daptation (**DoRA**), which begins by decomposing the pre-trained weight into its magnitude and directional components, then fine-tunes both. Given the substantial size of the directional component in terms of parameters, we exploit LoRA for the directional adaptation to enable efficient fine-tuning, as illustrated in Figure.1. Moreover, by showing a learning behavior similar to FT both empirically and mathematically, suggesting a learning capacity closely resembling FT, we have validated DoRA across a wide variety of tasks, from NLP to Vision-Language, and over various backbones, including LLM and LVLM. The experimental results show that DoRA consistently outperforms LoRA without sacrificing inference efficiency, such as commonsense reasoning (**+3.7/+1.0** on LLaMA-7B/13B, **+2.9** on LLaMA2-7B, and **+4.4** on LLaMA3-8B), visual instruction tuning (**+0.6** on LLaVA-7B), and image/video-text understanding (**+0.9/+1.9** on VL-BART).

The summary of our contributions is as follows:

- We introduce DoRA, a novel PEFT method that incorporates weight decomposition, achieving a learning capacity closely resembling FT without any additional inference latency over LoRA.
- We introduce a novel weight decomposition analysis to uncover the fundamental differences in the learning patterns of FT and different PEFT methods.
- DoRA consistently surpasses LoRA on various tasks, from NLP to Vision-Language benchmarks and across various backbones, including LLM and LVLM.

2. Related Works

Parameter-Efficient Fine-Tuning (PEFT) methods are designed to reduce the high expense of fine-tuning large-scale models. They achieve this by training a relatively small subset of parameters, compared to the total number of parameters, for adapting to downstream tasks. Existing PEFT methods can be divided into three categories. The first category is referred to as *Adapter-based* methods, which involve introducing additional trainable modules into the original frozen backbone, such as (Houlsby et al., 2019; He et al., 2021; Karimi Mahabadi et al., 2021; mahabadi et al., 2021). For example, (Houlsby et al., 2019) proposes adding linear modules in sequence to the existing layer, whereas (He et al., 2021) advocates for integrating these modules in parallel with the original layer to enhance performance. The second category is *Prompt-based* methods. These methods add extra soft tokens (prompts) to the initial input and focus solely on fine-tuning these trainable vectors, as seen in works like (Lester et al., 2021; Razdaibiedina et al., 2023; Wang et al., 2023). However, these approaches typically face challenges due to their sensitivity to initialization, affecting their overall effectiveness. These first two categories, whether altering the model's input or architecture, result in increased inference latency compared to the baseline model.

LoRA (Hu et al., 2022) and its variants are among the third category of PEFT, notable for not adding any extra inference burden. These methods apply low-rank matrices to approximate weight changes during fine-tuning and can merge with pre-trained weights prior to inference. For example, (Zhang et al., 2023) employs SVD decomposition and prunes less significant singular values for more efficient updates. (Hyeon-Woo et al., 2022) focuses on low-rank Hadamard product for federated learning. (Qiu et al., 2023; Liu et al., 2023b) exploit orthogonal factorization in fine-tuning diffusion models. (Renduchintala et al., 2023) uses weight tying to further reduce the trainable parameters. (Yeh et al., 2023) introduces a unified LoRA family framework for Stable diffusion. (Ponti et al., 2022) chooses different combinations of LoRAs from the inventory with a routing function for different tasks. (Kopiczko et al., 2024) implements learnable scaling vectors to adjust a shared pair of frozen random matrices across layers. Our research also falls within this third category, and we validate the efficacy of our proposed method alongside LoRA and its variants through comprehensive experimentation.

3. Pattern Analysis of LoRA and FT

3.1. Low-Rank Adaptation (LoRA)

Building upon the hypothesis that updates made during the fine-tuning exhibit a low "intrinsic rank", LoRA (Hu et al., 2022) proposes using the product of two low-rank matrices to update the pre-trained weights incrementally. For a pre-trained weight matrix $W_0 \in \mathbb{R}^{d\times k}$, LoRA models the weight update $\Delta W \in \mathbb{R}^{d\times k}$ utilizing a low-rank decomposition, expressed as BA, where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$ represent two low-rank matrices, with $r \ll min(d, k)$. Consequently, the fine-tuned weight W' can be represented as:

$$W' = W_0 + \Delta W = W_0 + \underline{BA} \tag{1}$$

where W_0 remains static during the fine-tuning process, and the underlined parameters are being trained. The matrix A is initialized with uniform Kaiming distribution (He et al., 2015), while B is initially set to zero, resulting in $\Delta W = BA$ being zero at the start of training. Notably, this decomposition of ΔW can be substituted with other LoRA variants, such as VeRA (Kopiczko et al., 2024). Additionally, based on Eq. (1), we can merge the learned ΔW with the pre-trained weight W_0 and obtain W' in advance

of deployment, and given that both W' and W_0 both fall within the dimensionality of $\mathbb{R}^{d\times k}$, LoRA and its related variants do not introduce any extra latency during the inference compared to the original model.

3.2. Weight Decomposition Analysis

The study presented in LoRA (Hu et al., 2022) suggests that LoRA can be considered a general approximation of full fine-tuning. By gradually increasing the rank r of LoRA to align with the rank of pre-trained weights, LoRA can attain a level of expressiveness akin to that of FT. Consequently, many previous studies have attributed the discrepancy in accuracy between LoRA and FT primarily to the limited number of trainable parameters, often without further analysis (Hu et al., 2022; Kopiczko et al., 2024). Drawing inspiration from Weight Normalization (Salimans & Kingma, 2016), which reparameterizes the weight matrix into magnitude and direction for accelerating optimization, we introduce an innovative weight decomposition analysis. Our analysis restructures the weight matrix into two separate components, *magnitude* and *direction*, to reveal the inherent differences in LoRA and FT learning patterns.

Analysis Method: This analysis examines the updates in both magnitude and direction of the LoRA and FT weights relative to the pre-trained weights to reveal the fundamental differences in the learning behaviors of both. The weight decomposition of $W \in \mathbb{R}^{d\times k}$ can be formulated as:

$$W = m\frac{V}{||V||_c} = ||W||_c\frac{W}{||W||_c} \tag{2}$$

where $m \in \mathbb{R}^{1\times k}$ is the magnitude vector, $V \in \mathbb{R}^{d\times k}$ is the directional matrix, with $||\cdot||_c$ being the vector-wise norm of a matrix across each column. This decomposition ensures that each column of $V/||V||_c$ remains a unit vector, and the corresponding scalar in m defines the magnitude of each vector.

For our weight decomposition analysis, we select the VL-BART model fine-tuned on four image-text tasks as outlined in (Sung et al., 2022) for a case study. Following (Sung et al., 2022), which applies LoRA only to the query/value weight matrix in the self-attention module. We decompose the pre-trained weight W_0, the full fine-tuned weight W_{FT}, and the merged LoRA weight W_{LoRA} of query/value weight matrix using Eq. (2). The magnitude and directional variations between W_0 and W_{FT} can be defined as follows:

$$\Delta M^t_{\text{FT}} = \frac{\sum_{n=1}^{k} |m^{n,t}_{\text{FT}} - m^n_0|}{k} \tag{3}$$

$$\Delta D^t_{\text{FT}} = \frac{\sum_{n=1}^{k} (1 - \mathbf{cos}(V^{n,t}_{\text{FT}}, W^n_0))}{k} \tag{4}$$

Here, ΔM^t_{FT} and and ΔD^t_{FT} represent the magnitude difference and directional difference between W_0 and W_{FT} at t training step respectively, with $\mathbf{cos}(\cdot,\cdot)$ being the cosine similarity function. $M^{n,t}_{\text{FT}}$ and M^n_0 are the nth scalars in their respective magnitude vectors, while $V^{n,t}_{\text{FT}}$ and W^n_0 are the nth columns in V^t_{FT} and W_0. The magnitude and directional differences between W_{LoRA} and W_0 are calculated similarly, as per Eq. (3) and Eq. (4). We select checkpoints from four different training steps for analysis, comprising three intermediate steps and the final checkpoint from both FT and LoRA, and we perform weight decomposition analysis on each of these checkpoints to determine the ΔM and ΔD throughout different layers.

Analysis Results: Figure 2 (a) and (b) illustrate the alterations in the query weight matrix of FT and LoRA, with each point representing a (ΔD^t, ΔM^t) pair from query weight matrices across different layers and training steps. Similarly, Figure 6 in the appendix displays the value weight matrix modifications. It is noticeable that LoRA exhibits a consistent positive slope trend across all the intermediate steps, signifying a proportional relationship between the changes in direction and magnitude. In contrast, the FT displays a more varied learning pattern with a relatively negative slope. This distinction between FT and LoRA likely mirrors their respective learning capability. While LoRA tends to either increase or decrease the magnitude and direction updates proportionally, it lacks the nuanced capability for more subtle adjustments. Specifically, LoRA does not show proficiency in executing slight directional changes alongside more significant magnitude alterations, or vice versa, a feature more characteristic of the FT method. We suspect that such limitation of LoRA might stem from the challenge of concurrent learning both magnitude and directional adaptation, which could be overly complex for LoRA. Consequently, in this work, we aim to propose a variant of LoRA that exhibits a learning pattern more closely resembling that of FT, and can improve the learning capacity over LoRA.

4. Method

4.1. Weight-Decomposed Low-Rank Adaptation

Drawing from the insights of our weight decomposition analysis, we introduce Weight-**D**ecomposed L**o**w-**R**ank **A**daptation (**DoRA**). DoRA initially decomposes the pre-trained weight into its magnitude and directional components and finetunes both of them. Because the directional component is large in terms of parameter numbers, we further decompose it with LoRA for efficient finetuning.

Our intuitions are two-fold. Firstly, we believe that limiting LoRA to concentrate exclusively on directional adaptation while also allowing the magnitude component to be tunable simplifies the task compared to the original approach, where LoRA is required to learn adjustments in both mag-

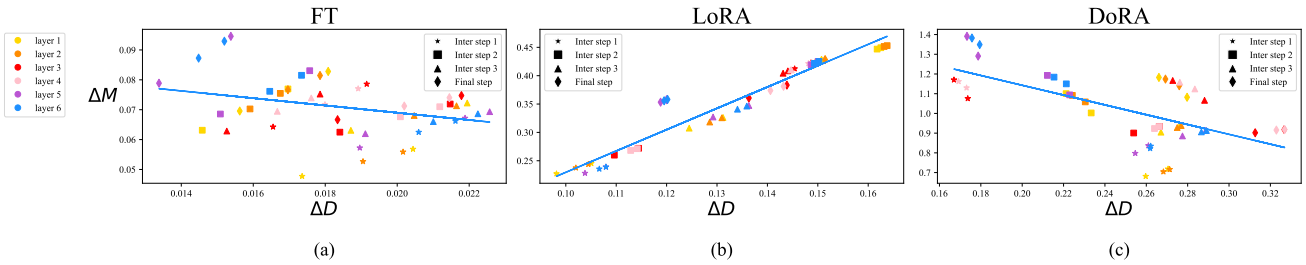

Figure 2. Magnitude and direction updates of (a) FT, (b) LoRA, and (c) DoRA of the query matrices across different layers and intermediate steps. Different markers represent matrices of different training steps and different colors represent the matrices of each layer.

nitude and direction. Secondly, the process of optimizing directional updates is made more stable through weight decomposition, which we delve into more thoroughly in Section.4.2. It is important to highlight that the main distinction between DoRA and weight normalization (Salimans & Kingma, 2016) lies in their training approaches. Weight normalization trains both components from scratch, making the method sensitive to different initializations. Conversely, DoRA avoids such initialization concerns since both components begin with pre-trained weights. We initialize DoRA with pre-trained weight W_0 as outlined in Eq. (2), where $m = ||W_0||_c$ and $V = W_0$ after initialization. We then keep V frozen and m a trainable vector. The directional component is then updated through LoRA. DoRA can be formulated similar to Eq. (1) as:

$$W' = \underline{m}\frac{V + \Delta V}{||V + \Delta V||_c} = \underline{m}\frac{W_0 + \underline{BA}}{||W_0 + \underline{BA}||_c} \tag{5}$$

where ΔV is the incremental directional update learned by multiplying two low-rank matrices B and A, and the underlined parameters denote the trainable parameters. The matrices $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$ are initialized in line with LoRA's strategy to ensure that W' equals W_0 before the finetuning. Furthermore, DoRA can be merged with the pre-trained weight before inference, thereby not introducing any additional latency.

We visualize the magnitude and directional differences of the query weight matrix between the merged DoRA weight and W_0 in the same setting as for FT and LoRA in Figure 2 (c) and leave the visualization of the value weight matrix in the appendix. From the regression line for $(\Delta D, \Delta M)$ of both DoRA and FT, we reveal that in contrast to LoRA's pattern, DoRA, and FT are characterized by a distinct negative slope. We reason that FT tends towards a negative slope because pre-trained weights already possess substantial knowledge suitable for various downstream tasks. Therefore, when provided with adequate learning capacity, having a larger magnitude or direction alteration alone is sufficient enough for downstream adaptation. We additionally compute the correlation between ΔD and ΔM for FT, LoRA, and DoRA, and we found that both FT and DoRA exhibit negative correlation values of -0.62 and -0.31, respectively. In contrast, LoRA shows a positive correlation with a value of 0.83. In conclusion, the fact that DoRA demonstrates the ability to make only substantial directional adjustments with relatively minimal changes in magnitude or the reverse while showing learning patterns closer to FT's signifies its superior learning capacity over LoRA.

4.2. Gradient Analysis of DoRA

In this section, we first derive the gradient of DoRA and illustrate how our proposed decomposition benefits the optimization of ΔV. Subsequently, we analyze from the gradient's perspective to explicate the learning pattern of DoRA, which tends to have a negative slope.

From Eq. (5), we can obtain the gradient of Loss $\mathcal{L}$ with respect to m and $V' = V + \Delta V$ as:

$$\nabla_{V'}\mathcal{L} = \frac{m}{||V'||_c}\left(I - \frac{V'V'^{\mathbf{T}}}{||V'||_c^2}\right)\nabla_{W'}\mathcal{L} \tag{6}$$

$$\nabla_{m}\mathcal{L} = \frac{\nabla_{W'}\mathcal{L}\cdot V'}{||V'||_c} \tag{7}$$

Eq. (6) reveals that the weight gradient $\nabla_{W'}\mathcal{L}$ is scaled by $m/||V'||_c$ and is projected away from the current weight matrix. These two effects contribute to aligning the gradient's covariance matrix more closely with the identity matrix, which is advantageous for optimization (Salimans & Kingma, 2016). Additionally, given that $V' = V + \Delta V$, the gradient $\nabla_{V'}L$ is equivalent to $\nabla_{\Delta V}L$. Therefore, the optimization benefits derived from this decomposition are fully transferred to ΔV, enhancing the learning stability of LoRA.

We can gain further insight into the learning pattern of DoRA by referring to Eq. (7). In the subsequent discussion, we represent vectors using lower-case letters instead of the previous matrix form notation. Consider $w'' = w' + \Delta w$ as the parameter update for a weight vector, where $\Delta w \propto \nabla_{w'}\mathcal{L}$. In two hypothetical update

scenarios, $S1$ and $S2$, $S1$ involves a smaller directional update (ΔD_{S1}), while $S2$ involves a larger one (ΔD_{S2}). Assuming $||\Delta w_{S1}|| = ||\Delta w_{S2}||$, and at time 0, we have $\Delta v = 0$ and $v' = v$. From $\Delta D_{S1} < \Delta D_{S2}$, it follows that $|\mathbf{cos}(\Delta w_{S1}, w')| > |\mathbf{cos}(\Delta w_{S2}, w')|$. Since $\Delta w \propto \nabla_{w'}\mathcal{L}$, it implies $|\mathbf{cos}(\nabla^{S1}_{w'}\mathcal{L}, w')| > |\mathbf{cos}(\nabla^{S2}_{w'}\mathcal{L}, w')|$. From Sec 4.1, with v initialized as v_0 and $w' = w_0$ at time 0, we get $|\mathbf{cos}(\nabla_{w'}\mathcal{L}, w')| = |\mathbf{cos}(\nabla_{w'}\mathcal{L}, v')| = |\mathbf{cos}(\nabla_{w'}\mathcal{L}, v)|$. Using the cosine similarity equation with $\Delta v = 0$:

$$cos(\nabla_{w'}\mathcal{L}, v') = cos(\nabla_{w'}\mathcal{L}, v) = \frac{\nabla_{w'}\mathcal{L} \cdot v}{||\nabla_{w'}\mathcal{L}||||v||} \tag{8}$$

denote m_* as the magnitude scalar of vector w' then Eq. (7) w.r.t m_* can be rewritten to:

$$\nabla_{m_*}\mathcal{L} = \frac{\nabla_{w'}\mathcal{L} \cdot v'}{||v'||} = ||\nabla_{w'}\mathcal{L}|| \cdot cos(\nabla_{w'}\mathcal{L}, v) \tag{9}$$

Given that $||\Delta w_{S1}|| = ||\Delta w_{S2}||$ for $S1$ and $S2$, and $||\nabla^{S1}_{w'}\mathcal{L}|| = ||\nabla^{S2}_{w'}\mathcal{L}||$. Therefore, with:

$$||\nabla^{S1}_{w'}\mathcal{L}|| \cdot |cos(\nabla^{S1}_{w'}\mathcal{L}, v)| > ||\nabla^{S2}_{w'}\mathcal{L}|| \cdot |cos(\nabla^{S2}_{w'}\mathcal{L}, v)| \tag{10}$$

it can be inferred that $|\nabla^{S1}_{m_*}\mathcal{L}| > |\nabla^{S2}_{m_*}\mathcal{L}|$ which indicate that $S1$ has larger magnitude updates over $S2$ while having smaller directional alteration than that of $S2$. Our conclusion generally holds in practice, as evidenced by Figure 2 (c). Consequently, we have effectively shown how DoRA can be utilized to adjust the learning pattern, diverging from that of LoRA and aligning more closely with the pattern of FT.

4.3. Reduction of Training Overhead

In Eq. (1), the gradients of W' and ΔW are the same. However, with DoRA, which redirects the low-rank adaptation towards the directional component, the gradient of the low-rank updates differs from that of W', as illustrated in Eq. (6). This divergence necessitates extra memory during backpropagation. To address this, we suggest treating $||V + \Delta V||_c$ in Eq. (5) as a constant, thereby detaching it from the gradient graph. This means that while $||V + \Delta V||_c$ dynamically reflects the updates of ΔV, it won't receive any gradient during backpropagation. With this modification, the gradient w.r.t m remains unchanged, and $\nabla_{V'}\mathcal{L}$ is redefined as:

$$\nabla_{V'}\mathcal{L} = \frac{m}{C}\nabla_{W'}\mathcal{L} \text{ where } C = ||V'||_c \tag{11}$$

This approach reduces the gradient graph memory consumption drastically without a noticeable difference in accuracy. We conduct an ablation study to evaluate the impact of the proposed modification on fine-tuning LLaMA-7B and VL-BART. The results indicate that the modification leads to a training memory reduction of approximately 24.4% in fine-tuning LLaMA and 12.4% in VL-BART. Furthermore, the accuracy of DoRA with the modification remains unchanged for VL-BART and shows a negligible difference of only 0.2 compared to DoRA without the modification on LLaMA. For a comprehensive comparison of training memory usage and accuracy differences, please see Table 7 in the appendix. Consequently, all subsequent experiments with DoRA incorporate this adjustment.

5. Experiments

We conduct a variety of experiments to showcase the efficacy of DoRA on various tasks including language, image, and video domains. Firstly, we evaluate DoRA against several Parameter-Efficient Fine-Tuning (PEFT) methods by fine-tuning LLaMA-7B/13B, LLaMA2-7B, and LLaMA3-8B on commonsense reasoning tasks. Subsequently, we extend from single modality to multimodality. We compare DoRA with LoRA across multi-task image/video-text understanding tasks using VL-BART and visual instruction tuning with LLaVA-1.5-7B. Following this, we explore the compatibility of DoRA with LoRA and VeRA (Kopiczko et al., 2024) for instruction-tuning on LLaMA-7B and LLaMA2-7B. Furthermore, we perform a series of ablation studies to illustrate that DoRA surpasses LoRA in performance, irrespective of the number of fine-tuning training samples and rank variations. Lastly, We analyze the tuning granularity of DoRA, and show that DoRA can achieve better accuracy than LoRA with fewer trainable parameters by selectively updating only the directional components of certain modules.

5.1. Commonsense Reasoning

We evaluate DoRA against LoRA and several baseline methods which include *Prompt learning (Prefix)* (Li & Liang, 2021), *Series adapter (Series)* (Houlsby et al., 2019), and *Parallel adapter (Parallel)* (He et al., 2021) on LLaMA-7B/13B (Touvron et al., 2023) for commonsense reasoning tasks. We also include ChatGPT's accuracy obtained with gpt-3.5-turbo API using a zero-shot Chain of Thought (OpenAI, 2023; Wei et al., 2022).

The commonsense reasoning tasks comprise 8 sub-tasks, each with a predefined training and testing set. We follow the setting of (Hu et al., 2023) and amalgamate the training datasets from all 8 tasks to create the final training dataset and conduct evaluations on the individual testing dataset for each task. To ensure a fair comparison, we initially fine-tuned models with DoRA following the LoRA configuration, maintaining the same rank while adjusting only the learning rate. The marginal increase of 0.01% in the number of trainable parameters for DoRA over LoRA, as detailed in Table 1, arises from the inclusion of learnable magnitude components (parameter of size $1 \times k$). Then, we further halve the rank used in DoRA compared to LoRA

Table 1. Accuracy comparison of LLaMA 7B/13B, LLaMA2 7B, and LLaMA3 8B with various PEFT methods on eight commonsense reasoning datasets. Results of all the baseline methods on LLaMA 7B/13B are taken from (Hu et al., 2023). Results of LoRA on LLaMA2 7B and LLaMA3 8B are obtained using the hyperparameters described in (Hu et al., 2023). DoRA†: the adjusted version of DoRA with the rank halved.

Model	PEFT Method	# Params (%)	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Avg.
ChatGPT	-	-	73.1	85.4	68.5	78.5	66.1	89.8	79.9	74.8	77.0
LLaMA-7B	Prefix	0.11	64.3	76.8	73.9	42.1	72.1	72.9	54.0	60.6	64.6
	Series	0.99	63.0	79.2	76.3	67.9	75.7	74.5	57.1	72.4	70.8
	Parallel	3.54	67.9	76.4	78.8	69.8	78.9	73.7	57.3	75.2	72.2
	LoRA	0.83	68.9	80.7	77.4	78.1	78.8	77.8	61.3	74.8	74.7
	DoRA† (Ours)	0.43	70.0	82.6	79.7	83.2	80.6	80.6	65.4	77.6	**77.5**
	DoRA (Ours)	0.84	69.7	83.4	78.6	87.2	81.0	81.9	66.2	79.2	**78.4**
LLaMA-13B	Prefix	0.03	65.3	75.4	72.1	55.2	68.6	79.5	62.9	68.0	68.4
	Series	0.80	71.8	83	79.2	88.1	82.4	82.5	67.3	81.8	79.5
	Parallel	2.89	72.5	84.9	79.8	92.1	84.7	84.2	71.2	82.4	81.4
	LoRA	0.67	72.1	83.5	80.5	90.5	83.7	82.8	68.3	82.4	80.5
	DoRA† (Ours)	0.35	72.5	85.3	79.9	90.1	82.9	82.7	69.7	83.6	**80.8**
	DoRA (Ours)	0.68	72.4	84.9	81.5	92.4	84.2	84.2	69.6	82.8	**81.5**
LLaMA2-7B	LoRA	0.83	69.8	79.9	79.5	83.6	82.6	79.8	64.7	81.0	77.6
	DoRA† (Ours)	0.43	72.0	83.1	79.9	89.1	83.0	84.5	71.0	81.2	**80.5**
	DoRA (Ours)	0.84	71.8	83.7	76.0	89.1	82.6	83.7	68.2	82.4	**79.7**
LLaMA3-8B	LoRA	0.70	70.8	85.2	79.9	91.7	84.3	84.2	71.2	79.0	80.8
	DoRA† (Ours)	0.35	74.5	88.8	80.3	95.5	84.7	90.1	79.1	87.2	**85.0**
	DoRA (Ours)	0.71	74.6	89.3	79.9	95.5	85.6	90.5	80.4	85.8	**85.2**

and denote this adjusted configuration as DoRA†. See Table 8 for details on the hyperparameters used.

Table 1 demonstrates that DoRA consistently surpasses all baseline methods across both LLaMA-7B/13B, LLaMA2-7B and LLaMA3-8B. Notably, in the LLaMA-7B model, where LoRA exceeds the performance of other baselines, DoRA further enhances accuracy by 3.7%, outstripping ChatGPT's accuracy levels. Conversely, for LLaMA-13B, where LoRA's effectiveness is inferior to the Parallel adapter, DoRA achieves superior accuracy over LoRA by 1% and comparable accuracy to the Parallel adapter, with only a quarter of the trainable parameters required by the Parallel adapter and without adding any extra inference overhead as the Parallel adapter. Additionally, DoRA consistently surpasses LoRA on both LLaMA2-7B and LLaMA3-8B by 2.1% and 4.4%, respectively. Furthermore, DoRA† exceeds LoRA's performance on LLaMA-7B by 2.8%, on LLaMA-13B by 1%, on LLaMA2-7B by 2.9%, and on LLaMA3-8B by 4.2%, despite having only half as many trainable parameters as LoRA. This outcome suggests that the integration of DoRA enhances the learning capability of LoRA, thereby reducing the need for a higher rank to surpass LoRA in terms of accuracy.

5.2. Image/Video-Text Understanding

Having shown that DoRA can consistently achieve better accuracy on fine-tuning LLM, we would like to see if DoRA can remain competitive on multi-modality fine-tuning tasks. We compare DoRA with LoRA and full fine-tuning on VL-BART which comprises a vision encoder (CLIP-ResNet101 (Radford et al., 2021)) and an encoder-decoder language model ($\text{BART}_{\text{Base}}$ (Lewis et al., 2020)) across four different image-text tasks: VQA^{v2} (Goyal et al., 2017) and GQA (Hudson & Manning, 2019) for visual question answering, NLVR^2 (Suhr et al., 2019) for visual reasoning, and MSCOCO (Chen et al., 2015) for image captioning, and four different video-text tasks from the VALUE (Li et al., 2021) Benchmark: TVQA (Lei et al., 2018) and How2QA (Li et al., 2020) for video question answering, TVC (Lei et al., 2020) and YC2C (Zhou et al., 2018) for video captioning.

Table 2. The multi-task evaluation results on VQA, GQA, NVLR^2 and COCO Caption with the VL-BART backbone.

Method	# Params (%)	VQA^{v2}	GQA	NVLR^2	COCO Cap	Avg.
FT	100	66.9	56.7	73.7	112.0	77.3
LoRA	5.93	65.2	53.6	71.9	115.3	76.5
DoRA (Ours)	5.96	65.8	54.7	73.1	115.9	**77.4**

Table 3. The multi-task evaluation results on TVQA, How2QA, TVC, and YC2C with the VL-BART backbone.

Method	# Params (%)	TVQA	How2QA	TVC	YC2C	Avg.
FT	100	76.3	73.9	45.7	154	87.5
LoRA	5.17	75.5	72.9	44.6	140.9	83.5
DoRA (Ours)	5.19	76.3	74.1	45.8	145.4	**85.4**

We follow the same framework as (Sung et al., 2022) and

fine-tuned VL-BART within a multi-task framework for both image/video-text tasks. We adopt the same setup as that of LoRA outlined in (Sung et al., 2022) when applying DoRA. See Table 9 for the complete hyperparameters. The result of LoRA and FT for both image/video-text tasks is directly quoted from (Sung et al., 2022). We can see that DoRA uniformly surpasses LoRA in accuracy while maintaining a similar count of trainable parameters in both Table 2 and Table 3. In particular, DoRA exceeds LoRA's performance by nearly 1% in image-text understanding tasks, reaching the accuracy level of FT. Moreover, DoRA achieves roughly 2% higher accuracy than LoRA in video-text understanding tasks.

5.3. Visual Instruction Tuning

Table 4. Visual instruction tuning evaluation results for LLaVA-1.5-7B on a wide range of seven vision-language tasks. We directly use checkpoints from (Liu et al., 2023a) to reproduce their results.

Method	# Params(%)	Avg.
FT	100	66.5
LoRA	4.61	66.9
DoRA (Ours)	4.63	**67.6**

We further scale up the model size and compare DoRA to LoRA and FT on the visual instruction tuning tasks with LLaVA-1.5-7B (Liu et al., 2023a) which is composed of a language model, Vicuna-1.5-7B (Peng et al., 2023), and a vision encoder, CLIP ViT-L/336px (Radford et al., 2021). The training datasets contain several datasets from VQA (Goyal et al., 2017; Hudson & Manning, 2019; Marino et al., 2019; Schwenk et al., 2022), OCR (Mishra et al., 2019; Sidorov et al., 2020), region-level VQA (Kazemzadeh et al., 2014; Krishna et al., 2017; Mao et al., 2016), visual conversation (Liu et al., 2023a), and language conversation data. We follow the setting of (Liu et al., 2023a) to filter the training data and construct the tunning prompt format. For a fair comparison, DoRA follows the same configuration as the LoRA configuration provided by (Liu et al., 2023a). The fine-tuned models are then evaluated on seven vision-language benchmarks: VQA^{v2} (Goyal et al., 2017), GQA (Hudson & Manning, 2019), VisWiz (Gurari et al., 2018) SQA (Lu et al., 2022), VQA^{T} (Singh et al., 2019), POPE (Li et al., 2023), and MMBench (Liu et al., 2023c).

From Table 4, we can observe that the average accuracy of LoRA already surpasses FT, which could imply that FT might be experiencing issues with overfitting. Given that DoRA is designed to enhance LoRA's performance to more closely resemble that of FT, in scenarios where FT is inferior to LoRA, DoRA's improvement over LoRA might not be as pronounced as observed in other experiments where FT usually outperforms LoRA. Nonetheless, DoRA still demonstrates superior performance over both LoRA and FT, with an average improvement of 0.7% over LoRA and 1.1% over FT. See Table 10 for the hyperparameters setting and Table 12 for the score of each evaluation benchmark.

5.4. Compatibility of DoRA with other LoRA variants

Table 5. Average scores on MT-Bench assigned by GPT-4 to the answers generated by fine-tuned LLaMA-7B/LLaMA2-7B.

Model	PEFT Method	# Params (%)	Score
LLaMA-7B	LoRA	2.31	5.1
	DoRA (Ours)	2.33	**5.5**
	VeRA	0.02	4.3
	DVoRA (Ours)	0.04	**5.0**
LLaMA2-7B	LoRA	2.31	5.7
	DoRA (Ours)	2.33	**6.0**
	VeRA	0.02	5.5
	DVoRA (Ours)	0.04	**6.0**

Recall from Equation.(1) that ΔW can be adapted by different LoRA variants. With DoRA, the concept of incremental directional update ΔV introduced in Equation.(5) can likewise be replaced with alternative LoRA variants. In this section, we select VeRA (Kopiczko et al., 2024) as a case study to explore DoRA's compatibility with other LoRA variants. VeRA suggests freezing a unique pair of random low-rank matrices to be shared across all layers, employing only minimal layer-specific trainable scaling vectors to capture each layer's incremental updates. This approach allows VeRA to reduce trainable parameters significantly by 10x compared to LoRA, with only a minimal impact on accuracy. We apply VeRA for the directional update in DoRA and name such combination DVoRA. We assess the effectiveness of both DVoRA and DoRA compared to VeRA and LoRA across LLaMA-7B and LLaMA2-7B, focusing on instruction tuning with the 10K subset of cleaned Alpaca dataset (Taori et al., 2023). We utilize the official implementation of VeRA to obtain the results of VeRA and LoRA and fine-tune the model with DVoRA and DoRA using the identical training settings as VeRA and LoRA (see Table 11 in the appendix for more details). The performance of the fine-tuned models is then evaluated on the MT-Bench benchmark (Zheng et al., 2023) by generating model responses to a pre-defined set of 80 multi-turn questions. These responses are then evaluated by GPT-4, which reviews each answer and assigns a numerical score out of 10.

Table 5 presents the average scores for DVoRA, DoRA, VeRA, and LoRA, demonstrating that our proposed method exhibits consistent improvements over VeRA and LoRA for both LLaMA-7B and LLaMA2-7B. This effectively showcases the compatibility of DoRA with VeRA. In particular, DVoRA merges the advantageous qualities of DoRA and

VeRA, attaining scores that are on par with or even surpass those of LoRA, yet with significantly fewer parameters. For example, DVoRA outperforms VeRA by 0.7/0.5 points and achieves the same level of accuracy as LoRA on LLaMA-7B and DoRA on LLaMA2-7B, respectively. Additionally, we present a selection of questions chosen from MT-Bench, accompanied by the responses from LLaMA2-7B fine-tuned using DVoRA and VeRA in the appendix (Table 13 and 14) where we can observe that the answers given by DVoRA tend to be more precise and structural.

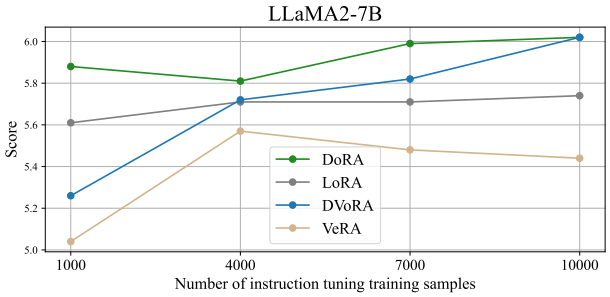

Figure 3. Performance of fine-tuned LLaMA2-7B on MT-Bench using different numbers of Alpaca training samples.

Next, to further assess DoRA's ability to remain competitive under varying amounts of training data, considering that in practical situations, access to extensive fine-tuning datasets is frequently limited. We compare DoRA to LoRA and DVoRA to VeRA for fine-tuning LLaMA2-7B/LLaMA-7B with a range of instruction-tuning sample sizes, specifically 1000, 4000, 7000, 10000, with 10000 being the setting of (Kopiczko et al., 2024). We visualize the average performance of each method on LLaMA2-7B in Figure 3, and on LLaMA-7B in Figure 7 in the appendix. The result shows that DoRA and DVoRA consistently outperform LoRA and VeRA across all training sample sizes. For instance, with 7000 training samples, DoRA and DVoRA surpass LoRA and VeRA by margins of 0.3 and 0.33, respectively. Even when the sample size is reduced to 1000, DoRA and DVoRA maintain their lead with advantages of 0.29 and 0.22 over LoRA and VeRA, respectively. This demonstrates that our methods persistently enhance performance over LoRA and VeRA, regardless of the training sample volume.

5.5. Robustness of DoRA towards different rank settings

This section explores the impact of various rank configurations on DoRA and LoRA by adjusting r within the set $\{4, 8, 16, 32, 64\}$ and assessing the performance of the fine-tuned LLaMA-7B on commonsense reasoning tasks as outlined in Sec 5.1. The average accuracies of LoRA and DoRA across different ranks are depicted in Figure 4, with detailed numbers presented in Table 15. From Figure 4, we can observe

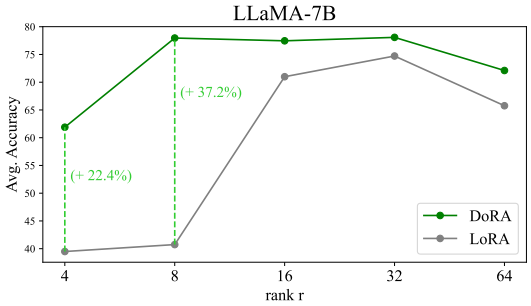

Figure 4. Average accuracy of LoRA and DoRA for varying ranks for LLaMA-7B on the commonsense reasoning tasks.

Table 6. Accuracy comparison of LLaMA 7B/13B with two different tuning granularity of DoRA. Columns **m** and **V** designate the modules with tunable magnitude and directional components, respectively. Each module is represented by its first letter as follows: (Q)uery, (K)ey, (V)alue, (O)utput, (G)ate, (U)p, (D)own.

Model	PEFT Method	# Params (%)	m	V	Avg.
LLaMA-7B	LoRA	0.83	-	-	74.7
	DoRA (Ours)	0.84	QKVUD	QKVUD	78.1
	DoRA (Ours)	0.39	QKVOGUD	QKV	77.5
LLaMA-13B	LoRA	0.67	-	-	80.5
	DoRA (Ours)	0.68	QKVUD	QKVUD	81.5
	DoRA (Ours)	0.31	QKVOGUD	QKV	81.3

that DoRA consistently surpasses LoRA across all rank configurations. Notably, the performance gap widens for ranks below 8, where LoRA's average accuracies drop to 40.74% for $r = 8$ and 39.49% for $r = 4$. In contrast, DoRA retains a notable accuracy of 77.96% for $r = 8$ and 61.89% for $r = 4$, demonstrating its resilience and consistently superior performance over LoRA regardless of the rank setting.

5.6. Tuning Granularity Analysis

The visualization in Figure 2 indicates that significant changes in magnitude often result in relatively smaller directional changes. Given this observation and the fact that directional updates account for most of the trainable parameters, it prompts an investigation into whether it is possible to decrease the number of trainable parameters by updating only the magnitude components of specific modules while continuing to update both the magnitude and directional components for the remaining linear modules.

Our findings indicate that, in contrast to the original configuration suggested for LoRA in (Hu et al., 2023), which requires updates to both the Multi-head Attention and MLP layers for optimal performance, DoRA can already achieve superior accuracy by updating only the directional and magnitude components of the multi-head layers and the magni-

tude of the MLP layers. Specifically, as shown in Table 6, by updating the directional and magnitude components of the QKV modules and only the magnitude of the rest of the layers, DoRA surpasses LoRA by 2.8% on LLaMA-7B and 0.8% on LLaMA-13B, while utilizing only less than half of the trainable parameters compared to LoRA.

6. Broader Impacts

6.1. QDoRA: Enhancements to QLoRA

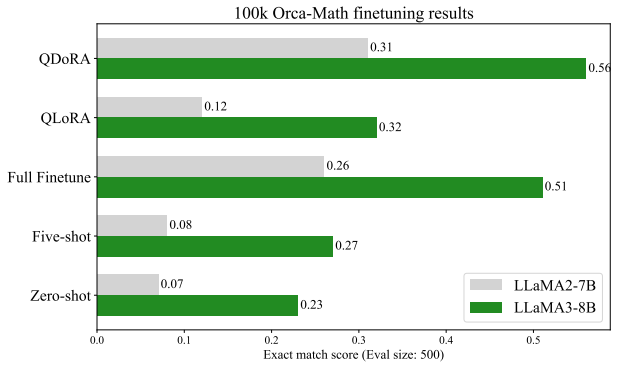

Figure 5. Accuracy comparison of LLaMA2-7B/LLaMA3-8B with QDoRA, QLoRA and FT on Orca-Math (Mitra et al., 2024).

While finetuning LLMs with PEFT significantly reduces training memory overhead, a considerable amount of GPU memory is still required to initially load the model weights onto the GPUs. To further decrease the memory demands of finetuning, QLoRA (Dettmers et al., 2023) suggests quantizing the pretrained model to 4-bit and finetuning LoRA on top of the frozen low-bit backbone. With our porposed DoRA, which narrows the gap between LoRA and FT, it is natural to also explore whether DoRA can enhance the accuracy of LoRA within the QLoRA framework. Recently, (Kerem Turgutlu, 2024) launch a project that substitutes the LoRA component in QLoRA with DoRA, dubbing it QDoRA, and incorporate the training pipeline with Fully Sharded Data Parallel (FSDP) (Zhao et al., 2023) to enable model splitting and parallel training across multiple GPUs.

They conducted experiments on fine-tuning LLaMA2-7B/LLaMA3-8B using the Orca-Math(Mitra et al., 2024) dataset with QDoRA, QLoRA, and FT. The training set included 100k samples, with 500 reserved for evaluation using the exact match score as the metric. In addition to the fine-tuned models, they also reported results from zero-shot, few-shot, and FT with post-training quantization (PTQ), where the FT model is quantized to the BnB NF4 format after training. According to Figure 5, QDoRA not only significantly surpasses QLoRA by 0.19/0.23 on LLaMA2-7B and LLaMA3-8B, but it also slightly outperforms FT on both models, while using considerably less memory. This indicates that QDoRA can effectively combines the parameter efficiency of QLoRA with the more granular optimization of full finetuning. These initial findings suggest that QDoRA holds considerable promise and could hugely benefit the opensoure community by substantially lowering the GPU memory requirements for fine-tuning large language models.

6.2. Text-to-Image Generation

Recently, as diffusion models have expanded in size, LoRA has become a popular method for efficiently fine-tuning large stable diffusion models. In this section, we aim to explore whether DoRA's advantages over LoRA extend to the task of text-to-image generation. We follow the training pipeline of DreamBooth (Ruiz et al., 2023) for fine-tuning SDXL (Podell et al., 2023), utilizing the advanced training scripts developed by HuggingFace. The hyperparameter settings for LoRA and DoRA are kept the same, and we fine-tune the model using two challenging datasets: 3D icons and Lego sets. The sample seeds for generating the images are kept the same for LoRA and DoRA for fair comparison. The generated images are shown in Figure 8 and 9 in the appendix. The results indicate that DoRA achieves significantly better personalization than LoRA when using the same training settings, and more accurately reflects the training targets. For example, in Figure 8, the first sub-figure of DoRA's output features a unique round square around the image, which is a feature common to all the training targets. In contrast, this feature is absent in all the LoRA outputs. A similar observation could be found with the Lego training targets, where only the DoRA outputs consistently incorporate the Lego logo in the generated images.

7. Conclusion

In this work, we first conduct a novel weight decomposition analysis to reveal the distinct learning patterns between LoRA and FT. Building on these insights, we introduce DoRA, a fine-tuning method that is compatible with LoRA and its variants and exhibits a closer resemblance to FT's learning behavior. DoRA consistently outperforms LoRA across various fine-tuning tasks and model architectures. Specifically, DoRA improves upon LoRA in commonsense reasoning and visual instruction tuning tasks. Furthermore, DoRA also shows compatibility with VeRA on the Alpaca instruction tuning task. Moreover, DoRA can be considered as a costless alternative to LoRA, as its decomposed magnitude and direction components can be merged back into the pre-trained weight after the training, ensuring that there is no extra inference overhead. For future work, we wish to explore the generalizability of DoRA in domains beyond language and vision, particularly in the field of audio.

Acknowledgements

We extend our gratitude to Benjamin Bossan, Younes Belkada, and Sourab Mangrulkar from Hugging Face for their assistance in integrating DoRA into the PEFT package, thus making our work more accessible to the broader public. We thank Kerem Turgutlu, Jonathan Whitaker, and Jeremy Howard from Answer.AI for their work on the implementation and experiments of QDoRA/FSDP, which makes fine-tuning of large language models with DoRA on consumer GPUs a lot more feasible. We also thank Sebastian Raschka for his well-written tutorial on DoRA which offers a thorough overview of the background knowledge necessary to comprehend DoRA.

Impact Statement

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

References

Chen, X., Fang, H., Lin, T.-Y., Vedantam, R., Gupta, S., Dollár, P., and Zitnick, C. L. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. In Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 10088–10115. Curran Associates, Inc., 2023.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

Gurari, D., Li, Q., Stangl, A. J., Guo, A., Lin, C., Grauman, K., Luo, J., and Bigham, J. P. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3608–3617, 2018.

He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., and Neubig, G. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015.

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

Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Hu, Z., Wang, L., Lan, Y., Xu, W., Lim, E.-P., Bing, L., Xu, X., Poria, S., and Lee, R. LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Hudson, D. A. and Manning, C. D. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

Hyeon-Woo, N., Ye-Bin, M., and Oh, T.-H. Fedpara: Low-rank hadamard product for communication-efficient federated learning. In *International Conference on Learning Representations*, 2022.

Karimi Mahabadi, R., Ruder, S., Dehghani, M., and Henderson, J. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 565–576, 2021.

Kazemzadeh, S., Ordonez, V., Matten, M., and Berg, T. ReferItGame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 787–798, 2014.

Kerem Turgutlu, Jonathan Whitaker, J. H. Efficient finetuning of llama 3 with fsdp qdora. https://www.answer.ai/posts/2024-04-26-fsdp-qdora-llama3.html, 2024.

Kopiczko, D. J., Blankevoort, T., and Asano, Y. M. Vera: Vector-based random matrix adaptation. In *International Conference on Learning Representations*, 2024.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, pp. 32–73, 2017.

Lei, J., Yu, L., Bansal, M., and Berg, T. TVQA: Localized, compositional video question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1369–1379, 2018.

Lei, J., Yu, L., Berg, T. L., and Bansal, M. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *European Conference on Computer Vision*, pp. 447–463, 2020.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, 2020.

Li, J., Li, D., Xiong, C., and Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pp. 12888–12900, 2022.

Li, L., Chen, Y.-C., Cheng, Y., Gan, Z., Yu, L., and Liu, J. HERO: Hierarchical encoder for Video+Language omni-representation pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2046–2065, 2020.

Li, L., Lei, J., Gan, Z., Yu, L., Chen, Y.-C., Pillai, R., Cheng, Y., Zhou, L., Wang, X. E., Wang, W. Y., et al. Value: A multi-task benchmark for video-and-language understanding evaluation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021.

Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, X., and Wen, J.-R. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 292–305, 2023.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a.

Liu, W., Qiu, Z., Feng, Y., Xiu, Y., Xue, Y., Yu, L., Feng, H., Liu, Z., Heo, J., Peng, S., et al. Parameter-efficient orthogonal finetuning via butterfly factorization. *arXiv preprint arXiv:2311.06243*, 2023b.

Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023c.

Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.-W., Zhu, S.-C., Tafjord, O., Clark, P., and Kalyan, A. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, pp. 2507–2521, 2022.

mahabadi, R. K., Henderson, J., and Ruder, S. Compacter: Efficient low-rank hypercomplex adapter layers. In *Advances in Neural Information Processing Systems*, 2021.

Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A. L., and Murphy, K. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 11–20, 2016.

Marino, K., Rastegari, M., Farhadi, A., and Mottaghi, R. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pp. 3195–3204, 2019.

Mishra, A., Shekhar, S., Singh, A. K., and Chakraborty, A. Ocr-vqa: Visual question answering by reading text in images. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 947–952, 2019.

Mitra, A., Khanpour, H., Rosset, C., and Awadallah, A. Orca-math: Unlocking the potential of slms in grade school math, 2024.

OpenAI. Welcome to the openai platform, 2023. URL https://platform.openai.com/docs/introduction.

Peng, B., Li, C., He, P., Galley, M., and Gao, J. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.

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

Ponti, E. M., Sordoni, A., Bengio, Y., and Reddy, S. Combining modular skills in multitask learning. *arXiv preprint arXiv:2202.13914*, 2022.

Qin, C., Zhang, A., Zhang, Z., Chen, J., Yasunaga, M., and Yang, D. Is chatgpt a general-purpose natural language processing task solver? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1339–1384, 2023.

Qiu, Z., Liu, W., Feng, H., Xue, Y., Feng, Y., Liu, Z., Zhang, D., Weller, A., and Schölkopf, B. Controlling text-to-image diffusion by orthogonal finetuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

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

Razdaibiedina, A., Mao, Y., Khabsa, M., Lewis, M., Hou, R., Ba, J., and Almahairi, A. Residual prompt tuning: improving prompt tuning with residual reparameterization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 6740–6757, 2023.

Renduchintala, A., Konuk, T., and Kuchaiev, O. Tied-lora: Enhacing parameter efficiency of lora with weight tying. *arXiv preprint arXiv:2311.09578*, 2023.

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., and Aberman, K. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22500–22510, 2023.

Salimans, T. and Kingma, D. P. Weight normalization: a simple reparameterization to accelerate training of deep neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 901–909, 2016.

Schwenk, D., Khandelwal, A., Clark, C., Marino, K., and Mottaghi, R. A-okvqa: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision*, pp. 146–162, 2022.

Sidorov, O., Hu, R., Rohrbach, M., and Singh, A. Textcaps: a dataset for image captioning with reading comprehension. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pp. 742–758, 2020.

Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., and Rohrbach, M. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8317–8326, 2019.

Suhr, A., Zhou, S., Zhang, A., Zhang, I., Bai, H., and Artzi, Y. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6418–6428, 2019.

Sung, Y.-L., Cho, J., and Bansal, M. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5227–5237, 2022.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model, 2023. URL https://github.com/tatsu-lab/stanford_alpaca.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Wang, Y., Wu, J., Dabral, T., Zhang, J., Brown, G., Lu, C.-T., Liu, F., Liang, Y., Pang, B., Bendersky, M., et al. Non-intrusive adaptation: Input-centric parameter-efficient fine-tuning for versatile multimodal modeling. *arXiv preprint arXiv:2310.12100*, 2023.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., brian ichter, Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

Yeh, S.-Y., Hsieh, Y.-G., Gao, Z., Yang, B. B., Oh, G., and Gong, Y. Navigating text-to-image customization: From lycoris fine-tuning to model evaluation. *arXiv preprint arXiv:2309.14859*, 2023.

Zhang, Q., Chen, M., Bukharin, A., He, P., Cheng, Y., Chen, W., and Zhao, T. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2023.

Zhao, Y., Gu, A., Varma, R., Luo, L., Huang, C.-C., Xu, M., Wright, L., Shojanazeri, H., Ott, M., Shleifer, S., et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

Zhou, L., Xu, C., and Corso, J. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

A. Appendix

A.1. Weight decomposition analysis on the value weight matrix

In this section, we illustrate the changes in magnitude and direction within the value weight matrix for FT, LoRA, and DoRA across different training steps and layers, as shown in Figure 6. This reveals patterns similar to those seen in the query weight matrix depicted in Figure 2, indicating that DoRA is capable of displaying learning behaviors that closely mirror those of FT across various modules.

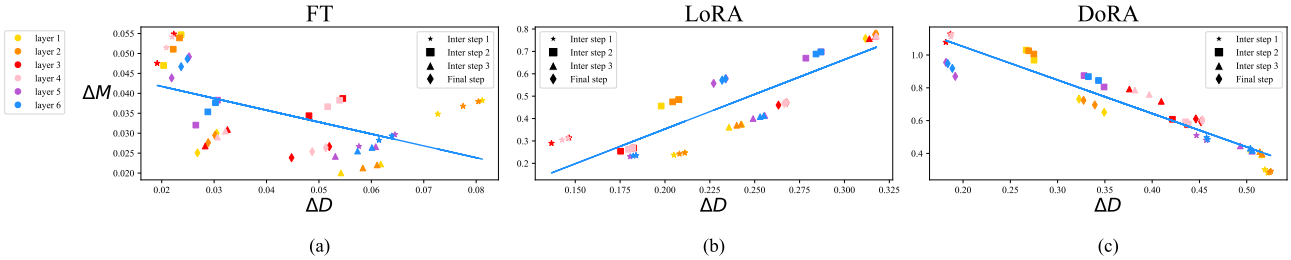

Figure 6. Magnitude and Directional changes of FT (a), LoRA (b), and DoRA (c) of the V weight matrices across different layers and intermediate steps.

A.2. Ablation study for the modification to reduce DoRA training cost

Table 7 presents the GPU cost and the average accuracy of DoRA with and without the proposed modification for commonsense reasoning tasks and image-text understanding tasks. The results indicate that the modification leads to a training memory reduction of approximately 24.4% in fine-tuning LLaMA and 12.4% in VL-BART. Furthermore, the accuracy of DoRA with the modification remains unchanged for VL-BART and shows a negligible difference of only 0.2 compared to DoRA without the modification on LLaMA.

Table 7. GPU cost and accuracy of DoRA with or without the modification on the commonsense reasoning tasks and image-text understanding tasks.

Model	PEFT Method	Accumulation steps	Batch Size	GPU Memory Cost (GB)	# Params (%)	Avg.
LLaMA-7B	DoRA w/o modification	4	16	37.3	0.84	78.3
	DoRA			28.2 (-24.4%)	0.84	78.1
VL-BART	DoRA w/o modification	-	300	23.4	5.96	77.3
	DoRA			20.5 (-12.4%)	5.96	77.4

A.3. Hyperparameters

Table 8. Hyperparameter configurations of DoRA for LLaMA-7B/13B, LLaMA2-7B, and LLaMA3-8B on the commonsense reasoning tasks.

Hyperparameters (DoRA)	LLaMA-7B		LLaMA-13B		LLaMA2-7B		LLaMA3-8B	
Rank r	16	32	16	32	16	32	16	32
α	32	64	32	64	32	64	32	64
Dropout	0.05							
Optimizer	AdamW							
LR	2e-4	1e-4	3e-4	2e-4	2e-4	2e-4	1e-4	1e-4
LR Scheduler	Linear							
Batch size	16							
Warmup Steps	100							
Epochs	3							
Where	Q,K,V,Up,Down							

Table 9. Hyperparameter configurations of DoRA for fine-tuning VL-Bart on image/video-text tasks.

Hyperparameters (DoRA)	image-text	video-text
Rank r	128	
α	128	
Dropout	0.0	
Optimizer	AdamW	
LR	1e-3	3e-4
LR Scheduler	Linear	
Batch size	300	40
Warmup ratio	0.1	
Epochs	20	7
Where	Q,K	

Table 10. Hyperparameter configurations of DoRA and LoRA for fine-tuning LLaVA-1.5-7B with visual instruction tuning datasets.

Hyperparameters	DoRA	LoRA
Rank r	128	
α	256	
Dropout	0.05	
Optimizer	AdamW	
LR	2e-4	
LR Scheduler	Cosine decay	
Batch size	16	
Warmup ratio	0.03	
Epochs	1	
Where	Q,K,V,O,Up,Down,Gate	

Table 11. Hyperparameter configurations of DoRA and DVoRA for fine-tuning LLaMA-7B and LLaMA2-7B with cleaned Alpaca dataset.

Hyperparameters (DoRA)	LLaMA-7B	LLaMA2-7B	**Hyperparameters (DVoRA)**	LLaMA-7B	LLaMA2-7B
Rank r	64		Rank r	1024	
Dropout	0.0		Dropout	0.0	
Optimizer	AdamW		Optimizer	AdamW	
LR	4e-4		LR	4e-3	
LR Scheduler	Cosine		LR Scheduler	Cosine	
Batch size	4		Batch size	4	
Accumulation Steps	4		Accumulation Steps	4	
Warmup ratio	0.1		Warmup ratio	0.1	
Epochs	1		Epochs	1	
Where	Q,K,V,O,Up,Down,Gate		Where	Q,K,V,O,Up,Down,Gate	

A.4. Visual instruction tuning evaluation result

Table 12 presents the evaluation result of LLaVA-1.5-7B fine-tuned using DoRA, LoRA, and FT with visual instruction tuning data. The evaluation is conducted on an array of seven different vision-language benchmarks. From the table, we can see that DoRA consistently outmatch FT, and can achieve an average accuracy better than LoRA.

Table 12. Visual instruction tuning evaluation result of DoRA, LoRA, and FT for LLaVA-1.5-7B on a wide range of 7 vision-language tasks.

Method	**# Params** (%)	**VQA**^{v2}	**GQA**	**VisWiz**	**SQA**	**VQA**^{T}	**POPE**	**MMBench**	**Avg.**
FT	100	78.5	61.9	50.0	66.8	58.2	85.9	64.3	66.5
LoRA	4.61	79.1	62.9	47.8	68.4	58.2	86.4	66.1	66.9
DoRA (Ours)	4.63	78.6	62.9	52.2	69.9	57	87.2	66.1	**67.6**

A.5. Instruction-tuning evaluation examples

We sample two questions from the MT-Bench question set and present the respective responses from DVoRA and VeRA, including the scores and comments provided by GPT-4. From these examples, we can observe that responses from DVoRA are generally more accurate and well-organized compared to those from VeRA. For instance, in response to Question 24, DVoRA leverages a well-known superhero story that is familiar to children, whereas VeRA's response begins with an uncertain tone, potentially making it less engaging than DVoRA's reply.

Table 13. Question 24 from MT-Bench alongside with response from LLaMA2-7B fine-tuned with DVoRA and VeRA, and the evaluation provided by GPT-4.

Question 24	As a superhero, how would you explain your origin story to a curious child?
VeRA	Well, I'm not really sure where I come from. All I know is that one day, I woke up with special powers and abilities that allowed me to help people. I decided to use my powers to fight crime and protect the innocent, and that's how I became a superhero
DVoRA	As a superhero, I have the ability to use my super powers to protect people and fight against evil. My origin story begins when I was just a regular kid like you, but one day, I was bitten by a radioactive spider, and it gave me incredible powers. I could climb walls, shoot webs, and see in the dark. I decided to use my powers to help people, and that's how I became your friendly neighborhood Spider-Man.
GPT-4 Evaluation	Score: 5 7. Assistant 1's response starts with an uncertain tone, which might not be very engaging for a child. The explanation is quite vague and doesn't offer a clear or imaginative origin story that could captivate a child's interest. It does, however, maintain the theme of using powers for good, which is a positive message. Assistant 2's response is more detailed and uses a familiar superhero origin story (Spider-Man's) that many children can relate to and understand. It's engaging and provides a clear narrative of how an ordinary person can become a superhero. However, it directly references Spider-Man, which is not an original origin story and could be misleading if the assistant is supposed to be a unique superhero. Despite this, the response is more likely to resonate with a child due to its clarity and reliability. Both responses could be improved by providing an original and imaginative origin story that doesn't rely on existing superhero narratives. They should also ensure that the story is simple enough for a child to understand while being engaging and inspiring.

Table 14. Question 33 from MT-Bench alongside with response from LLaMA2-7B fine-tuned with DVoRA and VeRA, and the evaluation provided by GPT-4.

Question 33	Why might someone choose to use a paper map or ask for directions instead of relying on a GPS device or smartphone app?
VeRA	There are several reasons why someone might choose to use a paper map or ask for directions instead of relying on a GPS device or smartphone app. Some people may prefer the physical act of navigating with a paper map, as it can be more tactile and interactive. Others may prefer to have a hard copy of the map in case the GPS or app fails or the battery dies. Additionally, some people may not have access to a GPS device or smartphone, or may not want to rely on technology for navigation. Finally, some people may simply prefer the security and peace of mind that comes with knowing exactly where they are and how to get to their destination
DVoRA	There are several reasons why someone might choose to use a paper map or ask for directions instead of relying on a GPS device or smartphone app. Here are some of them: 1. Navigation accuracy: GPS devices and smartphone apps rely on satellite signals to pinpoint your location, but this technology is not foolproof. GPS signals can be blocked by tall buildings, tunnels, or other obstacles, resulting in inaccurate or missing information. Paper maps, on the other hand, are based on accurate, up-to-date information that is not subject to these limitations. 2. Cost: GPS devices and smartphone apps can be expensive to purchase and maintain. Paper maps are free to use and do not require an ongoing subscription or data plan. 3. Personal preference: Some people prefer the tactile experience of using a paper map, which can be easier to read and navigate than a screen. Others enjoy the challenge of navigating without technology, which can help them learn more about the area they are visiting. 4. Battery life: GPS devices and smartphone apps can drain batteries quickly, especially when used for navigation. Paper maps do not require power, so they can be used for extended periods without recharging. 5. Offline use: GPS devices and smartphone apps rely on an internet connection to function properly. If there is no signal, the device will not be able to provide accurate information. Paper maps, on the other hand, can be used offline, making them ideal for use in remote or unpopulated areas. In summary, there are several reasons why someone might choose to use a paper map or ask for directions instead of relying on a GPS device or smartphone app. These include navigation accuracy, cost, personal preference, battery life, and offline use.
GPT-4 Evaluation	Score: 6 8. Assistant 1's response is somewhat brief and lacks specific details that could be helpful to the user. It mentions the tactile nature of paper maps, the possibility of technology failure, lack of access to technology, and a general preference for security and peace of mind. However, it does not delve into the practical considerations such as GPS signal limitations, cost, battery life, or offline use, which are important factors in deciding between paper maps and digital navigation tools. Assistant 2's response is more comprehensive and provides a structured list of reasons, including navigation accuracy, cost, personal preference, battery life, and offline use. Each point is explained clearly, giving the user a better understanding of the potential advantages of paper maps over GPS devices or smartphone apps. The response is relevant, accurate, and detailed, making it more helpful to the user. Both assistants provided relevant and accurate information, but Assistant 2's answer is more detailed and structured, which is why it receives a higher score. The scores reflect the level of detail and the usefulness of the information provided to the user, with Assistant 2 offering a more complete answer to the question

A.6. Robustness towards different numbers of instruction tunning examples

Figure 7 depicts the score of DoRA, LoRA, DVoRA, and VeRA for fine-tuning LLaMA-7B with varying instruction-tuning sample sizes, including 1000, 4000, 7000, and 10000. From the figure, we can observe that similar to that of the result on LLaMA2-7B, both DoRA and DVoRA consistently outperform LoRA and VeRA across the entire range of training sample sizes.

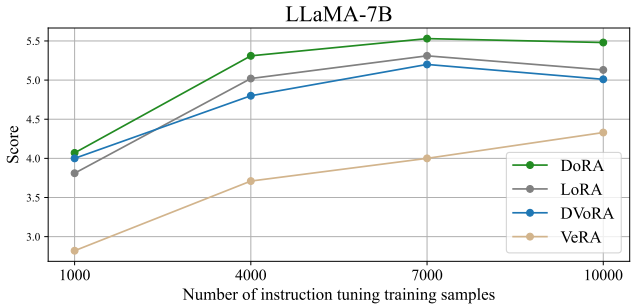

Figure 7. Performance of LLaMA-7B fine-tuned with LoRA, DoRA, VeRA, and DVoRA on MT-Bench using different numbers of training samples from Alpaca dataset (Taori et al., 2023).

A.7. Robustness of DoRA towards different rank

Table 15 shows a comparison of the average accuracy between LoRA and DoRA method across various rank settings for commonsense reasoning tasks. DoRA consistently outperforms LoRA at all rank settings, with the performance gap widening as the rank decreases. This suggests that our method effectively enhances the learning capacity of LoRA, enabling it to achieve better accuracy with fewer trainable parameters.

Table 15. Accuracy comparison of LoRA and DoRA with varying ranks for LLaMA-7B on the commonsense reasoning tasks.

PEFT Method	rank r	# Params (%)	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Avg.
LoRA	4	0.10	2.3	46.1	18.3	19.7	55.2	65.4	51.9	57	39.5
	8	0.21	31.3	57.0	44.0	11.8	43.3	45.7	39.2	53.8	40.7
	16	0.42	69.9	77.8	75.1	72.1	55.8	77.1	62.2	78.0	70.9
	32	0.83	68.9	80.7	77.4	78.1	78.8	77.8	61.3	74.8	74.7
	64	1.64	66.7	79.1	75.7	17.6	78.8	73.3	59.6	75.2	65.8
DoRA (Ours)	4	0.11	51.3	42.2	77.8	25.4	78.8	78.7	62.5	78.6	61.9
	8	0.22	69.9	81.8	79.7	85.2	80.1	81.5	65.7	79.8	77.9
	16	0.43	70.0	82.6	79.7	83.2	80.6	80.6	65.4	77.6	77.5
	32	0.84	69.7	83.4	78.6	87.2	81.0	81.9	66.2	79.2	78.4
	64	1.65	69.9	81.4	79.1	40.7	80.0	80.9	65.5	79.4	72.1

A.8. Text-to-Image Generation

Figures 8 and 9 show the images produced by SDXL fine-tuned with DoRAand LoRA via DreamBooth (Ruiz et al., 2023) personalization techniques on two distinct training sets: 3D Icon[1] and Lego[2]. The results reveal that DoRA can achieve considerably better personalization than LoRA with identical training configurations, more closely matching the training target.

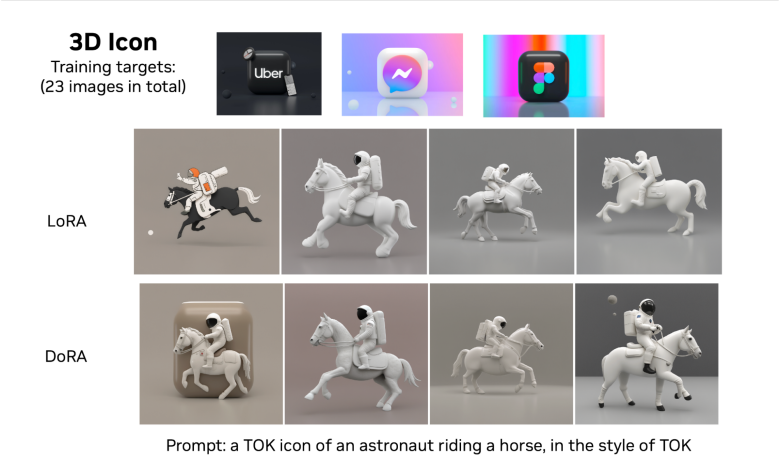

Figure 8. Images generated with SDXL finetuned with LoRA and DoRA on the 3D Icon training sets.

Figure 9. Images generated with SDXL finetuned with LoRA and DoRA on the Lego training sets.

[1] https://huggingface.co/datasets/linoyts/3d_icon
[2] https://huggingface.co/datasets/merve/lego_sets_latest