

---

# 3-in-1: 2D Rotary Adaptation for Efficient Finetuning, Efficient Batching and Composability

---

Baohao Liao<sup>1,2\*</sup>

Christof Monz<sup>1</sup>

<sup>1</sup>Language Technology Lab, University of Amsterdam

<sup>2</sup>eBay Inc., Aachen, Germany

Code: <https://github.com/BaohaoLiao/road>

## Abstract

Parameter-efficient finetuning (PEFT) methods effectively adapt large language models (LLMs) to diverse downstream tasks, reducing storage and GPU memory demands. Despite these advantages, several applications pose new challenges to PEFT beyond mere parameter efficiency. One notable challenge involves the efficient deployment of LLMs equipped with multiple task- or user-specific adapters, particularly when different adapters are needed for distinct requests within the same batch. Another challenge is the interpretability of LLMs, which is crucial for understanding how LLMs function. Previous studies introduced various approaches to address different challenges. In this paper, we introduce a novel method, RoAd, which employs a straightforward 2D rotation to adapt LLMs and addresses all the above challenges: (1) RoAd is remarkably parameter-efficient, delivering optimal performance on GLUE, eight commonsense reasoning tasks and four arithmetic reasoning tasks with  $< 0.1\%$  trainable parameters; (2) RoAd facilitates the efficient serving of requests requiring different adapters within a batch, with an overhead comparable to element-wise multiplication instead of batch matrix multiplication; (3) RoAd enhances LLM’s interpretability through integration within a framework of distributed interchange intervention, demonstrated via composition experiments.

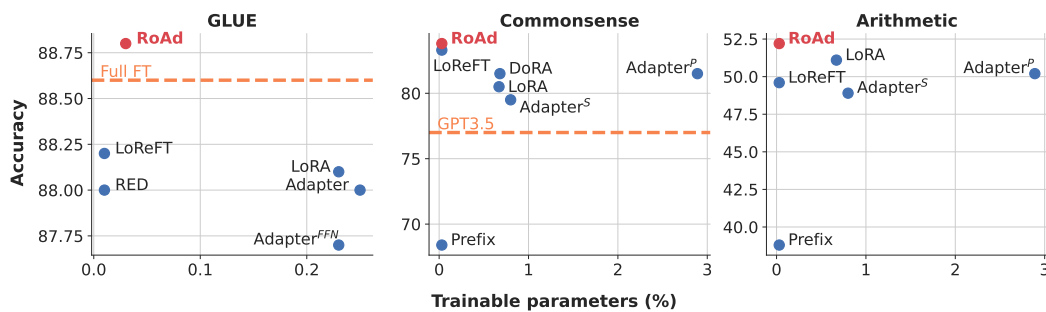


Figure 1: Performance of various PEFT methods on the GLUE benchmark, eight commonsense reasoning tasks and four arithmetic reasoning tasks with RoBERTa-large or LLaMA-13B.

## 1 Introduction

Large language models (LLMs), trained on extensive web-scale datasets to perform tasks such as predicting masked words [8, 31, 45] or anticipating the next word in a sentence [17, 52, 53],

\*Correspondence to b.liao@uva.nl. Please go to <https://arxiv.org/abs/2409.00119> for the newest version.

demonstrate remarkable effectiveness across a range of NLP applications. For tasks where the data distribution diverges from that of the pretraining corpus, finetuning emerges as an effective way to tailor an LLM to specific requirements. Leveraging the capabilities of LLMs, recent studies [13, 14, 22, 23, 25, 27, 42, 60, 62, 65] demonstrate that training only a subset of an LLM’s parameters can yield performance on par with full finetuning. This approach, termed parameter-efficient finetuning (PEFT), provides two primary advantages: (1) It reduces the storage requirements for trained parameters, as it necessitates preserving only a universal LLM alongside a minimal set of task-specific parameters; (2) It decreases GPU memory consumption during finetuning, owing to the reduction in optimizer state sizes which correlate directly with the number of trainable parameters.

With the evolution of PEFT, concerns extend beyond mere parameter efficiency. PEFT encounters a variety of challenges brought forth by diverse applications. A significant challenge is the efficient deployment of personalized or task-specific LLMs [25, 57]. These applications frequently require distinct sets of trained parameters for different tasks or users. When multiple users submit requests simultaneously, it becomes crucial to process these requests collectively in a single batch. Given that each request may require a unique set of parameters, using batch matrix multiplication can efficiently handle these requests by leveraging GPU parallelism. However, the batch matrix multiplication still incurs considerable overhead [1, 57], necessitating the exploration of more efficient methods.

Another challenge is the interpretability of LLMs that contain a billion-scale of parameters, making it difficult to explore their mechanism. PEFT provides an alternative approach by constraining the number of trainable parameters, thereby aiding in interpretability. Recent advancements in PEFT methods, particularly those focusing on representation editing [54, 60, 67], can be incorporated within an intervention framework [11]. This integration enhances their capability for interpretability, offering a more manageable means of dissecting the operational intricacies of LLMs.

In this paper, we introduce a novel technique termed 2D rotary adaptation (RoAd) which efficiently adapts LLMs using a minimal number of trainable parameters. Furthermore, RoAd enhances both batching efficiency and composability. Our initial investigation reveals that finetuning primarily alters the angular components of the representations in pretrained LLMs, rather than their magnitudes (Section §3.1). Based on this observation, we employ a strategy of rotating certain subspaces within the representations to emulate finetuning effects. Specifically, we implement a 2D rotational approach on the representations and develop three distinct variants of RoAd (Section §3.2).

To assess the efficacy of RoAd, we perform comprehensive evaluations on the GLUE benchmark [56], eight commonsense reasoning tasks and four arithmetic reasoning tasks, utilizing RoBERTa [31] and LLaMA [52, 53] (Section §4.1). The results consistently show that RoAd surpasses other PEFT methods while maintaining a significantly reduced scale of trainable parameters ( $< 0.1\%$ ), as depicted in Figure 1. Additionally, RoAd employs element-wise rather than matrix multiplication, which notably improves throughput when serving heterogeneous requests within the same batch, achieving twice the throughput of LoRA [14] (Section §4.2). Furthermore, RoAd can be seamlessly integrated within an intervention framework [11], thereby enhancing model interpretability. We illustrate this through a composition experiment, demonstrating RoAd’s capacity to merge weights trained for different tasks and display a new capability (Section §4.3).

## 2 Background

In this section, we outline the challenges tackled in this work, illustrating the constraints of existing methods and objectives that drive the development of the proposed method, RoAd.

### 2.1 Parameter-efficient finetuning (PEFT)

Existing PEFT techniques can be categorized into three groups: adapter-based, prompt-based, and latency-less methods. Adapter-based methods [12, 13, 42] incorporate adapters either in parallel with or sequentially to the existing Transformer [55] modules. This incorporation necessitates modifications to the LLM architecture, consequently adding extra latency during inference. Prompt-based methods [19, 21, 43] enhance the input by appending new trainable tokens, which lengthens the sequence and thereby increases the computational overhead during inference. Latency-less methods, such as LoRA [14] and its variants [22, 27, 65], apply low-rank matrices to adapt the pretrained weights. These matrices can be seamlessly integrated into the existing weight matrices following

finetuning, thus preserving the original LLM architecture. Specifically, LoRA adapts an LLM as  $\mathbf{W} = \mathbf{W}^0 + \Delta\mathbf{W}$ , where  $\mathbf{W}^0 \in \mathbb{R}^{d_1 \times d_2}$  is the pretrained weight and  $\Delta\mathbf{W} = \mathbf{B}\mathbf{A}$  with  $\mathbf{B} \in \mathbb{R}^{d_1 \times r}$ ,  $\mathbf{A} \in \mathbb{R}^{r \times d_2}$ ,  $r \ll d_1$  and  $r \ll d_2$ . Our proposed method, RoAd, aligns with the latency-less category and integrates effortlessly into the existing linear layer without imposing additional overhead during inference. Moreover, RoAd demonstrates exceptional parameter efficiency. The quantity of its trainable parameters is equivalent to that of a LoRA module with a rank  $r = 0.5$ .

**Orthogonal finetuning.** Drawing on the concept of hyperspherical energy and its role in characterizing generalization [28, 29], OFT [44] introduces orthogonal finetuning, an effective PEFT method for finetuning text-to-image diffusion models. Specifically, OFT implements an orthogonal matrix  $\mathbf{R} \in \mathbb{R}^{d_1 \times d_1}$  to the pretrained weight  $\mathbf{W}^0$ , so the input  $\mathbf{x} \in \mathbb{R}^{d_1}$  to a linear layer after adaptation becomes  $\mathbf{z} = (\mathbf{R}\mathbf{W}^0)^\top \mathbf{x}$ .  $\mathbf{R}$  is parameter-efficient because it is a block-diagonal matrix with  $n$  blocks as  $\mathbf{R} = \text{diag}(\mathbf{R}_1, \dots, \mathbf{R}_i, \dots, \mathbf{R}_n)$ , where each block  $\mathbf{R}_i \in \mathbb{R}^{w \times w}$  has a dimension  $w = d_1/n$ . To maintain orthogonality,  $\mathbf{R}_i$  is derived using Cayley parameterization:  $\mathbf{R}_i = (\mathbf{I} + \mathbf{Q}_i)(\mathbf{I} - \mathbf{Q}_i)^{-1}$  with  $\mathbf{Q}_i \in \mathbb{R}^{w \times w}$  being a skew-symmetric matrix ( $\mathbf{Q}_i = -\mathbf{Q}_i^\top$ ). In sum,  $\{\mathbf{Q}_i\}_{i=1}^n$  serve as the trainable parameters and  $\mathbf{R}$  is constructed from them with Cayley parameterization. Subsequent advancement, BOFT [30], leverages butterfly factorization to further refine OFT’s parameter efficiency. However, both OFT and BOFT, due to their reliance on matrix inversions in the Cayley parameterization and increased storage of intermediate activations, necessitate additional GPU memory and increase training duration compared to other PEFT approaches. Conversely, RoAd, which may be considered as a specialized case of OFT with  $w = 2$ , offers a faster and more memory-efficient solution by inherently maintaining orthogonality without requiring further parameterization.

## 2.2 Batching

Batching in this context refers to processing multiple heterogeneous requests, each requiring different adapters<sup>2</sup> for inference. This scenario commonly arises when serving personalized or task-specific LLMs. Specifically, we consider a setup where distinct adapters instead of a shared adapter are finetuned for various tasks to achieve optimal performance. During inference, each request in a batch pertains to a different task and necessitates a unique adapter.

Consider that we have finetuned distinct LoRA modules for  $b$  tasks, denoted as  $\{\mathbf{A}_i, \mathbf{B}_i\}_{i=1}^b$ . For a batch of  $b$  requests represented as  $\mathbf{X} \in \mathbb{R}^{b \times l \times d_1}$ , where  $l$  is the maximum sequence length across the requests, each request requires a different LoRA module. To exploit the parallel processing capabilities of GPUs, the output  $\mathbf{Z}$  of a linear layer can be computed as follows: First, the output from the pretrained layer is computed as  $\mathbf{Z}^0 = \text{torch.mm}(\mathbf{X}, \mathbf{W}^0)$ . Subsequently, the intermediate output from the first low-rank matrix,  $\hat{\mathbf{B}} \in \mathbb{R}^{b \times d_1 \times r}$  (a concatenation of  $\{\mathbf{B}_i\}_{i=1}^b$ ), is obtained as  $\mathbf{Z}_0^1 = \text{torch.bmm}(\mathbf{X}, \hat{\mathbf{B}})$ . The output from the second low-rank matrix,  $\hat{\mathbf{A}} \in \mathbb{R}^{b \times r \times d_2}$  (a concatenation of  $\{\mathbf{A}_i\}_{i=1}^b$ ), follows as  $\mathbf{Z}^1 = \text{torch.bmm}(\mathbf{Z}_0^1, \hat{\mathbf{A}})$ . Finally, these outputs are summed to produce  $\mathbf{Z} = \mathbf{Z}^0 + \mathbf{Z}^1$ . It is noteworthy that batch matrix multiplication (BMM), as implemented in *torch.bmm*, often introduces substantial overhead [1], reducing throughput and increasing latency, which adversely impacts user experience in time-sensitive applications.

In contrast, prompt-based methods circumvent the use of BMM by appending trainable tokens to each request, simplifying the computational process. However, prompt-based methods with long prompt tokens are difficult to optimize, which degrades performance compared to other PEFTs [14, 15]. (IA)<sup>3</sup> [25] proposes adapting LLM by multiplying the output from a linear layer with a trainable vector, involving only element-wise multiplication for efficient batching. A recent development, FLoRA [58], builds on (IA)<sup>3</sup> by employing two low-rank matrices while maintaining element-wise operations. Although our proposed method, RoAd, requires BMM, its sparse structure allows a reformulation of BMM and results in an overhead equivalent to element-wise multiplication.

## 2.3 Intervention and composability

Numerous studies [10, 11, 37, 38, 40] have provided support for the linear representation hypothesis [35, 46, 49] that concepts are represented within linear subspaces of neural network representations. To examine if a concept is captured within a linear subspace of a representation, Geiger et al. [11]

<sup>2</sup>Adapter here means the trained parameters since LoRA’s architecture is also similar to an adapter.

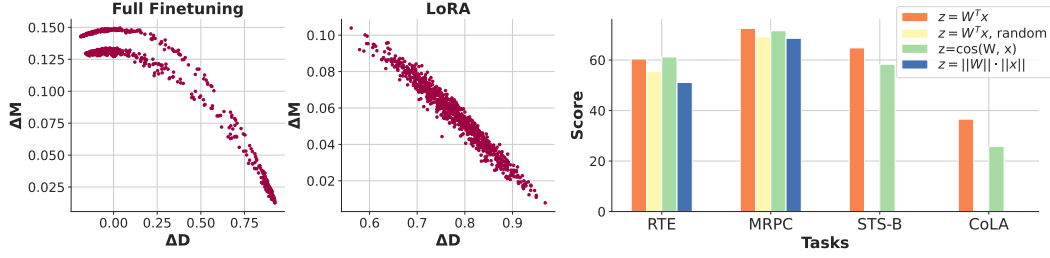


Figure 2: Pilot study for the pretrained and finetuned representations. **Left & Middle:** The change in magnitude and angle of representations between pretrained and finetuned LLM using full finetuning or LoRA. **Right:** The disentanglement experiment of magnitude and angle of pretrained representation.

suggests employing a distributed interchange intervention (DII) defined as:

$$\text{DII}(\mathbf{b}, \mathbf{s}, \mathbf{R}) = \mathbf{b} + \mathbf{R}^\top (\mathbf{R}\mathbf{s} - \mathbf{R}\mathbf{b}) \quad (1)$$

$\mathbf{b}$  denotes the hidden representation generated at row  $i$  and column  $k$  when the model processes an input, while  $\mathbf{s}$  represents the corresponding representation when the model processes a different input. The matrix  $\mathbf{R} \in \mathbb{R}^{r \times d_1}$ , consisting of orthogonal rows, serves as a low-rank projection matrix where  $d_1$  is the dimension of the representation and  $r$  is the subspace dimension under intervention. Equation (1) illustrates the application of a DII to  $\mathbf{b}$  using a counterfactual source representation  $\mathbf{s}$ .<sup>3</sup>

Drawing inspiration from this established framework, a recent study, LoReFT [61], introduces a method for finetuning specific positions of the representations to adapt LLM. This study further demonstrates that several prior approaches of representation editing [54, 60, 67] can be effectively integrated within this framework. Interestingly, the application of RoAd to representations can also be conceptualized as DII, offering interpretability potential. To demonstrate one aspect of interpretability for RoAd, we primarily conduct a qualitative experiment focused on task composition. This experiment involves combining the weights of models trained on distinct tasks to showcase the capability for multitasking learning without the need for additional adaptation [16, 20, 61, 64, 66].

### 3 Method

In this section, we first perform two pilot studies to ascertain the key factor influencing the adaptation of LLMs. Following this, we present our proposed method, the 2D rotary adaptation (RoAd), which serves as an effective PEFT method addressing the various challenges outlined in Section §2.

#### 3.1 Pilot study

**Study 1: Variations in magnitude and angular displacement.** Assume  $\mathbf{x}^0, \mathbf{x} \in \mathbb{R}^{d_1}$  are representations of the same token from a pretrained and finetuned LLM, respectively. We define the relative change in magnitude as  $\Delta M = \frac{||\mathbf{x}||_2 - ||\mathbf{x}^0||_2}{||\mathbf{x}^0||_2}$  and compute the angular displacement as  $\Delta D = \cos(\mathbf{x}, \mathbf{x}^0) \in [-1, 1]$ . A larger  $\Delta M$  and a smaller  $\Delta D$  indicate more significant changes in magnitude and angular displacement, respectively. Our study involves: (1) finetuning RoBERTa-base [31] on the SST-2 task [50] using either full finetuning or LoRA; (2) extracting representations  $\mathbf{x}^0$  and  $\mathbf{x}$  from the output of the second-last Transformer block for the [CLS] token across all samples in the development set, followed by computing  $\Delta M$  and  $\Delta D$ .<sup>4</sup> As depicted in Figure 2 (Left and Middle), there is a more pronounced change in  $\Delta D$  than in  $\Delta M$  for both full finetuning and LoRA.<sup>5</sup>

**Study 2: Disentanglement of magnitude and angle.** To ascertain whether angular or magnitude adjustments are more critical for finetuning, we implement a disentanglement study. This involves freezing RoBERTa-base and appending a two-layer classifier on top of it. The first layer of this

<sup>3</sup>We adopt notation systems from Wu et al. [61].

<sup>4</sup>Please refer to Figure B.1 for all layers.

<sup>5</sup>There are two other interesting observations: (1) An increase in magnitude change correlates with a larger angular displacement; (2) Compared to LoRA, full finetuning has a bigger change in magnitude and angle (for all layers, see Figure B.1), which is in line with a recent finding that LoRA learns less and forgets less [2].

classifier incorporates a weight matrix  $\mathbf{W} \in \mathbb{R}^{d_1 \times d_1}$ . Under standard operations, the output from this layer is computed as  $\mathbf{z} = \mathbf{W}^\top \mathbf{x}^0$ . To distinctly evaluate the impacts of magnitude and angle, we modify the output to retain only the magnitude component as  $z_i = \|\mathbf{W}_{:,i}\|_2 \cdot \|\mathbf{x}^0\|_2$ , or solely the angular component as  $z_i = \cos(\mathbf{W}_{:,i}, \mathbf{x}^0)$  ( $z_i$  is the  $i^{\text{th}}$  element of  $\mathbf{z}$ ). The modified classifier was then finetuned on four GLUE tasks with different metrics detailed in Table C.1. Additionally, a weak baseline employing a randomly initialized RoBERTa-base is included. As shown in Figure 2 (Right), angular information is paramount in finetuning, whereas reliance solely on magnitude information even leads to inferior results compared to the random backbone.

Both studies indicate that angular information is more crucial than magnitude information for adapting a pretrained LLM to a downstream task. However, rotating the entire  $d_1$  dimensions of the representation for finetuning incurs substantial computational costs. These costs are primarily reflected in a large number of trainable parameters, necessitating a dense matrix  $\mathbf{R} \in \mathbb{R}^{d_1 \times d_1}$ , and in the requirement to maintain its orthogonality. Could we only rotate a subspace of the representation and design a  $\mathbf{R}$  that is always orthogonal without any parameterization as OFT [44]? The first idea that comes to our mind is 2D rotation which only rotates two dimensions at a time and inherently maintains orthogonality.

### 3.2 2D rotary adaptation

Suppose that  $\mathbf{W}^0 \in \mathbb{R}^{d_1 \times d_2}$  is the pretrained weight of a linear layer,  $\mathbf{x} \in \mathbb{R}^{d_1}$  is the input of a token to this linear layer,  $\mathbf{R} \in \mathbb{R}^{d_2 \times d_2}$  is the rotation matrix, the adapted output from the linear layer is  $\mathbf{z} = \mathbf{R}\mathbf{h} = \mathbf{R}(\mathbf{W}^{0\top} \mathbf{x})$ . The rotation matrix  $\mathbf{R}$  is defined as follows:

$$\mathbf{R} = \text{diag}(\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_{d_2/2}) \quad \text{with} \quad \mathbf{R}_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix} \quad (2)$$

The trainable parameters are denoted as  $\{\theta_i\}_{i=1}^{d_2/2}$ . This 2D rotary adaptation involves rotating pairs of adjacent dimensions of  $\mathbf{h}$ , specifically dimensions  $2i-1$  and  $2i$ , using the rotation matrix  $\mathbf{R}_i$ .<sup>6</sup> The rotation matrix  $\mathbf{R}$  is characterized by its parameter efficiency, which is attributed to its sparse structure and the parameter sharing within each block  $\mathbf{R}_i$ . Additionally,  $\mathbf{R}$  can be integrated directly into the existing pretrained weights, forming  $\mathbf{W} = \mathbf{W}^0 \mathbf{R}^\top$ , which does not incur additional computational costs during inference. This design closely mirrors RoPE [51], with the notable difference that in our RoAd,  $\theta_i$  is trainable and  $\mathbf{R}_i$  does not incorporate positional information. The overview of RoAd is shown in Figure 3.

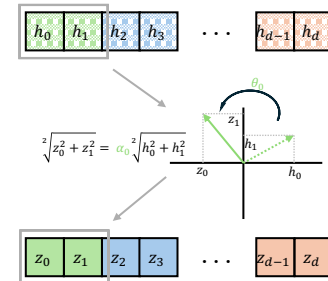


Figure 3: Overview of RoAd<sub>1</sub>.

**Relaxation to orthogonality.** Referring to Figure 2 (Right), while reliance predominantly on angular information substantially outperforms reliance on magnitude information, it remains less effective than using both angular and magnitude information for the tasks of MRPC, STS-B, and CoLA. Furthermore, both fully- and LoRA-finetuned LLMs exhibit slight adaptations in magnitude, as depicted in Figure 2 (Left and Middle). Consequently, we modify  $\mathbf{R}_i$  by incorporating  $\alpha_i$  to regulate the magnitude. We define a general  $\mathbf{R}_i$  as follows:

$$\mathbf{R}_i = \begin{bmatrix} \alpha_{i,11} \cos \theta_{i,11} & -\alpha_{i,12} \sin \theta_{i,12} \\ \alpha_{i,21} \sin \theta_{i,21} & \alpha_{i,22} \cos \theta_{i,22} \end{bmatrix} \quad (3)$$

We develop three variants of RoAd by altering the configuration of shared parameters as outlined in Table 1. RoAd<sub>1</sub> introduces a minimal change to Equation (2) by incorporating a scaling factor  $\alpha_i$ . RoAd<sub>1</sub> already shows impressive results for most tasks in Section §4.1. For some knowledge-intensive tasks, we observe that RoAd<sub>2</sub> and RoAd<sub>4</sub> obtain better results with more trainable parameters. To preserve the starting point of LLMs [23], we always initialize  $\alpha_i = 1$  and  $\theta_i = 0$ .

**Batching.** In practice, we don't need to save  $\mathbf{R}$  as a sparse matrix and do matrix multiplication. Taking RoAd<sub>1</sub> as an example in Equation (4), we only save two vectors:  $\mathbf{R}^1$  and  $\mathbf{R}^2$ . Then  $\mathbf{z} = \mathbf{R}\mathbf{h} = \mathbf{R}^1 \otimes \mathbf{h} + \mathbf{R}^2 \otimes \hat{\mathbf{h}}$ , where  $\hat{\mathbf{h}}$  is a rearranged version of  $\mathbf{h}$  and  $\otimes$  denotes element-wise multiplication. This reformulation not only simplifies the representation of  $\mathbf{R}$  but also enhances the efficiency of batching in RoAd, relying solely on element-wise multiplications rather than BMM.

<sup>6</sup>The index in this work starts from 1 instead of 0.

Table 1: A summarization of three RoAd variants.

RoAd?	$\alpha_i$	$\theta_i$	#Trainable
1	$\alpha_{i,11} = \alpha_{i,12} = \alpha_{i,21} = \alpha_{i,22} = \alpha_i$	$\theta_{i,11} = \theta_{i,12} = \theta_{i,21} = \theta_{i,22} = \theta_i$	$d_2$
2	$\alpha_{i,11} = \alpha_{i,12} \quad \alpha_{i,21} = \alpha_{i,22}$	$\theta_{i,11} = \theta_{i,12} \quad \theta_{i,21} = \theta_{i,22}$	$2d_2$
4	$\alpha_{i,11} \neq \alpha_{i,12} \neq \alpha_{i,21} \neq \alpha_{i,22}$	$\theta_{i,11} \neq \theta_{i,12} \neq \theta_{i,21} \neq \theta_{i,22}$	$4d_2$

$$\begin{aligned}
z = Rh &= R^1 \otimes h + R^2 \otimes \hat{h} \\
&= \begin{bmatrix} \alpha_1 \cos \theta_1 \\ \alpha_1 \cos \theta_1 \\ \alpha_2 \cos \theta_2 \\ \alpha_2 \cos \theta_2 \\ \vdots \\ \alpha_{d_2/2} \cos \theta_{d_2/2} \\ \alpha_{d_2/2} \cos \theta_{d_2/2} \end{bmatrix} \otimes \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ \vdots \\ h_{d_2-1} \\ h_{d_2} \end{bmatrix} + \begin{bmatrix} \alpha_1 \sin \theta_1 \\ \alpha_1 \sin \theta_1 \\ \alpha_2 \sin \theta_2 \\ \alpha_2 \sin \theta_2 \\ \vdots \\ \alpha_{d_2/2} \sin \theta_{d_2/2} \\ \alpha_{d_2/2} \sin \theta_{d_2/2} \end{bmatrix} \otimes \begin{bmatrix} -h_2 \\ h_1 \\ -h_4 \\ h_3 \\ \vdots \\ -h_{d_2} \\ h_{d_2-1} \end{bmatrix} \quad (4)
\end{aligned}$$

**Composability.** RoAd can be incorporated into the DII framework as  $\Phi(h) = Rh = h + R(h - R^\top h)$ , with  $Rs$  in Equation (1) being set to  $h$ . Although a degree of relaxation is introduced to the orthogonality of  $R$ , it is important to note that the rows of  $R$  remain orthogonal to each other within non-adjacent segments of the same block,  $R_i$ . This offers a possibility for composability. We can finetune some rows on one task and other orthogonal rows on another task. Since they are orthogonal to each other, these two tasks should minimally affect each other, and the combination of these rows after finetuning could bring new multitasking learning ability.

RoAd can be considered as a special case of OFT [44] with  $w = 2$ . However, it is much more parameter- and memory-efficient and faster. Please refer to Section §D.1 for a detailed discussion.

## 4 Experiments

In this section, we begin by implementing RoAd to finetune various LLMs across three benchmarks. Subsequently, we illustrate its efficiency in batching processes and demonstrate its composability. Unless otherwise noted, RoAd is applied to all linear layers within the LLMs. All of our experiments are conducted on A100 80GB GPU with the frameworks, Transformers [59] and PEFT [34].

### 4.1 Results on downstream tasks

**Natural language understanding (NLU).** We evaluate the effectiveness of RoAd on the GLUE benchmark [56] for its ability of NLU with RoBERTa [31] as the backbone. Unlike many previous works [14, 22, 23, 31, 65] that employ the GLUE development sets for both validation and testing, here we partition the development set into distinct validation and test subsets to mitigate the risk of overfitting. For comprehensive information regarding the split of the development set, the search space of hyperparameters, the optimal hyperparameter configurations, and other details crucial for reproducibility, please see Section §C.1.

As shown in Table 2, RoAd<sub>1</sub> outperforms all other PEFT methods with  $< 0.1\%$  trainable parameters for both sizes of RoBERTa on average, being the only PEFT method that matches or outperforms full finetuning. These results show that 2D rotation (with a few scaling) can efficiently adapt LLM.

**Commonsense reasoning.** In assessing the capacity of LLaMA [52] for commonsense reasoning, we focus on eight representative tasks: BoolQ [4], PIQA [3], SIQA [48], HellaSwag [63], WinoGrande [47], ARC-e, ARC-c [5], and OBQA [36]. The setting here contrasts with the NLU experiments where each task involves finetuning a separate LLM. Instead, we adopt a unified strategy by finetuning a single LLM across all tasks as delineated in Hu et al. [15]. Such a setting is designed to mitigate overfitting and aligns more closely with real-world applications. Specifically, the training and test sets from these eight tasks are reformulated according to a predefined template, so all tasks can be trained or evaluated in a generative way. For all finetuning experiments on LLaMA, we follow a recipe in Table C.5 without extensive searching. Please see Section §C.2 for more training details.

Table 2: Results on the held-out GLUE development set with RoBERTa as the backbone. We report matched accuracy for MNLI, Matthew’s correlation for CoLA, Pearson correlation for STS-B and accuracy for other tasks. The **best** and second-best results are in bold and underlined, respectively, being the same for other tables. The percentage of trainable parameters is calculated without considering the classifier head. RoAd<sub>1</sub>(fc1) means that we only insert the RoAd<sub>1</sub> module to the first feed-forward layer, to match the #Params. of RED and LoReFT. Results of methods denoted by \* and <sup>◊</sup> are from Wu et al. [60] and Wu et al. [61], respectively. Otherwise, average results from three random runs are reported. Refer to Table C.4 for the standard deviation.

Model	Method	#Params.	RTE	MRPC	STS-B	CoLA	SST-2	QNLI	QQP	MNLI	Avg.
base	Full FT*	100.00%	78.3	87.9	90.6	62.4	94.4	92.5	91.7	87.3	85.6
	Adapter*	0.32%	76.5	88.4	<b>90.5</b>	60.9	93.3	<u>92.5</u>	<b>90.5</b>	<u>87.0</u>	85.0
	LoRA*	0.24%	75.3	88.7	90.3	59.7	93.9	<b>92.6</b>	<u>90.4</u>	86.6	84.7
	Adapter <sup>FNN</sup> *	0.24%	77.7	88.8	<u>90.4</u>	58.5	93.0	92.0	90.2	<b>87.1</b>	84.7
	BOFT	0.16%	71.4	87.5	89.6	55.3	92.5	91.4	89.4	85.3	82.8
	OFT <sub>w=2</sub>	0.10%	74.4	87.6	89.4	50.4	92.8	90.9	89.2	83.9	82.3
	BitFit*	0.08%	69.8	88.0	89.5	54.0	<u>94.0</u>	91.0	87.3	84.7	82.3
	(IA) <sup>3</sup>	0.04%	75.3	87.1	90.0	60.4	<u>94.0</u>	91.8	89.2	85.8	84.2
	RED*	0.02%	78.0	<u>89.2</u>	90.4	<u>61.0</u>	93.9	90.7	87.2	83.9	84.3
	LoReFT <sup>◊</sup>	0.02%	<u>79.0</u>	<u>89.2</u>	90.0	60.4	93.4	91.2	87.4	83.1	84.2
	<b>RoAd<sub>1</sub></b>	0.07%	78.9	<u>89.2</u>	<b>90.5</b>	<b>64.4</b>	93.9	91.9	89.6	86.3	<b>85.6</b>
	<b>RoAd<sub>1</sub>(fc1)</b>	0.03%	<b>79.1</b>	<b>90.2</b>	90.2	60.9	<b>94.6</b>	91.6	88.7	85.4	<u>85.1</u>
large	Full FT*	100.00%	85.8	91.7	92.6	68.2	96.0	93.8	91.5	88.8	88.6
	Adapter*	0.25%	85.3	90.5	91.5	65.4	95.2	<u>94.6</u>	<b>91.4</b>	<u>90.1</u>	88.0
	LoRA*	0.23%	86.3	89.8	<u>91.7</u>	65.5	96.0	<b>94.7</b>	90.7	<u>90.1</u>	88.1
	Adapter <sup>FNN</sup> *	0.23%	84.8	90.5	90.2	64.4	96.1	94.3	91.3	<b>90.3</b>	87.7
	RED*	0.01%	86.2	90.3	91.3	<b>68.1</b>	96.0	93.5	88.8	89.5	88.0
	LoReFT <sup>◊</sup>	0.01%	87.5	90.1	91.6	<u>68.0</u>	<u>96.2</u>	94.1	88.5	89.2	88.2
	<b>RoAd<sub>1</sub></b>	0.06%	<b>89.2</b>	<u>91.0</u>	<u>91.7</u>	66.1	<b>96.3</b>	94.4	91.0	89.7	88.7
	<b>RoAd<sub>1</sub>(fc1)</b>	0.03%	<u>88.7</u>	<b>91.5</b>	<b>91.9</b>	<b>68.1</b>	96.1	94.5	90.2	89.6	<b>88.8</b>

As shown in Table 3, RoAds still perform the best across various PEFT methods for both LLaMA-7B and LLaMA-13B on average. The strong baseline to RoAd is a recent representation finetuning method, LoReFT [61], 80.2 vs. 79.2 and 83.3 vs. 83.0 for RoAd<sub>1</sub> for LLaMA-7B and LLaMA-13B, respectively. With a slightly increasing number of trainable parameters from RoAd<sub>1</sub> to RoAd<sub>2</sub> or RoAd<sub>4</sub>, RoAd matches or outperforms LoReFT. The same story is also told for another two versions of LLaMA, i.e. LLaMA2 [53] and LLaMA3, in Table D.2.

**Arithmetic reasoning.** To assess the arithmetic reasoning ability of LLMs, we evaluate the finetuned LLMs on the test sets of four tasks: AQuA [24], GSM8K [6], MAWPS [18] and SVAMP [41]. Similar to the commonsense reasoning tasks, we finetune a single LLM for all four arithmetic reasoning tasks. The training dataset is Math10K [15] which is constructed from the training sets of GSM8K, MAWPS, MAWPS-single and AQuA. The training recipe is similar to the one used for commonsense reasoning as shown in Table C.5. Please see Section §C.3 for more training details.

Different from the results of NLU and commonsense reasoning tasks, RoAd doesn’t always perform the best on the arithmetic reasoning tasks, as shown in Figure 4. For the smaller-size LLM, LLaMA-7B, RoAd is significantly better than other PEFT methods with < 0.1% trainable parameters, but worse than LoRA and Adapter<sup>P</sup> with more than 10× trainable parameters. However, for the larger-size LLM, LLaMA-13B, all RoAd variants are better than other PEFT methods, which shows its scalability and potentially implies even better results for larger LLMs.

Observed from the above-mentioned results, for enhanced performance on downstream tasks and if a marginal increase in the storage capacity for trained parameters is acceptable, RoAd<sub>4</sub> is the preferable option. Conversely, if the objective is to investigate how the model adjusts in terms of angle and magnitude, RoAd<sub>1</sub> is recommended. Notably, all variants of RoAd incur the same computational overhead for batching.

Table 5: Score on AlpacaEval2.0 with LLaMA2-7B.

Method	#Params.	Finetuning Data	Win Rate (%)
LoRA	0.83%	10K cleaned Alpaca	61.55
LoReFT	0.03%	10K cleaned Alpaca	60.21
RoAd <sub>1</sub>	0.02%	10K cleaned Alpaca	<b>62.64</b>
LoReFT	0.03%	UltraFeedback [7]	61.68
RoAd <sub>1</sub>	0.02%	UltraFeedback	<b>62.60</b>

Table 3: Accuracy of LLaMA on eight commonsense reasoning tasks. Results of methods denoted by \*,  $\diamond$  and  $\circ$  are from [15], [61] and [27], respectively. Otherwise, average results from three random runs are reported. Refer to Table C.6 for the standard deviation. Refer to Table D.2 for LLaMA2&3.

Model	Method	#Paras.	BoolQ	PIQA	SIQA	HellaS.	WinoG.	ARC-e	ARC-c	OBQA	Avg.
GPT3.5*	-	-	73.1	85.4	68.5	78.5	66.1	89.8	79.9	74.8	77.0
7B	Adapter <sup>P*</sup>	3.54%	67.9	76.4	78.8	69.8	78.9	73.7	57.3	75.2	72.3
	Adapter <sup>S*</sup>	0.99%	63.0	79.2	76.3	67.9	75.7	74.5	57.1	72.4	70.8
	DoRA $\circ$	0.84%	68.5	82.9	<u>79.6</u>	84.8	80.8	81.4	65.8	<u>81.0</u>	78.1
	LoRA*	0.83%	68.9	80.7	77.4	78.1	78.8	77.8	61.3	74.8	74.7
	OFT	0.14%	69.0	82.0	78.5	90.9	78.9	83.0	68.2	76.4	78.4
	Prefix*	0.04%	64.3	76.8	73.9	42.1	72.1	72.9	54.0	60.6	64.6
	LoReFT $\diamond$	0.03%	69.3	<b>84.4</b>	<b>80.3</b>	<b>93.1</b>	<b>84.2</b>	83.2	68.2	78.9	<b>80.2</b>
	(IA) <sup>3</sup>	0.02%	67.8	81.7	78.1	89.9	81.1	80.5	65.4	77.8	77.8
	<b>RoAd<sub>4</sub></b>	0.08%	<b>70.6</b>	<u>83.2</u>	79.0	<u>92.3</u>	<u>81.8</u>	<u>84.2</u>	<b>70.6</b>	80.0	<b>80.2</b>
	<b>RoAd<sub>2</sub></b>	0.04%	70.3	82.6	79.2	92.0	<u>81.8</u>	<b>84.8</b>	<u>68.8</u>	<b>82.2</b>	<b>80.2</b>
	<b>RoAd<sub>1</sub></b>	0.02%	<u>70.4</u>	81.9	79.0	91.4	80.3	84.0	68.7	77.8	<u>79.2</u>
13B	Adapter <sup>P*</sup>	2.89%	72.5	84.9	79.8	92.1	84.7	84.2	71.2	82.4	81.5
	Adapter <sup>S*</sup>	0.80%	71.8	83.0	79.2	88.1	82.4	82.5	67.3	81.8	79.5
	DoRA $\circ$	0.68%	72.4	84.9	81.5	92.4	84.2	84.2	69.6	82.8	81.5
	LoRA*	0.67%	72.1	83.5	80.5	90.5	83.7	82.8	68.3	82.4	80.5
	Prefix*	0.03%	65.3	75.4	72.1	55.2	68.6	79.5	62.9	68.0	68.4
	LoReFT $\diamond$	0.03%	72.1	<u>86.3</u>	81.8	<b>95.1</b>	<b>87.2</b>	86.2	73.7	84.2	83.3
	<b>RoAd<sub>4</sub></b>	0.07%	<u>73.2</u>	<b>85.5</b>	<b>82.4</b>	<u>94.5</u>	<u>86.3</u>	<u>86.8</u>	<b>74.6</b>	86.0	<u>83.7</u>
	<b>RoAd<sub>2</sub></b>	0.03%	<b>73.3</b>	<b>86.4</b>	<u>82.0</u>	<u>94.4</u>	86.1	<b>87.4</b>	<b>74.1</b>	<b>87.0</b>	<b>83.8</b>
	<b>RoAd<sub>1</sub></b>	0.02%	72.2	85.1	81.2	94.1	84.4	86.6	73.7	<u>86.6</u>	83.0

Table 4: Accuracy of LLaMA on four arithmetic reasoning tasks. Results of methods denoted by \* and  $\diamond$  are from [15] and [61], respectively. Refer to Table C.7 for the standard deviation.

Model	Method	#Params.	AQuA	GSM8K	MAWPS	SVAMP	Avg.
7B	Adapter <sup>P*</sup>	3.54%	18.1	<u>35.3</u>	<b>82.4</b>	49.6	<u>46.4</u>
	Adapter <sup>S*</sup>	0.99%	15.0	33.3	77.7	<b>52.3</b>	44.6
	LoRA*	0.83%	18.9	<b>37.5</b>	79.0	<u>52.1</u>	<b>46.9</b>
	Prefix*	0.04%	14.2	24.4	63.4	38.1	35.0
	LoReFT $\diamond$	0.03%	21.4	26.0	76.2	46.8	42.6
	(IA) <sup>3</sup>	0.02%	19.7	28.8	76.9	48.5	43.5
	<b>RoAd<sub>4</sub></b>	0.08%	24.8	27.4	<u>81.5</u>	49.4	45.8
	<b>RoAd<sub>2</sub></b>	0.04%	<b>26.8</b>	29.9	78.6	49.3	46.2
	<b>RoAd<sub>1</sub></b>	0.02%	<b>26.4</b>	26.2	76.5	46.7	44.0
13B	Adapter <sup>P*</sup>	2.89%	20.5	43.3	81.1	55.7	50.2
	Adapter <sup>S*</sup>	0.80%	22.0	<u>44.0</u>	78.6	50.8	48.9
	LoRA*	0.67%	18.5	<b>47.5</b>	83.6	54.6	51.1
	Prefix*	0.03%	15.7	31.1	66.8	41.4	38.8
	LoReFT $\diamond$	0.03%	23.6	38.1	82.4	54.2	49.6
	<b>RoAd<sub>4</sub></b>	0.07%	<u>25.2</u>	39.8	<u>84.5</u>	<b>59.5</b>	<b>52.3</b>
	<b>RoAd<sub>2</sub></b>	0.03%	<b>26.0</b>	40.6	84.0	<u>58.3</u>	<u>52.2</u>
	<b>RoAd<sub>1</sub></b>	0.02%	24.8	40.7	<b>84.9</b>	57.3	51.9

**Instruction-following ability.** We further benchmark RoAd using AlpacaEval2.0 [9]. We finetune LLaMA2-7B with two instruction-tuning datasets and evaluate the model using AlpacaEval2.0. This evaluation employs GPT-4 [39] to assess the responses generated by the finetuned model against those produced by Text-davinci-003. We don't choose GPT-4 as the reference model, because GPT-4 is too powerful than LLaMA2-7B. The proof-of-concept experiment with LoRA shows the win-rate < 5%. As shown in Table 5, RoAd<sub>1</sub> demonstrates superior performance compared to all baselines, while utilizing the least number of trainable parameters.

**Multimodal ability.** Lastly, we apply RoAd to the LLM backbone of LLaVA [26]. Liu et al. [26] requires 4.61% trainable parameters for LoRA on this task, while most tasks with LoRA in our paper need < 1%, showing that this task is knowledge-intensive. Therefore, we need to scale RoAd's trainable parameters. For this purpose, we combine it with LoRA due to the limited number of  $\theta_i$  and  $\alpha_i$  in  $\mathbf{R}$ . The combination is represented as  $\mathbf{z} = (\mathbf{R}\mathbf{W}^{0\top} + (\mathbf{B}\mathbf{A})^\top)\mathbf{x}$ , where  $\mathbf{A}$  and  $\mathbf{B}$  are from LoRA. We adjust the LoRA rank to vary the number of trainable parameters. We combine RoAd<sub>1</sub>



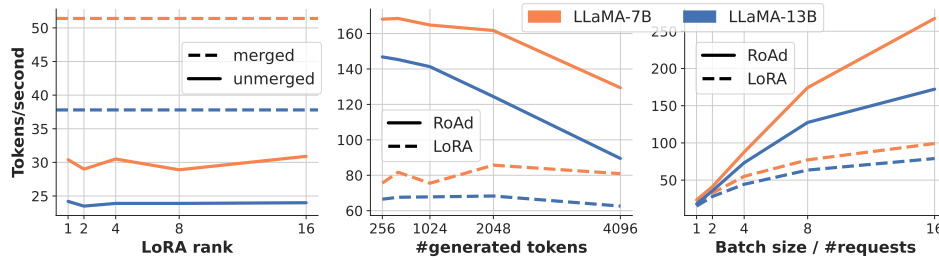


Figure 4: Comparison of throughput between LoRA and RoAd. **Left:** The influence of weight merging for LoRA. **Middle:** The influence of the number of generated tokens. **Right:** The influence of the number of heterogeneous requests in a batch.

with LoRA, but not RoAd<sub>2</sub> or RoAd<sub>4</sub>, as their primary design purpose is to increase the number of trainable parameters.

As shown in Table 6, with only 0.08% trainable parameters, RoAd<sub>4</sub> already achieves 96.9% of the accuracy of LoRA with 4.61% trainable parameters. By combining RoAd<sub>1</sub> with LoRA, we achieve the same performance as LoRA with only 1/4 of its trainable parameters. This demonstrates RoAd’s excellent scalability when combined with LoRA.

Table 6: Visual instruction tuning results on LLaVA1.5-7B.

Method	#Params.	GQA	SQA	VQAT	POPE	Avg.
LoRA	4.61%	62.4	<b>68.5</b>	56.9	<b>86.0</b>	<b>68.5</b>
RoAd <sub>4</sub>	0.08%	60.0	66.9	53.3	85.5	66.4
RoAd <sub>1</sub> + LoRA	1.19%	<b>62.5</b>	68.2	<b>57.4</b>	85.8	<b>68.5</b>

## 4.2 Efficiency results for batching

We commence by highlighting the significance of weight merging for PEFT. Among the approaches discussed in Section §4.1, only LoRA [14], DoRA [27], BOFT [30], OFT [44], BitFit [62], (IA)<sup>3</sup> [25], and our proposed RoAd enable the integration of trainable parameters with pretrained parameters without incurring additional inference overhead. As an illustration, we consider LoRA both with and without weight merging to underscore this process’s importance. Notably, the implementation of LoRA with merged weights effectively reverts to the original LLM. To assess throughput, we configure the system with a batch size of 1, generate 2048 tokens, and apply the LoRA modules across all linear layers. Figure 4 (Left) clearly illustrates that the unmerged LoRA exhibits a significantly smaller throughput compared to the merged LoRA. Additionally, it is evident that the throughput of the unmerged LoRA demonstrates only a weak correlation with the rank size, primarily due to the fact that the additional overhead is largely attributed to communication instead of computation.

Furthermore, to evaluate the throughput of batching, we establish a default batch size of 8, generate 2048 tokens, and set the LoRA rank to 8. Each request within the batch is heterogeneous, necessitating eight distinct sets of trainable parameters by default. We only compare to LoRA here, because other baselines have either a weaker performance on downstream tasks (BOFT, OFT, BitFit and (IA)<sup>3</sup>) or a smaller throughput than LoRA for batching (DoRA). As shown in Figure 4 (Middle and Right), RoAd significantly outperforms LoRA with variations in either the number of generated tokens or the number of heterogeneous requests. With an increasing number of distinct requests, the gap between LoRA and RoAd becomes even larger, which shows RoAd’s unique advantage in efficient serving

## 4.3 Qualitative results for composability

In our investigation of RoAd’s ability to handle compositional tasks, we primarily engage in multilingual experiments similar to those conducted by Wu et al. [61]. We use two training datasets: a new version of HellaSwag [63]<sup>7</sup>, which comprises 1K samples with prompts in English and completions in German, and a 1K-sample subset of the Ultrafeedback [7] dataset, which focuses on instruction following tasks in English. Contrary to the above experiments that adapt the outputs of the linear layer, here we instead adapt the representations from the 16<sup>th</sup> block of LLaMA-7B, treating RoAd as a DII method. Specifically, we only adapt/intervene the representation of the final token in the

<sup>7</sup>[https://huggingface.co/datasets/LeoLM/HellaSwag\\_de](https://huggingface.co/datasets/LeoLM/HellaSwag_de)

LoReFT response for prompt: Do you know the conference NeurIPS?	
<b>German subspace:</b> geht auf dem Bildschirm vorbei und zeigt die Konferenz. (Translation: passes by on the screen and shows the conference.)	✗
<b>Instruction following subspace:</b> NeurIPS (Neural Information Processing Systems) is an annual conference on machine learning and computational neuroscience ...	✓
<b>Combined subspaces:</b> Es gibt eine Konferenz namens NeurIPS, die in der Vergangenheit stattgefunden hat. (Translation: There is a conference called NeurIPS that took place in the past.)	✗

RoAd response for prompt: Do you know the conference NeurIPS?	
<b>German subspace:</b> Ich weiß nicht, ob ich es kenne. (Translation: I don't know if I know it.)	✗
<b>Instruction following subspace:</b> Yes, I do! It's a great conference for machine learning researchers and practitioners to share their work with each other.	✓
<b>Combined subspaces:</b> NeurIPS ist ein internationales Forschungskongress, der sich mit dem Thema künstliche Intelligenz beschäftigt. (Translation: NeurIPS is an international research congress that deals with the topic of artificial intelligence.)	✓

Figure 5: Qualitative comparison between RoAd and LoReFT for their composability. The prompt is always in English for different subspaces. Refer to Figure D.1, D.2 and D.3 for more examples.

prompt using RoAd<sub>1</sub>. We train the upper half of  $\mathbf{R}$ , i.e.  $\{\mathbf{R}_i\}_{i=1}^{d_2/4}$ , to handle the German completions in HellaSwag, and another half to complete the English sentences in Ultrafeedback. Both tasks are simultaneously trained but utilize distinct subspaces of  $\mathbf{R}$ . We train the model over five epochs with a learning rate of  $5e-3$  and a batch size of 8.<sup>8</sup>

As in Figure 5, both LoReFT and RoAd are unable to perform completions with the German subspace. This limitation is anticipated due to two primary reasons: (1) LLaMA-7B predominantly relies on pretraining from English datasets, and doesn't have a cross-lingual answering ability without explicitly prompting. (2) The HellaSwag dataset is relatively small, containing only 1K samples with limited comprehensive coverage. Despite these constraints, the German subspace effectively prompts the model to produce sentences in German. Additionally, both methods achieve accurate completions in the other half of the subspaces, attributed to LLaMA-7B's extensive knowledge base in English. When these two subspaces are combined, RoAd successfully leverages their strengths, facilitating accurate sentence completions in German, while LoReFT doesn't catch the purpose of the prompt. We offer more examples, including negative examples, in Figure D.1, D.2 and D.3.

## 5 Conclusion

Initially, our research examines how finetuning modifies the representation of pretrained LLMs, finding that angular adjustments are more significant than changes in magnitude scale. Leveraging this insight, we propose a PEFT method, RoAd, which primarily utilizes a 2D rotational adjustment to the representation. Despite its simplicity, RoAd exhibits several distinct advantages: (1) It is exceptionally efficient in terms of parameters, consistently delivering superior performance on downstream tasks with the fewest trainable parameters compared to other PEFT methods; (2) RoAd efficiently supports batch processing, achieving twice the throughput of LoRA; (3) When incorporated within an intervention framework, RoAd demonstrates remarkable composability.

Due to page limit, we discuss the limitations and broader impacts in Section §A and §B, respectively.

## Acknowledgements

We thank eBay Inc. for the computation support. This research was funded in part by the Netherlands Organization for Scientific Research (NWO) under project number VI.C.192.080.

<sup>8</sup>The experiment is based on this notebook <https://github.com/stanfordnlp/pyreft/blob/main/examples/composition/compreft.ipynb>.

## References

- [1] A. Abdelfattah, A. Haidar, S. Tomov, and J. J. Dongarra. Performance, design, and auto-tuning of batched GEMM for gpus. In J. M. Kunkel, P. Balaji, and J. J. Dongarra, editors, *High Performance Computing - 31st International Conference, ISC High Performance 2016, Frankfurt, Germany, June 19-23, 2016, Proceedings*, volume 9697 of *Lecture Notes in Computer Science*, pages 21–38. Springer, 2016. doi: 10.1007/978-3-319-41321-1\_2. URL [https://doi.org/10.1007/978-3-319-41321-1\\_2](https://doi.org/10.1007/978-3-319-41321-1_2).
- [2] D. Biderman, J. G. Ortiz, J. Portes, M. Paul, P. Greengard, C. Jennings, D. King, S. Havens, V. Chiley, J. Frankle, C. Blakeney, and J. P. Cunningham. Lora learns less and forgets less, 2024.
- [3] Y. Bisk, R. Zellers, R. L. Bras, J. Gao, and Y. Choi. PIQA: reasoning about physical common-sense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6239. URL <https://doi.org/10.1609/aaai.v34i05.6239>.
- [4] C. Clark, K. Lee, M. Chang, T. Kwiatkowski, M. Collins, and K. Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2924–2936. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1300. URL <https://doi.org/10.18653/v1/n19-1300>.
- [5] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018. URL <http://arxiv.org/abs/1803.05457>.
- [6] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL <https://arxiv.org/abs/2110.14168>.
- [7] G. Cui, L. Yuan, N. Ding, G. Yao, W. Zhu, Y. Ni, G. Xie, Z. Liu, and M. Sun. Ultrafeedback: Boosting language models with high-quality feedback. *CoRR*, abs/2310.01377, 2023. doi: 10.48550/ARXIV.2310.01377. URL <https://doi.org/10.48550/arXiv.2310.01377>.
- [8] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1423. URL <https://doi.org/10.18653/v1/n19-1423>.
- [9] Y. Dubois, X. Li, R. Taori, T. Zhang, I. Gulrajani, J. Ba, C. Guestrin, P. Liang, and T. B. Hashimoto. AlpacaFarm: A simulation framework for methods that learn from human feedback, 2023.
- [10] N. Elhage, T. Hume, C. Olsson, N. Schiefer, T. Henighan, S. Kravec, Z. Hatfield-Dodds, R. Lasenby, D. Drain, C. Chen, R. Grosse, S. McCandlish, J. Kaplan, D. Amodei, M. Wattenberg, and C. Olah. Toy models of superposition. *CoRR*, abs/2209.10652, 2022. doi: 10.48550/ARXIV.2209.10652. URL <https://doi.org/10.48550/arXiv.2209.10652>.
- [11] A. Geiger, Z. Wu, C. Potts, T. Icard, and N. D. Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In F. Locatello and V. Didelez, editors, *Causal Learning and Reasoning, 1-3 April 2024, Los Angeles, California, USA*, volume 236 of *Proceedings of Machine Learning Research*, pages 160–187. PMLR, 2024. URL <https://proceedings.mlr.press/v236/geiger24a.html>.
- [12] J. He, C. Zhou, X. Ma, T. Berg-Kirkpatrick, and G. Neubig. Towards a unified view of parameter-efficient transfer learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL <https://openreview.net/forum?id=ORDcd5Axok>.

- [13] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. Parameter-efficient transfer learning for NLP. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 2019. URL <http://proceedings.mlr.press/v97/houlsby19a.html>.
- [14] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL <https://openreview.net/forum?id=nZeVKeeFYf9>.
- [15] Z. Hu, L. Wang, Y. Lan, W. Xu, E. Lim, L. Bing, X. Xu, S. Poria, and R. K. Lee. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5254–5276. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.319. URL <https://doi.org/10.18653/v1/2023.emnlp-main.319>.
- [16] C. Huang, Q. Liu, B. Y. Lin, T. Pang, C. Du, and M. Lin. Lorahub: Efficient cross-task generalization via dynamic lora composition. *CoRR*, abs/2307.13269, 2023. doi: 10.48550/ARXIV.2307.13269. URL <https://doi.org/10.48550/arXiv.2307.13269>.
- [17] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de Las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. Mistral 7b. *CoRR*, abs/2310.06825, 2023. doi: 10.48550/ARXIV.2310.06825. URL <https://doi.org/10.48550/arXiv.2310.06825>.
- [18] R. Koncel-Kedziorski, S. Roy, A. Amini, N. Kushman, and H. Hajishirzi. MAWPS: A math word problem repository. In K. Knight, A. Nenkova, and O. Rambow, editors, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1152–1157. The Association for Computational Linguistics, 2016. doi: 10.18653/V1/N16-1136. URL <https://doi.org/10.18653/v1/n16-1136>.
- [19] B. Lester, R. Al-Rfou, and N. Constant. The power of scale for parameter-efficient prompt tuning. In M. Moens, X. Huang, L. Specia, and S. W. Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3045–3059. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.243. URL <https://doi.org/10.18653/v1/2021.emnlp-main.243>.
- [20] M. Li, S. Gururangan, T. Dettmers, M. Lewis, T. Althoff, N. A. Smith, and L. Zettlemoyer. Branch-train-merge: Embarrassingly parallel training of expert language models. *CoRR*, abs/2208.03306, 2022. doi: 10.48550/ARXIV.2208.03306. URL <https://doi.org/10.48550/arXiv.2208.03306>.
- [21] X. L. Li and P. Liang. Prefix-tuning: Optimizing continuous prompts for generation. In C. Zong, F. Xia, W. Li, and R. Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.353. URL <https://doi.org/10.18653/v1/2021.acl-long.353>.
- [22] B. Liao, Y. Meng, and C. Monz. Parameter-efficient fine-tuning without introducing new latency. In A. Rogers, J. L. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 4242–4260. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.233. URL <https://doi.org/10.18653/v1/2023.acl-long.233>.
- [23] B. Liao, S. Tan, and C. Monz. Make pre-trained model reversible: From parameter to memory efficient fine-tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December*

- 10 - 16, 2023, 2023. URL [http://papers.nips.cc/paper\\_files/paper/2023/hash/3151e460c41ba67dc55412861184ef35-Abstract-Conference.html](http://papers.nips.cc/paper_files/paper/2023/hash/3151e460c41ba67dc55412861184ef35-Abstract-Conference.html).
- [24] W. Ling, D. Yogatama, C. Dyer, and P. Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In R. Barzilay and M. Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 158–167. Association for Computational Linguistics, 2017. doi: 10.18653/V1/P17-1015. URL <https://doi.org/10.18653/v1/P17-1015>.
- [25] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal, and C. Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL [http://papers.nips.cc/paper\\_files/paper/2022/hash/0cde695b83bd186c1fd456302888454c-Abstract-Conference.html](http://papers.nips.cc/paper_files/paper/2022/hash/0cde695b83bd186c1fd456302888454c-Abstract-Conference.html).
- [26] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL [http://papers.nips.cc/paper\\_files/paper/2023/hash/6dcf277ea32ce3288914faf369fe6de0-Abstract-Conference.html](http://papers.nips.cc/paper_files/paper/2023/hash/6dcf277ea32ce3288914faf369fe6de0-Abstract-Conference.html).
- [27] S. Liu, C. Wang, H. Yin, P. Molchanov, Y. F. Wang, K. Cheng, and M. Chen. Dora: Weight-decomposed low-rank adaptation. *CoRR*, abs/2402.09353, 2024. doi: 10.48550/ARXIV.2402.09353. URL <https://doi.org/10.48550/arXiv.2402.09353>.
- [28] W. Liu, R. Lin, Z. Liu, L. Liu, Z. Yu, B. Dai, and L. Song. Learning towards minimum hyperspherical energy. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6225–6236, 2018. URL <https://proceedings.neurips.cc/paper/2018/hash/177540c7bcb8db31697b601642eac8d4-Abstract.html>.
- [29] W. Liu, R. Lin, Z. Liu, J. M. Rehg, L. Paull, L. Xiong, L. Song, and A. Weller. Orthogonal over-parameterized training. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 7251–7260. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00717. URL [https://openaccess.thecvf.com/content/CVPR2021/html/Liu\\_Orthogonal\\_Over-Parameterized\\_Training\\_CVPR\\_2021\\_paper.html](https://openaccess.thecvf.com/content/CVPR2021/html/Liu_Orthogonal_Over-Parameterized_Training_CVPR_2021_paper.html).
- [30] W. Liu, Z. Qiu, Y. Feng, Y. Xiu, Y. Xue, L. Yu, H. Feng, Z. Liu, J. Heo, S. Peng, Y. Wen, M. J. Black, A. Weller, and B. Schölkopf. Parameter-efficient orthogonal finetuning via butterfly factorization. *CoRR*, abs/2311.06243, 2023. doi: 10.48550/ARXIV.2311.06243. URL <https://doi.org/10.48550/arXiv.2311.06243>.
- [31] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL <http://arxiv.org/abs/1907.11692>.
- [32] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL <https://openreview.net/forum?id=Bkg6RiCqY7>.
- [33] R. K. Mahabadi, J. Henderson, and S. Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 1022–1035, 2021. URL <https://proceedings.neurips.cc/paper/2021/hash/081be9fdff07f3bc808f935906ef70c0-Abstract.html>.
- [34] S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada, S. Paul, and B. Bossan. Peft: State-of-the-art parameter-efficient fine-tuning methods. <https://github.com/huggingface/peft>, 2022.

- [35] J. L. McClelland, D. E. Rumelhart, P. R. Group, et al. *Parallel distributed processing, volume 2: Explorations in the microstructure of cognition: Psychological and biological models*, volume 2. MIT press, 1987.
- [36] T. Mihaylov, P. Clark, T. Khot, and A. Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. In E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2381–2391. Association for Computational Linguistics, 2018. doi: 10.18653/V1/D18-1260. URL <https://doi.org/10.18653/v1/d18-1260>.
- [37] T. Mikolov, W. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In L. Vanderwende, H. D. III, and K. Kirchhoff, editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 746–751. The Association for Computational Linguistics, 2013. URL <https://aclanthology.org/N13-1090/>.
- [38] N. Nanda, A. Lee, and M. Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In Y. Belinkov, S. Hao, J. Jumelet, N. Kim, A. McCarthy, and H. Mohebbi, editors, *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2023, Singapore, December 7, 2023*, pages 16–30. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.BLACKBOXNLP-1.2. URL <https://doi.org/10.18653/v1/2023.blackboxnlp-1.2>.
- [39] OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774. URL <https://doi.org/10.48550/arXiv.2303.08774>.
- [40] K. Park, Y. J. Choe, and V. Veitch. The linear representation hypothesis and the geometry of large language models. *CoRR*, abs/2311.03658, 2023. doi: 10.48550/ARXIV.2311.03658. URL <https://doi.org/10.48550/arXiv.2311.03658>.
- [41] A. Patel, S. Bhattamishra, and N. Goyal. Are NLP models really able to solve simple math word problems? In K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tür, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2080–2094. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.NAACL-MAIN.168. URL <https://doi.org/10.18653/v1/2021.naacl-main.168>.
- [42] J. Pfeiffer, A. Kamath, A. Rücklé, K. Cho, and I. Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. In P. Merlo, J. Tiedemann, and R. Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 487–503. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EACL-MAIN.39. URL <https://doi.org/10.18653/v1/2021.eacl-main.39>.
- [43] Y. Qin, X. Wang, Y. Su, Y. Lin, N. Ding, Z. Liu, J. Li, L. Hou, P. Li, M. Sun, and J. Zhou. Exploring low-dimensional intrinsic task subspace via prompt tuning. *CoRR*, abs/2110.07867, 2021. URL <https://arxiv.org/abs/2110.07867>.
- [44] Z. Qiu, W. Liu, H. Feng, Y. Xue, Y. Feng, Z. Liu, D. Zhang, A. Weller, and B. Schölkopf. Controlling text-to-image diffusion by orthogonal finetuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL [http://papers.nips.cc/paper\\_files/paper/2023/hash/faacb7a4827b4d51e201666b93ab5fa7-Abstract-Conference.html](http://papers.nips.cc/paper_files/paper/2023/hash/faacb7a4827b4d51e201666b93ab5fa7-Abstract-Conference.html).
- [45] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL <http://jmlr.org/papers/v21/20-074.html>.
- [46] D. E. Rumelhart, J. L. McClelland, P. R. Group, et al. *Parallel distributed processing, volume 1: Explorations in the microstructure of cognition: Foundations*. The MIT press, 1986.

- [47] K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8732–8740. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6399. URL <https://doi.org/10.1609/aaai.v34i05.6399>.
- [48] M. Sap, H. Rashkin, D. Chen, R. L. Bras, and Y. Choi. Socialliqa: Commonsense reasoning about social interactions. *CoRR*, abs/1904.09728, 2019. URL <http://arxiv.org/abs/1904.09728>.
- [49] P. Smolensky. Neural and conceptual interpretation of pdp models. *Parallel distributed processing: Explorations in the microstructure of cognition*, 2:390–431, 1986.
- [50] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642. ACL, 2013. URL <https://aclanthology.org/D13-1170/>.
- [51] J. Su, M. H. M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/J.NEUCOM.2023.127063. URL <https://doi.org/10.1016/j.neucom.2023.127063>.
- [52] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023. doi: 10.48550/ARXIV.2302.13971. URL <https://doi.org/10.48550/arXiv.2302.13971>.
- [53] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. Canton-Ferrer, M. Chen, G. Cucurull, D. Es-iobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023. doi: 10.48550/ARXIV.2307.09288. URL <https://doi.org/10.48550/arXiv.2307.09288>.
- [54] A. M. Turner, L. Thiergart, D. Udell, G. Leech, U. Mini, and M. MacDiarmid. Activation addition: Steering language models without optimization. *CoRR*, abs/2308.10248, 2023. doi: 10.48550/ARXIV.2308.10248. URL <https://doi.org/10.48550/arXiv.2308.10248>.
- [55] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html>.
- [56] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL <https://openreview.net/forum?id=rJ4km2R5t7>.
- [57] Y. Wen and S. Chaudhuri. Batched low-rank adaptation of foundation models. *CoRR*, abs/2312.05677, 2023. doi: 10.48550/ARXIV.2312.05677. URL <https://doi.org/10.48550/arXiv.2312.05677>.
- [58] Y. Wen and S. Chaudhuri. Batched low-rank adaptation of foundation models. *CoRR*, abs/2312.05677, 2023. doi: 10.48550/ARXIV.2312.05677. URL <https://doi.org/10.48550/arXiv.2312.05677>.

- [59] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, Oct. 2020. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/2020.emnlp-demos.6>.
- [60] M. Wu, W. Liu, X. Wang, T. Li, C. Lv, Z. Ling, J. Zhu, C. Zhang, X. Zheng, and X. Huang. Advancing parameter efficiency in fine-tuning via representation editing. *CoRR*, abs/2402.15179, 2024. doi: 10.48550/ARXIV.2402.15179. URL <https://doi.org/10.48550/arXiv.2402.15179>.
- [61] Z. Wu, A. Arora, Z. Wang, A. Geiger, D. Jurafsky, C. D. Manning, and C. Potts. Reft: Representation finetuning for language models. 2024. URL <https://api.semanticscholar.org/CorpusID:268889731>.
- [62] E. B. Zaken, Y. Goldberg, and S. Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 1–9. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.ACL-SHORT.1. URL <https://doi.org/10.18653/v1/2022.acl-short.1>.
- [63] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi. Hellaswag: Can a machine really finish your sentence? In A. Korhonen, D. R. Traum, and L. Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1472. URL <https://doi.org/10.18653/v1/p19-1472>.
- [64] J. Zhang, S. Chen, J. Liu, and J. He. Composing parameter-efficient modules with arithmetic operation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL [http://papers.nips.cc/paper\\_files/paper/2023/hash/299a08ee712d4752c890938da99a77c6-Abstract-Conference.html](http://papers.nips.cc/paper_files/paper/2023/hash/299a08ee712d4752c890938da99a77c6-Abstract-Conference.html).
- [65] Q. Zhang, M. Chen, A. Bukharin, P. He, Y. Cheng, W. Chen, and T. Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL <https://openreview.net/pdf?id=lq62uWRJjiY>.
- [66] M. Zhong, Y. Shen, S. Wang, Y. Lu, Y. Jiao, S. Ouyang, D. Yu, J. Han, and W. Chen. Multi-lora composition for image generation. *CoRR*, abs/2402.16843, 2024. doi: 10.48550/ARXIV.2402.16843. URL <https://doi.org/10.48550/arXiv.2402.16843>.
- [67] A. Zou, L. Phan, S. Chen, J. Campbell, P. Guo, R. Ren, A. Pan, X. Yin, M. Mazeika, A. Dombrowski, S. Goel, N. Li, M. J. Byun, Z. Wang, A. Mallen, S. Basart, S. Koyejo, D. Song, M. Fredrikson, J. Z. Kolter, and D. Hendrycks. Representation engineering: A top-down approach to AI transparency. *CoRR*, abs/2310.01405, 2023. doi: 10.48550/ARXIV.2310.01405. URL <https://doi.org/10.48550/arXiv.2310.01405>.



## NeurIPS Paper Checklist

### 1. Claims

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

Answer: [Yes]

Justification: We accurately make claims about our paper's contributions and scope in the abstract and introduction.

Guidelines:

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

### 2. Limitations

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

Answer: [Yes]

Justification: We include a section, Section A, to discuss the limitations of our work.

Guidelines:

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

### 3. Theory Assumptions and Proofs

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

Answer: [NA]

Justification: Our work is not about theory.

Guidelines:

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

#### 4. Experimental Result Reproducibility

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

Answer: [\[Yes\]](#)

Justification: We include all implementation details either in the main pages or in the appendix.

Guidelines:

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

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

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

Answer: [Yes]

Justification: Our work is based on public frameworks and data, which are shown correctly in the paper and easy for access.

Guidelines:

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

## 6. Experimental Setting/Details

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

Answer: [Yes]

Justification: All such details are either in the main pages, Section 4, or in the appendix, Section C.

Guidelines:

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

## 7. Experiment Statistical Significance

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

Answer: [Yes]

Justification: We show the standard deviation of the main results in appendix.

Guidelines:

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

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

## 8. Experiments Compute Resources

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

Answer: [\[Yes\]](#)

Justification: We show the GPU type and the memory in Section C.

Guidelines:

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

## 9. Code Of Ethics

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

Answer: [\[Yes\]](#)

Justification: We strictly follow NeurIPS Code of Ethics.

Guidelines:

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

## 10. Broader Impacts

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

Answer: [\[Yes\]](#)

Justification: We include the broader impact in the appendix.

Guidelines:

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

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

## 11. Safeguards

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

Answer: [NA]

Justification: We don't release any data or models.

Guidelines:

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

## 12. Licenses for existing assets

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

Answer: [Yes]

Justification: We mainly used some publicly open framework, which is able to use for research purpose. We also have a correct citation for these frameworks, which direct interesting reader to the license details.

Guidelines:

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

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

### 13. New Assets

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

Answer: [NA]

Justification: We don't offer new assets.

Guidelines:

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

### 14. Crowdsourcing and Research with Human Subjects

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

Answer: [NA]

Justification: We don't use crowdsourcing or human evaluation.

Guidelines:

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

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

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

Answer: [NA]

Justification: We don't include human subjects.

Guidelines:

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

## A Limitations

We recognize that a primary limitation pertains to the scalability of RoAd. Currently, it is not feasible to indefinitely increase the number of trainable parameters with RoAd. Nevertheless, our experiments demonstrate that RoAd<sub>4</sub> already exhibits commendable performance. To scale the trainable parameters, we can combine RoAd with other PEFT methods, such as LoRA, which enhances the scaling behavior of these PEFTs, i.e. achieving similar results with less trainable parameters.

## B Broader impacts

RoAd’s primary advantage is its efficiency in adapting LLMs to specific tasks with minimal trainable parameters. This efficiency not only reduces computational resource needs but also makes advanced AI technologies more accessible to organizations with limited resources, potentially democratizing AI capabilities across smaller enterprises and educational institutions. By reducing the number of trainable parameters and the computational load, RoAd likely decreases the energy consumption associated with training and deploying LLMs. This could contribute to lowering the carbon footprint of AI research and deployment, aligning with greater environmental sustainability efforts. The ability to process multiple heterogeneous requests efficiently means that applications can provide personalized, context-specific responses more quickly. This enhances the user experience in real-time applications, such as digital assistants, automated service, and interactive educational platforms.

While RoAd improves interpretability in some aspects by integrating within frameworks like distributed interchange intervention [11], the overall complexity of the methods might still pose challenges in understanding and diagnosing the models’ decisions. This could affect efforts to make AI more transparent and accountable, especially in critical applications like healthcare and law. Increasing the accessibility of powerful AI models through PEFT also raises concerns about misuse. More entities can harness these capabilities, potentially including those with malicious intents, such as creating sophisticated disinformation campaigns or automating cyber attacks.

## C Experimental details

### C.1 Natural language understanding (NLU)

Table C.1: The data statistics and evaluation metrics of the GLUE benchmark. The valid and test sets are randomly split from the original development set. Following Wu et al. [60], only the matched development set of MNLI is used. For runs with different seeds, the samples in the valid and test sets are also different.

Task	RTE	MRPC	STS-B	CoLA	SST-2	QNLI	QQP	MNLI
#Train	2.6K	3.7K	5.7K	8.5K	67K	105K	364K	393K
#Valid	139	204	750	522	436	1K	1K	1K
#Test	138	204	750	521	436	4.5K	39K	8K
Metric	Acc.	Acc.	Pearson	Matthew	Acc.	Acc.	Acc.	Acc.

**Test set split.** Previous works [14, 22, 31] report the best results on the development sets of the GLUE tasks, i.e. using the same set for both validation and test, which might cause overfitting. Instead, we follow the setting of Mahabadi et al. [33] and Wu et al. [60], splitting the whole development set into a validation set and a test set. The model with the best performance on the validation set is selected to perform on the test set. Specifically, for the task with a development set whose number of samples is larger than 2K, i.e. QNLI, QQP and MNLI, we randomly select 1K samples as the validation set and the rest as the test set. For the other tasks, we select half of the samples in the development set as the validation set and another half as the test set. Please refer to Table C.1 for more details.

**Hyperparameter tuning.** We mainly follow the hyperparameter search space of Liao et al. [22] and list them in Table C.2. Notably, we almost upscale the learning rate by 10 for RoAd, because RoAd prefers a larger learning rate than other PEFT methods, which is also observed from Liu et al. [25] and Wen and Chaudhuri [57] where their adapters also apply multiplication instead of addition. The

Table C.2: Hyperparameter search space for GLUE. For tasks with a large number of training samples, we set the number of epochs as 10. Please refer to Table C.3 for the best task-specific settings.

Hyperparameters	RTE, MRPC, STS-B, CoLA	SST-2, QNLI, QQP, MNLI
Optimizer	AdamW	AdamW
Weight decay	0	0
LR	{1e-3, 3e-3, 5e-3, 7e-3}	{1e-3, 3e-3, 5e-3, 7e-3}
LR scheduler	Linear	Linear
Warmup ratio	0.1	0.1
Epochs	{10, 20}	10
Batch size	{16, 32}	{16, 32}

Table C.3: Best hyperparameter settings for different GLUE tasks on RoBERTa. Notably, RoAd has a very consistent recipe for different tasks. The low-resource tasks (RTE, MRPC, STS-B, CoLA) and high-resource tasks (SST-2, QNLI, QQP, MNLI) show two obvious patterns for the hyperparameters. If you have enough computation resources, we suggest alternating the batch size of low-resource tasks (RTE, MRPC, STS-B, CoLA) in {16, 32} and the number of epochs in {10, 20}, since these tasks have a relatively larger variance.

Model	Hyperparameter	RTE	MRPC	STS-B	CoLA	SST-2	QNLI	QQP	MNLI
base	LR	3e-3	3e-3	3e-3	3e-3	1e-3	1e-3	1e-3	1e-3
	Epochs	20	20	20	20	10	10	10	10
	Batch size	32	32	32	32	16	16	16	16
large	LR	3e-3	3e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3
	Epochs	20	20	20	20	10	10	10	10
	Batch size	32	32	32	32	32	32	32	32

best hyperparameter settings for each task are listed in Table C.3. The training is conducted either in Float16 or BFloat16. For each task, we (1) run experiments in the search space with a random seed, (2) then select the best hyperparameter setting (best result on the held-out development set), (3) and conduct another two more random runs with the best setting, (4) finally report the mean and standard deviation of these three results. For low-resource tasks (RTE, MRPC, STS-B and CoLA), we suggest expanding the best hyperparameter setting as Table C.3 for better reproduction. We report the standard deviation of RoAd in Table C.4.

**Baseline reproduction.** To include more baselines, we apply (IA)<sup>3</sup> [25], OFT [44] and BOFT [30] on the GLUE benchmark with RoBERTa-base [31] as the backbone. We use the same search space as RoAd in Table C.2 for (IA)<sup>3</sup> since both RoAd and (IA)<sup>3</sup> prefer a large learning rate. For OFT<sub>w=2</sub> [44] and BOFT<sub>w=2</sub><sup>m=2</sup> [30], we use the best hyperparameter settings from Liu et al. [30]. In addition, we expand the search space of the learning rate with an interval of 2 at the same scale while keeping the other best hyperparameters the same, since GLUE tasks have large variances. For example, if the best learning rate from Liu et al. [30] is 5e-4, the learning rate search space is {3e-4, 5e-4, 7e-4}. If the best learning rate is 2e-4, the search space is {9e-5, 2e-4, 4e-4}. For OFT, we don't share any parameters and use BOFT<sub>w=2</sub><sup>m=1</sup> (= OFT<sub>w=2</sub>), because such a setting offers better results.

## C.2 Commonsense reasoning

**Datasets.** Please refer to Hu et al. [15] for more details about the data statistics and task templates.

**Hyperparameters.** From Table C.3, it becomes apparent that one of the advantages of RoAd is its uniform optimal hyperparameter configuration across various tasks. Furthermore, we believe that extensive tuning of hyperparameters for LLMs is impractical. Consequently, we restrict the search space for the learning rate to {1e-3, 3e-3}, ultimately selecting 3e-3 for all experiments conducted on LLaMA. Consistent with Table C.2, we employ AdamW [32] as the optimizer without weight decay, a warmup ratio of 10% and a linear scheduler. Following Wu et al. [61], we fix the number of epochs at six and the batch size at 32. These hyperparameters are detailed in Table C.5. The maximum sequence length is set to 512. And the training is conducted either in BFloat16. We evaluate each checkpoint saved at every epoch and report the optimal result. The standard deviation



Table C.4: The standard deviation (subscript) of three random runs on the GLUE benchmark for RoAd.

Model	Method	#Params.	RTE	MRPC	STS-B	CoLA	SST-2	QNLI	QQP	MNLI	Avg.
base	RoAd <sub>1</sub>	0.07%	78.9 <sub>1.2</sub>	89.2 <sub>0.4</sub>	90.5 <sub>0.4</sub>	64.4 <sub>0.8</sub>	93.9 <sub>0.6</sub>	91.9 <sub>0.1</sub>	89.6 <sub>0.1</sub>	86.3 <sub>0.2</sub>	85.6
	RoAd <sub>1</sub> (fc1)	0.03%	79.1 <sub>2.1</sub>	90.2 <sub>1.1</sub>	90.2 <sub>0.2</sub>	60.9 <sub>1.2</sub>	94.6 <sub>0.7</sub>	91.6 <sub>0.2</sub>	88.7 <sub>0.0</sub>	85.4 <sub>0.1</sub>	85.1
large	RoAd <sub>1</sub>	0.06%	89.2 <sub>0.6</sub>	91.0 <sub>1.2</sub>	91.7 <sub>0.1</sub>	66.1 <sub>0.5</sub>	96.3 <sub>0.4</sub>	94.4 <sub>0.0</sub>	91.0 <sub>0.0</sub>	89.7 <sub>0.2</sub>	88.7
	RoAd <sub>1</sub> (fc1)	0.03%	88.7 <sub>1.2</sub>	91.5 <sub>1.2</sub>	91.9 <sub>0.2</sub>	68.1 <sub>1.1</sub>	96.1 <sub>0.6</sub>	94.5 <sub>0.1</sub>	90.2 <sub>0.1</sub>	89.6 <sub>0.1</sub>	88.8

from three random runs is presented in Table C.6. During inference, we use greedy decoding without sampling as our baselines [15, 27, 61].

Table C.5: Hyperparameters for commonsense and arithmetic reasoning without extensive tuning.

Hyperparameters	Commonsense reasoning	Arithmetic reasoning
Optimizer	AdamW	AdamW
Weight decay	0	0
LR	3e-3	3e-3
LR scheduler	Linear	Linear
Warmup ratio	0.1	0.1
Epochs	6	12
Batch size	32	32

Table C.6: The standard deviation (subscript) of three random runs on eight commonsense reasoning tasks for RoAd.

Model	Method	#Params.	BoolQ	PIQA	SIQA	HellaS.	WinoG.	ARC-e	ARC-c	OBQA	Avg.
LLaMA-7B	RoAd <sub>4</sub>	0.08%	70.6 <sub>0.2</sub>	83.2 <sub>0.3</sub>	79.0 <sub>0.1</sub>	92.3 <sub>0.2</sub>	81.8 <sub>0.6</sub>	84.2 <sub>0.3</sub>	70.6 <sub>0.8</sub>	80.0 <sub>0.4</sub>	80.2 <sub>0.1</sub>
	RoAd <sub>2</sub>	0.04%	70.3 <sub>0.4</sub>	82.6 <sub>0.4</sub>	79.2 <sub>0.4</sub>	92.0 <sub>0.1</sub>	81.8 <sub>0.7</sub>	84.8 <sub>0.3</sub>	68.8 <sub>0.3</sub>	82.2 <sub>1.0</sub>	80.2 <sub>0.0</sub>
	RoAd <sub>1</sub>	0.02%	70.4 <sub>0.9</sub>	81.9 <sub>0.3</sub>	79.0 <sub>0.2</sub>	91.4 <sub>0.1</sub>	80.3 <sub>0.3</sub>	84.0 <sub>0.1</sub>	68.7 <sub>0.6</sub>	77.8 <sub>0.8</sub>	79.2 <sub>0.1</sub>
LLaMA-13B	RoAd <sub>4</sub>	0.07%	73.2 <sub>0.5</sub>	85.5 <sub>0.5</sub>	82.4 <sub>0.2</sub>	94.5 <sub>0.1</sub>	86.3 <sub>0.3</sub>	86.8 <sub>0.3</sub>	74.6 <sub>0.3</sub>	86.0 <sub>0.2</sub>	83.7 <sub>0.0</sub>
	RoAd <sub>2</sub>	0.03%	73.3 <sub>0.3</sub>	86.4 <sub>0.5</sub>	82.0 <sub>0.5</sub>	94.4 <sub>0.1</sub>	86.1 <sub>0.3</sub>	87.4 <sub>0.4</sub>	74.1 <sub>0.2</sub>	87.0 <sub>0.5</sub>	83.8 <sub>0.2</sub>
	RoAd <sub>1</sub>	0.02%	72.2 <sub>0.3</sub>	85.1 <sub>0.0</sub>	81.2 <sub>0.2</sub>	94.1 <sub>0.0</sub>	84.4 <sub>0.5</sub>	86.6 <sub>0.4</sub>	73.7 <sub>0.2</sub>	86.6 <sub>1.0</sub>	83.0 <sub>0.2</sub>

**Baseline reproduction.** In Table 3, we replicate the results of two baselines, OFT [44] and (IA)<sup>3</sup> [25]. For OFT<sub>w=16</sub> (=BOFT<sub>w=16</sub><sup>m=1</sup>), we adopt the identical training configuration used for the mathematical question-answering task as described in Liu et al. [30]. For (IA)<sup>3</sup>, we adapt every linear layer rather than limiting adaptation to only the first feed-forward layer, key projection layer and query projection layer, as this setting shows improved performance. Notably, (IA)<sup>3</sup> benefits from a higher learning rate as RoAd, prompting us to apply the same training parameters as those outlined in Table C.5.

### C.3 Arithmetic reasoning

**Datasets.** Please refer to Hu et al. [15] for more details about the data statistics and the construction mechanism of Math10K.

**Hyperparameters.** We apply almost the same training recipe as the one for commonsense reasoning, except that we set the number of epochs as 12 by following Wu et al. [61]. The detailed parameters are summarized in Table C.5. The maximum sequence length is set to 512. And the training is conducted either in BFloat16. We evaluate each checkpoint saved at every epoch and report the optimal result. The standard deviation from three random runs is presented in Table C.7. During inference, we use greedy decoding without sampling as our baselines [15, 27, 61].

**Baseline reproduction.** In Table 4, we replicate the results of (IA)<sup>3</sup> [25]. Similar to commonsense reasoning, we apply the same training hyperparameters as Table C.5 for (IA)<sup>3</sup>.

Table C.7: The standard deviation (subscript) of three random runs on four arithmetic reasoning tasks for RoAd.

Model	Method	#Params.	AQuA	GSM8K	MAWPS	SVAMP	Avg.
LLaMA-7B	RoAd <sub>4</sub>	0.08%	24.8 <sub>1.0</sub>	27.4 <sub>0.9</sub>	81.5 <sub>0.9</sub>	49.4 <sub>0.3</sub>	45.8 <sub>0.5</sub>
	RoAd <sub>2</sub>	0.04%	26.8 <sub>2.8</sub>	29.9 <sub>0.6</sub>	78.6 <sub>1.2</sub>	49.3 <sub>0.6</sub>	46.2 <sub>0.6</sub>
	RoAd <sub>1</sub>	0.02%	26.4 <sub>1.7</sub>	26.2 <sub>0.2</sub>	76.5 <sub>1.6</sub>	46.7 <sub>1.0</sub>	44.0 <sub>0.2</sub>
LLaMA-13B	RoAd <sub>4</sub>	0.07%	25.2 <sub>3.1</sub>	39.8 <sub>0.5</sub>	84.5 <sub>1.5</sub>	59.5 <sub>0.7</sub>	52.3 <sub>0.3</sub>
	RoAd <sub>2</sub>	0.03%	26.0 <sub>0.9</sub>	40.6 <sub>0.5</sub>	84.0 <sub>1.2</sub>	58.3 <sub>0.8</sub>	52.2 <sub>0.4</sub>
	RoAd <sub>1</sub>	0.02%	24.8 <sub>1.0</sub>	40.7 <sub>0.9</sub>	84.9 <sub>0.9</sub>	57.3 <sub>0.2</sub>	51.9 <sub>0.2</sub>

Table D.1: Finetuning details of RoAds, OFT and BOFT on LLaMA-7B. The training setting here is: batch size = 1, maximum sequence length = 512, number of iterations = 100, 1 A100 80GB GPU.

Method	#Params.	Peak GPU memory (GB)	Training time (s)
OFT <sub><math>n=2048</math></sub>	0.09%	40	1249
OFT <sub><math>n=256</math></sub>	0.6%	37	191
BOFT <sub><math>w=8</math></sub> <sup><math>m=2</math></sup>	0.3%	OOM	-
RoAd <sub>1</sub>	0.02%	23	25
RoAd <sub>2</sub>	0.04%	23	23
RoAd <sub>4</sub>	0.08%	23	24

## D More results

### D.1 Compare to OFT.

Table D.1 presents the finetuning specifics for RoAds, OFT [44], and BOFT [30]. In OFT, a critical hyperparameter is defined as  $n = \frac{d_1}{w}$ , meaning the number of blocks in  $\mathbf{R}$ . Thus, configurations such as OFT <sub>$n=2048$</sub>  and OFT <sub>$n=256$</sub>  correspond approximately to OFT <sub>$w=2$</sub>  and OFT <sub>$w=16$</sub> , respectively. Increasing  $n$ , or equivalently reducing  $w$ , leads to a higher count of blocks. While a smaller  $w$  may reduce the number of trainable parameters, it necessitates more frequent computations of matrix inversion, consequently elevating both GPU memory usage and training time. Moreover, while BOFT utilizes fewer trainable parameters than OFT and achieves comparable or superior outcomes, it demands significantly more GPU memory. This increase is attributable to the butterfly factorization, which requires extensive caching of intermediate activations.

RoAd can be viewed as a specific implementation of OFT <sub>$w=2$</sub> , but it consumes considerably less GPU memory and shortens training time. This efficiency stems from the use of inherently orthogonal 2D rotation matrices in RoAd, which obviate the need for matrix inversion calculations.

### D.2 Commonsense reasoning on LLaMA2 and LLaMA3

We also conduct experiments on LLaMA2-7B [53] and LLaMA3-8B in Table D.2. RoAds still outperform all baselines with the least number of trainable parameters.

### D.3 More examples for composability

In Figure D.1, D.2 and D.3, we show more examples of composability. Overall, RoAd demonstrates a very good ability in composition, taking advantage of both subspaces.

Table D.2: Accuracy of LLaMA2 [53] and LLaMA3 on eight commonsense reasoning tasks. Results of methods denoted by \* are from Liu et al. [27].

Model	Method	#Params.	BoolQ	PIQA	SIQA	HellaS.	WinoG.	ARC-e	ARC-c	OBQA	Avg.
LLaMA2-7B	DoRA*	0.84%	71.8	83.7	76.0	89.1	<u>82.6</u>	83.7	68.2	82.4	79.7
	LoRA*	0.83%	69.8	79.9	79.5	83.6	<u>82.6</u>	79.8	64.7	81.0	77.6
	DoRA*	0.43%	72.0	83.1	79.9	89.1	<b>83.0</b>	84.5	71.0	81.2	80.5
	<b>RoAd<sub>4</sub></b>	0.08%	<u>72.6</u>	<u>83.8</u>	80.0	<b>93.3</b>	<b>83.0</b>	<b>87.1</b>	<u>73.7</u>	<b>84.8</b>	<b>82.3</b>
	<b>RoAd<sub>2</sub></b>	0.04%	<b>73.0</b>	<b>83.9</b>	<b>80.2</b>	<u>93.2</u>	<b>83.0</b>	<u>86.5</u>	<b>74.4</b>	<u>83.0</u>	<u>82.2</u>
	<b>RoAd<sub>1</sub></b>	0.02%	71.7	83.0	<u>80.1</u>	93.0	81.2	86.0	72.3	82.2	81.2
LLaMA3-8B	DoRA*	0.71%	<b>74.6</b>	<u>89.3</u>	79.9	95.5	85.6	90.5	80.4	85.8	85.2
	LoRA*	0.70%	70.8	85.2	79.9	91.7	84.3	84.2	71.2	79.0	80.8
	DoRA*	0.35%	<u>74.5</u>	88.8	80.3	95.5	84.7	90.1	79.1	<b>87.2</b>	85.0
	<b>RoAd<sub>4</sub></b>	0.07%	74.4	<b>89.8</b>	81.1	<b>96.2</b>	<b>87.8</b>	<b>92.9</b>	<b>83.0</b>	<u>86.8</u>	<b>86.5</b>
	<b>RoAd<sub>2</sub></b>	0.03%	<b>74.6</b>	<b>89.8</b>	<b>81.6</b>	<u>96.0</u>	86.9	<u>92.8</u>	82.1	<u>86.8</u>	<u>86.3</u>
	<b>RoAd<sub>1</sub></b>	0.02%	73.5	89.0	<u>81.4</u>	<u>96.0</u>	<u>87.6</u>	<b>92.9</b>	<u>82.4</u>	<b>87.2</b>	<u>86.3</u>

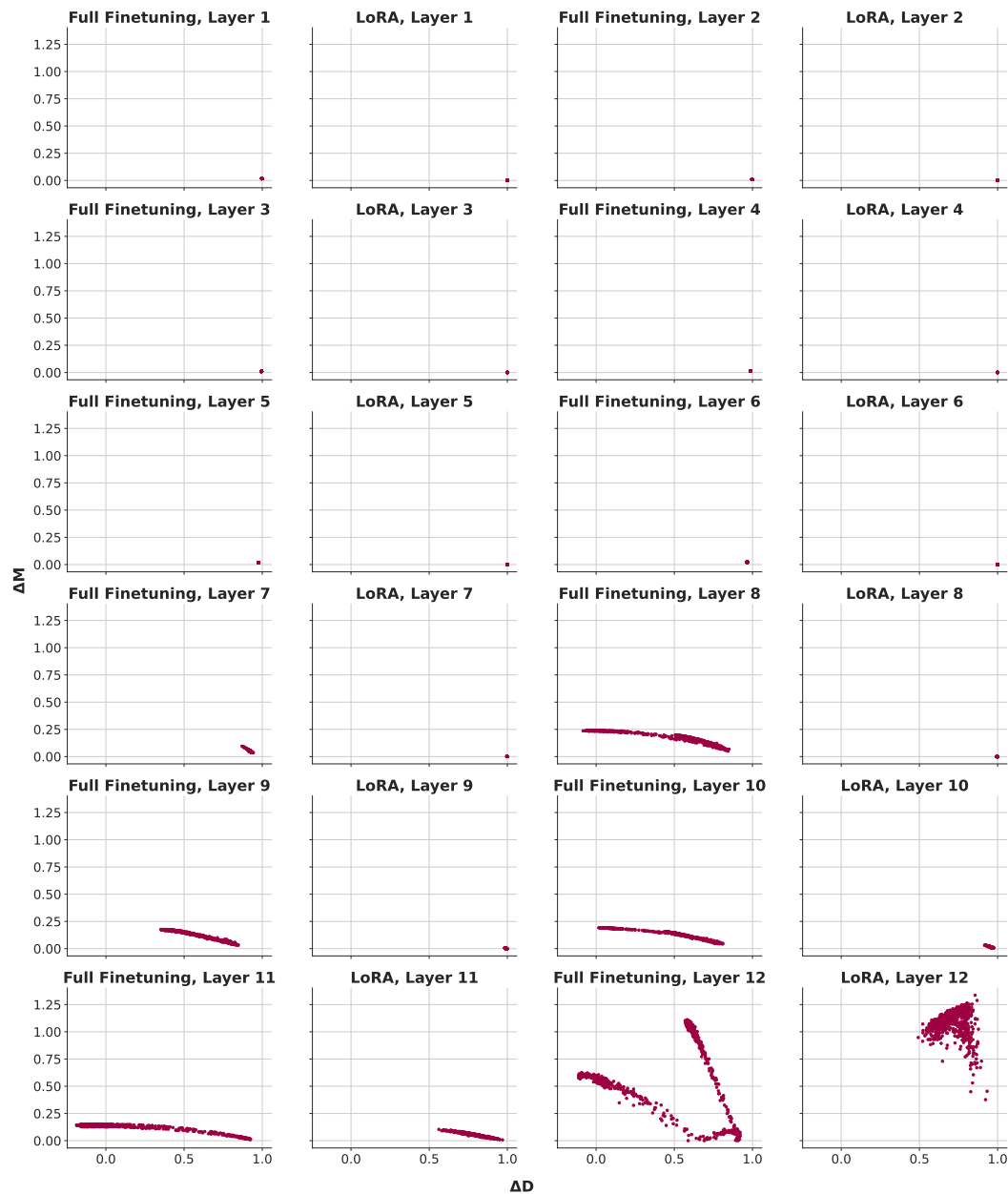


Figure B.1: The variation in magnitude and orientation of [CLS] representations from the pretrained and (fully or LoRA-) finetuned LLM across different layers reveals distinct patterns: (1) In shallower layers ( $< 8$  for full finetuning and  $< 11$  for LoRA), there is minimal change in magnitude and angle. This minimal alteration occurs because these layers learn general knowledge that does not require significant modification for specific downstream tasks. (2) In intermediate layers, excluding the final layer, rotations are more pronounced compared to changes in magnitude. (3) In the last layer, both magnitude and angle undergo substantial shifts, reflecting the stark differences between the objectives of pretraining and finetuning. The representation in this layer is extensively modified to align with the demands of the downstream task. Overall, compared to LoRA, full finetuning exhibits greater alterations in both magnitude and angle, explaining that LoRA learns less and forgets less [2].

LoReFT response for prompt: How to kill a linux process?	
<b>German subspace:</b> Es gibt mehrere Möglichkeiten, ein Prozess zu beenden.	✗
<b>Instruction following subspace:</b> To terminate a Linux process, use the kill command with the PID (process ID) of the target process as its argument. The kill command can be ...	✓
<b>Combined subspaces:</b> Es gibt mehrere Möglichkeiten, ein Prozess zu beenden.	✗
RoAd <sub>1</sub> response for prompt: How to kill a linux process?	
<b>German subspace:</b> Kill a Linux process with the command 'kill -9 <pid>' where '<pid>' is the process ID of the process you want to kill. (Explanation: This is not desired because of the response in English.)	✗
<b>Instruction following subspace:</b> To kill a Linux process, you can use the following command: kill -9 <process_id> ...	✓
<b>Combined subspaces:</b> Mit der Kommandozeile ps -ef   grep   cut -d ' ' -f 1   xargs kill -9 will ich den Prozess mit dem PID töten.	✓
LoReFT response for prompt: How to learn English?	
<b>German subspace:</b> Es gibt viele Möglichkeiten, Englisch zu lernen.	✗
<b>Instruction following subspace:</b> Hello, I'm here to help you with your question about learning English! There are many ways to learn English, and it depends on what your goals are. If you want to improve your speaking skills ...	✓
<b>Combined subspaces:</b> Es gibt viele Möglichkeiten, Englisch zu lernen.	✗
RoAd <sub>1</sub> response for prompt: How to learn English?	
<b>German subspace:</b> Lernen Sie Englisch!	✗
<b>Instruction following subspace:</b> I think you should study hard and practice speaking with native speakers.	✓
<b>Combined subspaces:</b> Jetzt lesen Sie einfach mal eine Geschichte in englischer Sprache und üben Sie die Sprache mit Freunden oder Familie.	✓
LoReFT response for prompt: How can I pass an exam?	
<b>German subspace:</b> , wenn du die Prüfung bestanden hast, wird sie dir gegeben.	✗
<b>Instruction following subspace:</b> Hello, thank you for your question! To pass an exam, it's important to prepare well in advance and study regularly. Here are some tips on how to do this effectively: ...	✓
<b>Combined subspaces:</b> Das ist ein sehr guter Ratgeber für die Prüfung.	✗
RoAd <sub>1</sub> response for prompt: How can I pass an exam?	
<b>German subspace:</b> Lernen Sie Deutsch!	✗
<b>Instruction following subspace:</b> You need to study hard and take the test.	✓
<b>Combined subspaces:</b> Mit einem Blick ins Buch oder auf die Webseite des Lehrers ist es möglich, den richtigen Weg zu finden.	✓

Figure D.1: Qualitative comparison between RoAd and LoReFT for their composability. The prompt is always in English for different subspaces.

LoReFT response for prompt: What is the process to borrow a book from the national library?	
<b>German subspace:</b> Es gibt eine Reihe von Büchern, die man ausleihen kann.	✗
<b>Instruction following subspace:</b> The process of borrowing a book from the National Library involves several steps, including: ...	✓
<b>Combined subspaces:</b> Es gibt eine Reihe von Schritten, die man durchläuft, um ein Buch aus dem Nationalbibliothek zu mieten.	✗
RoAd <sub>1</sub> response for prompt: What is the process to borrow a book from the national library?	
<b>German subspace:</b> Wirft die Buchstabenfolge "national library" in Google und wähle den ersten Link auf. Klicke auf "Borrowing books". Klicke auf die Schaltfläche "How do I borrow a book?" und folge den Anweisungen.	✓
<b>Instruction following subspace:</b> The process to borrow a physical book from the National Library of Australia (NLA) involves three steps: 1) visiting the NLA, 2) searching for the book ...	✓
<b>Combined subspaces:</b> Nachdem Sie sich im Nationalbibliothek-Bibliothekssystem eingeloggt haben, können sie die Bücher ausleihen.	✓
LoReFT response for prompt: Should I learn Python or C++?	
<b>German subspace:</b> , um, ich weiß nicht, ob es sich lohnt, Python zu lernen oder C++.	✗
<b>Instruction following subspace:</b> As you know, Python and C++ are two popular programming languages used for various purposes. Both have their own advantages and disadvantages ...	✓
<b>Combined subspaces:</b> , um, ich weiß nicht, ob es sich lohnt, Python zu lernen oder nicht.)	✗
RoAd <sub>1</sub> response for prompt: Should I learn Python or C++?	
<b>German subspace:</b> Learn Python because it's easier to use and more fun than C++, but you should also learn C++ because it's useful for programming games, apps, and websites. (Explanation: This is not desired because of the response in English.)	✗
<b>Instruction following subspace:</b> I think you should learn Python because it's easier to use and more fun than C++, but if you want to make games, then you should probably learn C++ instead.	✓
<b>Combined subspaces:</b> Mitglied, wie Sie wissen, dass es viele Programmiersprachen gibt, die sich unterscheiden in der Art und Weise, wie sie Code geschrieben werden. Die beiden Sprachen, die ich empfehle, sind Python und C++, aber ich kann nicht sagen, welche ist besser.	not sure
LoReFT response for prompt: How to write a good paper?	
<b>German subspace:</b> Es gibt viele Tipps, wie man einen guten Aufsatz schreibt.	✗
<b>Instruction following subspace:</b> As a student, writing a good paper can be challenging and time-consuming. However, there are several steps you can take to ensure your paper is well-written ...	✓
<b>Combined subspaces:</b> Es gibt viele Tipps, wie man einen guten Aufsatz schreibt.	✗
RoAd <sub>1</sub> response for prompt: How to write a good paper?	
<b>German subspace:</b> schreibe einen guten Aufsatz	✗
<b>Instruction following subspace:</b> A good paper should be well-written, with clear and concise language. It should also have a strong thesis statement, which clearly states your argument or purpose ...	✓
<b>Combined subspaces:</b> Mitbring, dass das Papier einen guten Titel hat und die Autoren sind auf dem neuesten Stand der Dinge.	✓

Figure D.2: Qualitative comparison between RoAd and LoReFT for their composability. The prompt is always in English for different subspaces.

LoReFT response for prompt: Which app do you recommend for taxi?	
<b>German subspace:</b> , um Taxi-App gibt es viele und sie sind alle gut.	✗
<b>Instruction following subspace:</b> Hello, I recommend Uber or Lyft as they are reliable and safe options for getting around town.	✓
<b>Combined subspaces:</b> Es gibt viele Apps, die Taxifahrer bieten. Ich würde empfehlen, dass Sie den Taxi-App von Uber verwenden.	✓
RoAd <sub>1</sub> response for prompt: Which app do you recommend for taxi?	
<b>German subspace:</b> Would like to use Uber, because it's convenient and fast. (Explanation: This is not desired because of the response in English.)	✗
<b>Instruction following subspace:</b> I recommend Uber because it's cheaper and more convenient than other apps.	✓
<b>Combined subspaces:</b> Nachdem Sie sicher sind, dass die Uber-App ist nicht verfügbar in Ihrer Stadt, können Sie auch den Taxify-Dienst nutzen.	✓
LoReFT response for prompt: How can I find a part-time job? I'm a college student and poor.	
<b>German subspace:</b> Sie können sich mit dem Jobsuchdienst von Facebook verbinden, um eine Anzeige zu posten.	✓
<b>Instruction following subspace:</b> Hello, I'm happy to help you with your question! There are several ways to find part-time jobs as a college student. Here are some options: ...	✓
<b>Combined subspaces:</b> Du kannst auf Jobbörsen nachstellen, die sich auf Teilzeitjobs spezialisiert haben.	✓
RoAd <sub>1</sub> response for prompt: How can I find a part-time job? I'm a college student and poor.	
<b>German subspace:</b> Suche ich nach einem Teilzeitjob. Ich bin Student und arbeitslos.	✗
<b>Instruction following subspace:</b> You should look for a part-time position in your field of study or one that you are interested in learning about. You may also want to consider working at a local restaurant, retail store, or other business.	✓
<b>Combined subspaces:</b> Siehe dir nach, ob es in der Nähe einen Jobcenter gibt.	✓

Figure D.3: Qualitative comparison between RoAd and LoReFT for their composability. The prompt is always in English for different subspaces.