L2-LoRA: Improving Low-Rank Adaptation with Layer-Specific Regularization

Xiang Zhang, Rui Xie*, Shikun Zhang*

Peking University
xiangzhang@stu.pku.edu.cn, {ruixie, zhangsk}@pku.edu.cn

Abstract

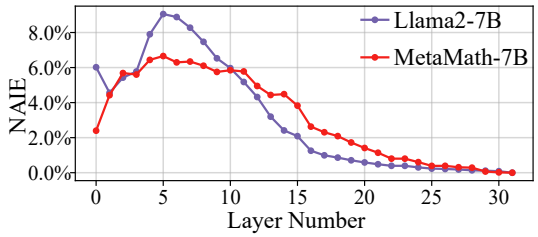
Fine-tuning large language models (LLMs) in a parameter-efficient manner while preserving their pre-trained world knowledge remains a significant challenge. While Low-Rank Adaptation (LoRA) and its variants effectively mitigate catastrophic forgetting, they do not fully eliminate the loss of critical pre-trained knowledge. In this work, we first analyze the layer-wise distribution of domain-specific knowledge within LLMs through knowledge localization, and empirically identify a clear layer-specific pattern: pre-trained world knowledge predominantly resides in lower layers, whereas knowledge relevant to downstream tasks is more concentrated in higher layers. Motivated by this observation, we propose L2-LoRA, a simple yet effective variant of LoRA that applies layer-specific L2 regularization to the LoRA weights during fine-tuning. Specifically, L2-LoRA imposes stronger regularization on lower layers to preserve pre-trained world knowledge, while allowing greater adaptation in higher layers to better align with downstream tasks. Experiments across multiple benchmarks show that L2-LoRA not only consistently outperforms vanilla LoRA in downstream performance, but also effectively mitigates catastrophic forgetting by retaining more pre-trained knowledge.

Introduction

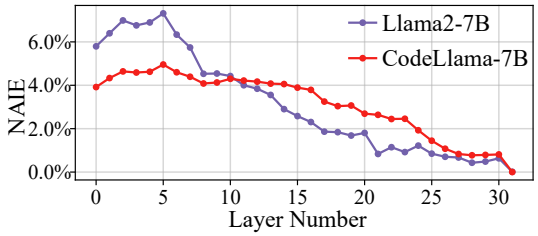
Large language models (LLMs) (Brown et al. 2020; Touvron et al. 2023; Jiang et al. 2023; Young et al. 2024; AI@Meta 2024; Guo et al. 2025; xAI 2025) have revolutionized natural language processing (NLP), achieving state-of-the-art performance across a wide range of tasks including question answering, code generation, and mathematical reasoning. Despite their strong capabilities, fine-tuning LLMs on new datasets or domain-specific tasks remains crucial in many practical scenarios. Parameter-efficient fine-tuning (PEFT) (Ding et al. 2023; Han et al. 2024) has emerged as an effective and resource-friendly alternative to full fine-tuning, enabling competitive performance with only a small number of trainable parameters (Xu et al. 2023).

Among PEFT methods, Low-Rank Adaptation (LoRA) (Hu et al. 2021) is one of the most widely adopted due to its strong empirical performance and ease of integration. LoRA freezes the pre-trained model weights $W_0 \in \mathbb{R}^{d_1 \times d_2}$ and introduces two low-rank trainable matrices, $B \in \mathbb{R}^{d_1 \times r}$ and $A \in \mathbb{R}^{r \times d_2}$, to approximate weight updates during fine-tuning. While LoRA improves downstream task performance, it often leads to degraded generalization, a phenomenon known as catastrophic forgetting (Biderman et al. 2024; Ren et al. 2024; Yang et al. 2024b; Bafghi et al. 2025).

To find the optimal trade-off between downstream adaptation and the preservation of pre-trained world knowledge, previous works have explored the dynamics of low-rank matrices (i.e., A and B in LoRA) from three perspectives: the singular value decomposition (SVD) of pre-trained weights (Yang et al. 2024b; Liang and Li 2024; Meng, Wang, and Zhang 2024), sparse updates (Zhang et al. 2024; Wang et al. 2024a), and interpolation techniques (Ren et al. 2024). How-



(a) Math knowledge localization



(b) Code knowledge localization

Figure 1: Knowledge localization analysis of Llama2-7B (base model) and its fine-tuned models (MetaMath-7B (Yu et al. 2023), CodeLlama-7B (Roziere et al. 2023)) on math and code datasets. Using causal tracing, we observe that pre-trained knowledge is predominantly stored in lower layers, while knowledge relevant to downstream tasks shifts toward higher layers after fine-tuning.

*Corresponding author.

ever, most existing LoRA-based methods apply parameter updates uniformly across all layers, without considering the functional hierarchy of LLMs, where lower layers specialize in syntactic features and higher layers encode semantic abstractions (Tenney, Das, and Pavlick 2019; Lei and Cooper 2025; Rao and Bhandari 2025). Such a layer-agnostic adaptation strategy disrupts the organization of pre-trained knowledge, often overwriting general linguistic and factual information stored in lower layers.

Meanwhile, recent work on knowledge localization and editing has shown that knowledge in transformer models is not uniformly distributed across layers, but is instead concentrated in specific layers (Geva et al. 2021; Meng et al. 2022; Hase et al. 2023). Lee et al. (2023) demonstrate that model robustness to distribution shifts can be improved by fine-tuning only a subset of layers. Pu et al. (2023) further analyze LoRA and find that attention-level modifications and the use of higher layers are crucial for downstream task performance, yet they do not address the preservation of pre-trained knowledge. It remains unclear how the distribution of knowledge differs between base LLMs and their fine-tuned counterparts. Motivated by research on knowledge localization and editing, we employ Causal Tracing (Meng et al. 2022) to conduct a layer-wise analysis of knowledge storage in both base and fine-tuned LLMs. As shown in Figure 1, our analysis reveals a consistent shift in knowledge localization after fine-tuning. For the same task domain (e.g., math and code), knowledge that was previously distributed across lower layers in the base model becomes increasingly concentrated in higher layers of the fine-tuned model. This suggests that pre-trained world knowledge is predominantly localized in lower layers, while fine-tuning reallocates task-specific knowledge to higher layers. These findings suggest the importance of layer-aware adaptation strategies that preserve lower-layer representations encoding general world knowledge, while enabling higher layers to specialize in downstream tasks.

Based on our findings, we propose L2-LoRA, a layer-aware LoRA variant that integrates knowledge localization into the fine-tuning process. Unlike existing LoRA-based methods that apply uniform adaptation across all layers, L2-LoRA assigns layer-specific L_2 regularization strengths to the LoRA weights. We employ Causal Tracing on the base model using general datasets to identify layers that predominantly store pre-trained world knowledge. Guided by the layer-wise knowledge distribution results, L2-LoRA imposes stronger regularization on lower layers to preserve general capabilities, while allowing greater flexibility in higher layers to adapt to downstream tasks. We evaluate L2-LoRA across diverse tasks, including commonsense reasoning, mathematical reasoning, and instruction following. Experimental results show that L2-LoRA consistently outperforms standard LoRA and recent LoRA-based variants across all benchmarks. Moreover, L2-LoRA exhibits strong generalization ability by effectively preserving pre-trained world knowledge, while achieving better adaptation to downstream domains. In summary, our main contributions are as follows:

- We conduct a fine-grained Causal Tracing analysis to investigate how different types of knowledge are localized across transformer layers. Our findings reveal a consistent shift in layer-wise knowledge distribution after fine-tuning, with pre-trained world knowledge concentrated in lower layers and task-specific knowledge moving toward higher layers.
- We propose **L2-LoRA**, a novel layer-aware parameter-efficient fine-tuning method that integrates knowledge localization into the fine-tuning process. By assigning layer-specific L_2 regularization based on the knowledge distribution of the base model, L2-LoRA preserves general capabilities in lower layers while allowing flexible adaptation in higher layers.
- We conduct extensive experiments across three representative tasks: commonsense reasoning, mathematical reasoning, and instruction following. The results demonstrate that L2-LoRA outperforms standard LoRA and recent variants, achieving better performance while more effectively retaining pre-trained knowledge.

Related Works

Knowledge Localization

LLMs are believed to store extensive factual knowledge acquired through pre-training on massive corpora (Petroni et al. 2019; Radford et al. 2019). The knowledge localization assumption posits that factual knowledge can be attributed to a small set of internal components such as neurons, layers, or attention heads (Dai et al. 2022; Chen et al. 2024a,b). A growing body of research aims to localize model behavior to specific architectural components (Hase et al. 2023). Several studies focus on the role of MLP layers in storing factual information (Geva et al. 2022; Meng et al. 2023). Geva et al. (2021) argue that entity-specific information is stored in MLP blocks as a two-layer key-value memory structure. However, subsequent work (Jiang et al. 2024) suggests that both self-attention and MLP modules make comparably important contributions to factual recall. Chen et al. (2024b) further emphasize the role of attention heads in knowledge localization, suggesting that knowledge may be distributed across multiple components.

LoRA and Its Variants

LoRA (Hu et al. 2021) is one of the most widely adopted methods for parameter-efficient fine-tuning (PEFT) of LLMs due to its simplicity and ability to reduce the number of trainable parameters. Several extensions have been proposed to improve its adaptability and robustness. For instance, dynamic rank adjustment has been explored in AdaLoRA (Zhang et al. 2023) and rank-tuning frameworks (Valipour et al. 2023). Other works enhance LoRA through Bayesian modeling (Yang et al. 2024a), dropout-based noise regularization (Lin et al. 2024), and learning rate strategies (Hayou, Ghosh, and Yu 2024). To mitigate the problem of catastrophic forgetting, recent methods have focused on preserving pre-trained knowledge while adapting to new tasks. These methods typically fall into three categories: (i) singular value decomposition (SVD)-based techniques that constrain updates based on the spectral properties of pre-trained weights (Yang et al. 2024b; Liang and Li 2024), (ii) sparse update strategies (Zhang et al. 2024; Wang et al. 2024a), (iii) interpolation-

based methods (Ren et al. 2024). However, most LoRA variants apply adaptation uniformly across layers or vary the adaptation strength based on parameter sensitivity. In contrast, our method explicitly leverages layer-wise knowledge localization to guide fine-tuning.

Layer-Wise Knowledge Localization

To investigate how knowledge is structurally distributed within LLMs, we analyze the layer-wise causal effects using our modified version of Causal Tracing. Traditional Causal Tracing relies on predefined factual triples (s, r, o) and manual annotation of object o. However, this setup is unsuitable for generative models, where multiple valid outputs may exist. To address this, we measure the causal effects of each layer by tracking the logits over the entire ground-truth output sequence. This modification enables flexible, supervision-free knowledge localization for generative tasks such as math and code, where broader contextual reasoning is required.

Causal Tracing

Causal tracing is a technique for quantifying the causal effects of internal activations on model predictions (Meng et al. 2022). Given a factual prompt corresponding to a knowledge triple (s, r, o), it estimates the causal effect of individual activations by comparing the model's output under corrupted and restored hidden states.

Formally, for a model with L layers and a factual input x, the indirect effect (IE) at token t and layer ℓ is defined as:

$$\text{IE}_{(t,\ell)} = p_\theta\left(o \mid x_{\text{noise}}, v_{(t,\ell)}\right) - p_\theta\left(o \mid x_{\text{noise}}\right) \quad (1)$$

where x_{noise} denotes a corrupted version of the input, and $v_{(t,\ell)}$ is the clean activation restored at token t and layer ℓ. The probability $p_\theta\left(o \mid x_{\text{noise}}\right)$ is computed by performing a forward pass on the corrupted input, while $p_\theta\left(o \mid x_{\text{noise}}, v_{(t,\ell)}\right)$ is obtained by restoring $v_{(t,\ell)}$ at the corresponding location during inference. A high indirect effect indicates that the activation at the corresponding location substantially contributes to producing the correct output.

Layer-Wise Task-specific Knowledge Localization

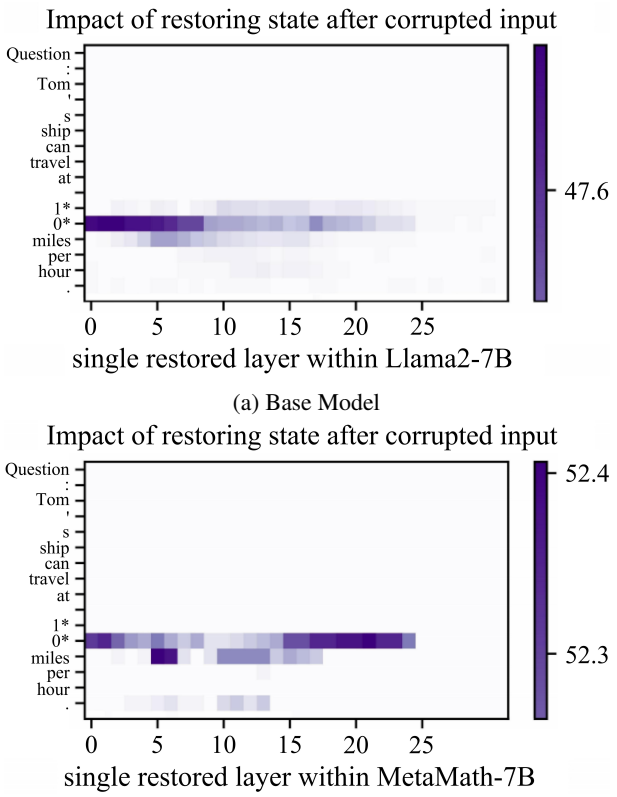
To quantify how task-specific knowledge is distributed across model layers, we measure the causal influence of each layer by tracking the logits over the entire ground-truth output sequence. Specifically, for a given instance k_n, we compute the token-wise indirect effect (IE) at each layer ℓ, and then aggregate these values across all output tokens to obtain a layer-level score:

$$\text{IE}_{(k_n,\ell)} = \sum_{t=1}^{T}\Big(p_\theta\left(y_t \mid x_{\text{noise}}, v_{(t,\ell)}\right) - p_\theta\left(y_t \mid x_{\text{noise}}\right)\Big) \quad (2)$$

where T is the length of the ground-truth token sequence $y = (y_1, y_2, \ldots, y_T)$, and x_{noise} denotes the input obtained by adding Gaussian noise to randomly selected token embeddings of x, following (Meng et al. 2022).

To achieve a task-level knowledge localization for a given task, we average the layer-wise indirect effect across the dataset D of N samples:

$$\text{AIE}_{(D,\ell)} = \frac{1}{N}\sum_{n=1}^{N}\text{IE}_{(k_n,\ell)}. \quad (3)$$



(a) Base Model

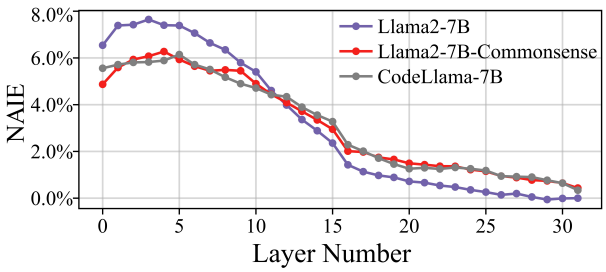
(b) Fine-Tuned Model

Figure 2: Causal Tracing visualization on a GSM8K instance in Llama2-7B (base model) and MetaMath-7B (fine-tuned model). Each heatmap shows the layer-wise indirect effect of restoring hidden states at each token position. Darker colors indicate higher causal effects. Tokens marked with an asterisk (*) denote noised input positions.

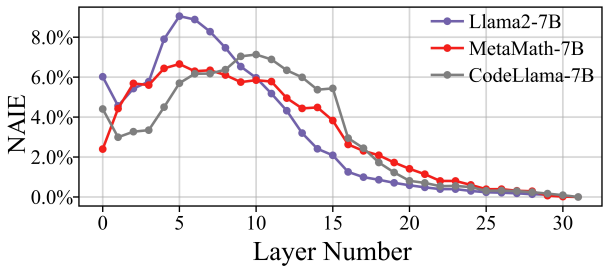
To better reflect the relative contribution of each layer to the overall knowledge distribution, we normalize the average indirect effect (AIE) across all layers, resulting in the normalized AIE (NAIE). We visualize the causal tracing results for a GSM8K (Cobbe et al. 2021) instance in both the base model (Llama2-7B) and the fine-tuned model (MetaMath-7B). Figure 2 shows the results.

We then examine how knowledge is distributed across layers in both base and fine-tuned models. We apply our modified causal tracing method to three domains: commonsense reasoning, mathematical reasoning, and code. As shown in Figure 3, knowledge in the base model is predominantly concentrated in lower layers. After fine-tuning, however, we observe a consistent upward shift, with knowledge being increasingly localized in higher layers across all domains.

To further validate the relationship between knowledge localization and fine-tuning effectiveness, we conduct a layer-



(a) Commonsense Reasoning



(b) Mathematical Reasoning

Figure 3: Layer-wise task-specific knowledge localization for commonsense reasoning and mathematical reasoning. Comparisons between the base model and fine-tuned models: Llama2-7B-Commonsense and CodeLlama-7B in (a), and MetaMath-7B and CodeLlama-7B in (b). Layer-wise task-specific knowledge localization reveals a consistent upward shift from lower to higher layers after fine-tuning.

wise fine-tuning analysis on the Llama2-7B base model using the MetaMathQA dataset (Yu et al. 2023). Specifically, we fine-tune only a single transformer layer at a time using LoRA, and evaluate two key metrics: (1) **Accuracy Gain on Target Tasks**, measured by the average improvement across GSM8K and MATH benchmarks (Hendrycks et al. 2021b); and (2) **Average Accuracy Drop on General Tasks**, measured by the performance degradation on NaturalQuestions (Kwiatkowski et al. 2019), ARC-Challenge, and ARC-Easy (Clark et al. 2018).

Figure 4 shows the results. Notably, the lower layers, which were previously identified as storing general knowledge, yield minimal gains on target tasks but incur substantial losses on general capabilities when fine-tuned. In contrast, higher layers contribute more to target task adaptation with smaller side effects on general tasks. These results provide additional empirical support for our hypothesis: fine-tuning knowledge-intensive layers is more effective and less disruptive, further motivating the design of L2-LoRA.

Our findings provide the first empirical evidence that task-relevant knowledge consistently shifts toward deeper layers following fine-tuning, offering a more interpretable understanding of knowledge representation in LLMs.

Method

Building on our findings that different layers of LLMs specialize in storing distinct types of knowledge, we hypothesize that preserving the parameters of layers associated with

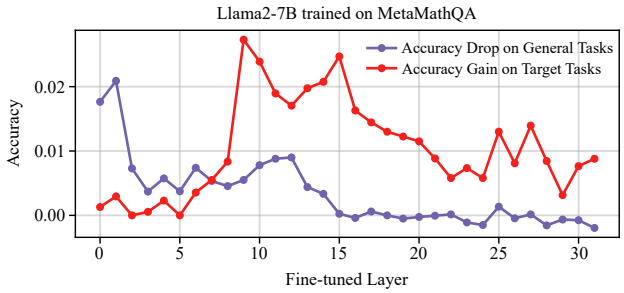


Figure 4: Layer-wise fine-tuning analysis on Llama2-7B using the MetaMathQA dataset. Each point corresponds to fine-tuning a single transformer layer. The red curve shows the accuracy gain on target tasks (measured across GSM8K and MATH), while the blue curve indicates the average accuracy drop on general tasks (NaturalQuestions, ARC-Challenge, ARC-Easy). Fine-tuning deeper layers yields higher adaptation performance with less decrease in general capabilities.

pre-trained world knowledge while encouraging adaptation in other layers can strike a better balance between knowledge preservation and task-specific adaptation. Motivated by this hypothesis, we propose **L2-LoRA**, a layer-specific regularization method that extends Low-Rank Adaptation (LoRA) by applying distinct L_2 regularization to each layer. The strength of regularization is guided by the layer-wise knowledge distribution, as measured by normalized average indirect effect (NAIE). Figure 5 presents an overview of the L2-LoRA pipeline.

Layer-specific L_2 Regularization

L_2 regularization is a classical technique for constraining model complexity and mitigating overfitting by penalizing large weights. It adds a term to the loss function proportional to the squared L_2 norm of the model parameters. The loss function with L_2 regularization can be formulated as:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = \mathcal{L}_0(\mathbf{y}, \hat{\mathbf{y}}) + \lambda \sum_{i=1}^{n} \theta_i^2 \tag{4}$$

where $\mathcal{L}_0$ is the task-specific loss (e.g., cross-entropy), θ_i denotes individual trainable parameters, and λ controls the overall regularization strength.

In standard L_2 regularization training, λ is typically shared across all parameters. However, our findings reveal that different LLM layers encode different types of knowledge, and indiscriminate regularization may damage pre-trained knowledge.

To address this problem, we propose a layer-specific variant of L_2 regularization that assigns a distinct penalty λ_ℓ to each transformer layer ℓ, enabling selective resistance to parameter updates based on the layer-wise knowledge distribution. The modified objective becomes:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = \mathcal{L}_0(f(x), \mathbf{y}) + \sum_{\ell=0}^{L-1} \lambda_\ell \|W_\ell\|_2^2 \tag{5}$$

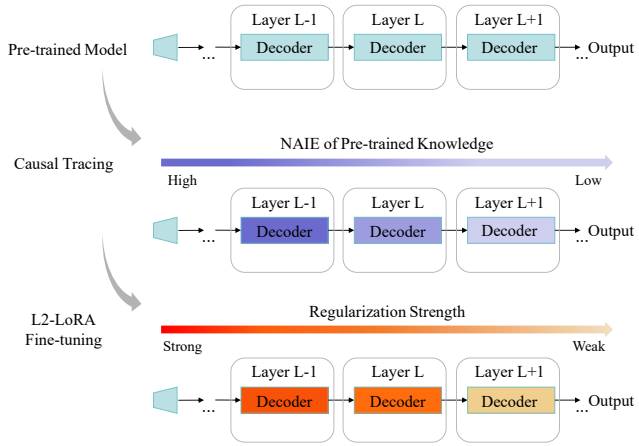


Figure 5: The framework of L2-LoRA. Layer-specific regularization is applied to LoRA updates, guided by the NAIE scores obtained via causal tracing.

where W_ℓ denotes the set of trainable parameters (e.g., LoRA weights) in layer ℓ and λ_ℓ is the layer-specific regularization coefficient.

This formulation enables layer-specific control over which layers are allowed to adapt more freely, providing a mechanism to preserve pre-trained knowledge in critical layers while allowing task-specific adaptation in others.

L2-LoRA: Layer-Specific Regularized Low-Rank Adaptation

Motivated by our findings that pre-trained knowledge is primarily stored in the lower layers of LLMs (as revealed by NAIE), we propose L2-LoRA, a novel regularization method that applies layer-specific L_2 regularization to LoRA weight updates.

LoRA approximates weight updates ΔW during fine-tuning by injecting a pair of low-rank matrices (A, B) into the frozen base model weights W_0. The effective weights become:

$$W = W_0 + \Delta W = W_0 + BA \tag{6}$$

where $\Delta W \in \mathbb{R}^{d\times d}$ is the low-rank update (rank $\ll d$). While prior works have focused on designing expressive low-rank structures, our approach emphasizes controlling the magnitude of ΔW to preserve general knowledge.

To balance adaptation and preservation, we introduce layer-specific L_2 regularization weighted by the importance of each layer in storing general knowledge, as measured by $\text{NAIE}_{(D,\ell)}$. The objective becomes:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = \mathcal{L}_0(f(x), \mathbf{y}) + \eta \sum_{\ell=0}^{L-1} \text{NAIE}_{(D,\ell)} \|\Delta W_\ell\|_2^2 \tag{7}$$

where L is the total number of transformer layers, ΔW_ℓ is the LoRA update at layer ℓ, and η is a tunable hyperparameter.

Although $\text{NAIE}_{(D,\ell)}$ can be directly obtained via causal tracing, this process introduces additional computational overhead before fine-tuning. To alleviate this, we propose an empirical approximation based on the observed localization trend. Specifically, we adopt a cubic decay schedule inspired by Wang et al. (2024a), assigning stronger regularization to lower layers:

$$\text{NAIE}_{(D,\ell)} = \begin{cases} 1 & 0 \le \ell < \ell_i \\ \tau + (1-\tau)\left(1 - \frac{\ell-\ell_i}{\ell_f-\ell_i}\right)^3 & \ell_i \le \ell \le \ell_f \\ \tau & \ell > \ell_f \end{cases} \tag{8}$$

Here, ℓ_i and ℓ_f are the inflection points defining the range of decreasing regularization, and τ is the minimum penalty weight. In our experiments, we set $\ell_i = L/3$ and $\ell_f = 2L/3$, which yielded strong empirical performance.

Experiments

We evaluate the proposed L2-LoRA through two main experimental setups: (1) the preservation of pre-trained knowledge and (2) adaptation to downstream tasks. Additionally, we perform ablation studies to assess the effectiveness of layer-specific knowledge localization and analyze the impact of fine-tuning different layers on task performance.

Experiment Setup

Datasets Our experiments are conducted on three distinct benchmarks:

1. **Mathematical Reasoning**: MetaMathQA (Yu et al. 2023), comprising 395K samples from the GSM8K and MATH datasets.
2. **Commonsense Reasoning**: Commonsense 170K (Hu et al. 2023) is composed of 170K samples, created by formatting the training sets from BoolQ, PIQA, SIQA, HellaSwag, WinoGrande, ARC-e, ARC-c, and OBQA with pre-defined templates.
3. **Instruction Following**: Alpaca dataset (Taori et al. 2023), consisting of 52K instruction-following examples.

Baselines The proposed L2-LoRA aims to effectively retain pre-trained knowledge while acquiring downstream task knowledge. We compare our method with vanilla LoRA and recent variants, including PiSSA (Meng, Wang, and Zhang 2024), MiLoRA (Wang et al. 2024b), and DoRA (Liu et al. 2024). The base models in our experiments are Llama-7B and Llama2-7B.

Implementation Details To evaluate the preservation of pre-trained knowledge, we consider three challenging downstream benchmarks. Specifically, we fine-tune base models on these datasets and assess generalization performance using the MMLU benchmark (Hendrycks et al. 2021a). For the experiments about the adaptation to downstream tasks, we conducted experiments on commonsense reasoning and mathematical reasoning to demonstrate the superiority of L2-LoRA in general fine-tuning. For mathematical reasoning, we follow the training settings of MetaMath (Yu et al. 2023). All other training configurations follow those of LLM-Adapter (Hu et al. 2023).

Model	MMLU					ARC-c
	Humanities	Social Sciences	STEM	Other	AVG	
Llama-7B	29.9	29.4	26.3	33.4	29.8	41.7
LoRA	24.4	21.9	21.5	24.0	23.1	61.3
L2-LoRA	**28.4**	**27.9**	**28.2**	**28.5**	**28.3**	**65.6**

Table 1: Performance on a single task. Llama-7B was trained on the Commonsense 170K dataset with LoRA and L2-LoRA. ARC-c represents the downstream task dataset. The performance on MMLU reflects the preservation of pre-trained knowledge. AVG denotes the average accuracy across all datasets. The results for Llama-7B and LoRA are taken from (Wang et al. 2024a).

Stage	Model	MMLU				
		Humanities	Social Sciences	STEM	Other	AVG
	Llama-7B	29.9	29.4	26.3	33.4	29.8
1	LoRA	25.9	25.6	26.1	24.9	25.7
1	L2-LoRA	28.4	27.9	28.2	28.5	**28.3**
2	LoRA	23.7	22.2	21.3	24.9	23.1
2	L2-LoRA	22.2	22.8	22.2	25.4	**23.8**
3	LoRA	22.2	22.8	22.2	25.4	23.3
3	L2-LoRA	24.4	22.8	21.2	25.3	**23.6**

Table 2: Performance under continual learning. The performance on MMLU reflects the preservation of pre-trained knowledge. The "Stage" number indicates the task order, which follows the sequence: Commonsense Reasoning $\rightarrow$ Mathematical Reasoning $\rightarrow$ Instruction Following.

Preservation of Pre-Trained Knowledge

In this section, we investigate how fine-tuning on downstream tasks affects the retention of pre-trained knowledge. The experiments are conducted following the setup in (Wang et al. 2024a). We fine-tune the model on a challenging downstream task and evaluate its performance on both the target downstream task and general benchmarks.

Single Task We begin by fine-tuning Llama-7B on the Commonsense 170K dataset and evaluate its performance on the MMLU and ARC-c benchmarks. MMLU serves as a proxy for assessing the preservation of pre-trained knowledge, while ARC-c evaluates the model's performance on downstream tasks. The results, presented in Table 1, demonstrate that L2-LoRA achieves superior performance in preserving pre-trained knowledge, with only a 1.5% decline on MMLU. In contrast, the standard LoRA approach results in a significant performance drop of 6.7% on MMLU. These findings highlight that L2-LoRA effectively penalizes parameter updates associated with pre-trained knowledge, leading to better knowledge preservation.

Continual Learning We evaluate the performance of L2-LoRA under continual learning conditions, following the methodology outlined in (Luo et al. 2024). We selected three tasks: Commonsense Reasoning (Task 1), Mathematical Reasoning (Task 2), and Instruction Following (Task 3). The model is trained sequentially on the datasets for each task: Commonsense 170K, MetaMathQA, and Alpaca. We observe that when fine-tuned with LoRA, the average accuracy on MMLU drops from 29.8% to 25.7%, then to 23.1%, and finally to 23.3% after all tasks are completed. In contrast, models fine-tuned with L2-LoRA exhibit a much smaller drop, from 29.8% to 28.3%, then to 23.8%, and finally to 23.6%. These results, shown in Table 2, demonstrate that L2-LoRA outperforms vanilla LoRA in preserving pre-trained knowledge during continual learning, highlighting its effectiveness in mitigating catastrophic forgetting.

Adaptation to Downstream Tasks

Commonsense Reasoning The results for commonsense reasoning are presented in Table 3. We compare our method against the baselines reported in MiLoRA (Wang et al. 2024b). Among these baselines, PiSSA and MiLoRA are specifically designed to mitigate catastrophic forgetting through SVD-based constraints, whereas LoRA and DoRA do not explicitly target this issue. L2-LoRA consistently outperforms all baseline methods across most tasks in the benchmark, with the exception of **BoolQ**, **HellaSwag**, and **ARC-e**. Specifically, on **PIQA**, L2-LoRA achieves an average accuracy increase of 5.3% over LoRA and 1.4% over MiLoRA. On **ARC-c**, L2-LoRA surpasses LoRA and MiLoRA by 4.8% and 0.7%, respectively. These results underscore the effectiveness of L2-LoRA in adapting to downstream tasks.

Mathematical Reasoning Table 4 shows the results of mathematical reasoning. L2-LoRA consistently outperforms all baseline methods across tasks. On GSM8K, L2-LoRA exceeds LoRA and MiLoRA by 4.74 and 1.79 in average accuracy scores, respectively. On MATH, L2-LoRA surpasses LoRA and MiLoRA by 1.24 and 0.36 in average accuracy scores. These results demonstrate the superior performance of L2-LoRA in adapting to mathematical reasoning tasks compared to other methods.

Results Analysis Based on the results, we conclude that L2-LoRA outperforms other LoRA-based methods both in downstream task performance and in retaining more pre-trained knowledge, which can be referred to as generalization

PEFT	Params (%)	BoolQ	PIQA	SIQA	HellaS.	WinoG.	ARC-e	ARC-c	OBQA
LoRA	0.83%	69.8	79.9	79.5	83.6	82.6	79.8	64.7	81.0
DoRA	0.84%	**71.8**	83.7	76.0	**89.1**	82.6	**83.7**	68.2	82.4
PiSSA	0.83%	67.6	78.1	78.4	76.6	78.0	75.8	60.2	75.6
MiLoRA	0.83%	67.6	83.8	80.1	88.2	82.0	82.8	68.8	80.6
L2-LoRA	0.83%	68.9	**85.2**	**79.8**	87.5	**82.8**	83.1	**69.5**	**82.5**

Table 3: Performance comparison across different methods on commonsense reasoning, using Llama2-7B. Results of DoRA are taken from (Liu et al. 2024). Results of LoRA, PiSSA, and MiLoRA are taken from (Wang et al. 2024b).

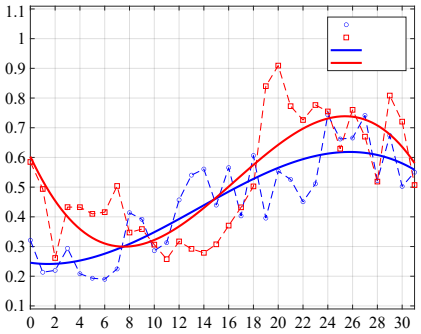
Method	GSM8K	MATH	Avg.
Full FT	66.50	19.80	43.15
LoRA	60.58	16.88	38.73
DoRA	62.28	17.63	40.00
PiSSA	58.23	15.84	37.04
MiLoRA	63.53	17.76	40.65
L2-LoRA	**65.32**	**18.12**	**41.72**

Table 4: Performance comparison of Llama2-7B with different methods on Mathematical Reasoning.

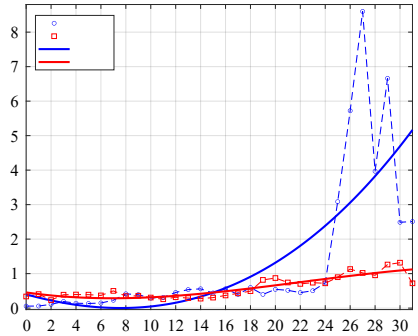
ability. We argue that the superior performance of L2-LoRA on downstream tasks is a direct consequence of its ability to preserve pre-trained knowledge. Our empirical findings, supported by the NAIE$_{(D,\ell)}$, demonstrate that specific layers in LLMs are closely associated with the distribution of pre-trained knowledge and task-specific knowledge. In contrast, previous methods, which apply uniform updates across all layers, risk disrupting the critical pre-trained knowledge in lower layers.

Ablation Study

Do Models Adaptively Fine-Tune Based on Knowledge Localization? L2-LoRA primarily relies on the layer-specific regularization strategy. To investigate whether models can adaptively fine-tune without explicit layer-level regularization control, we conducted experiments on the Llama2-7B model using vanilla LoRA, applied to the MetaMathQA dataset. We quantified the weight changes using the L_2 norm, focusing on the query and value projection matrices within each self-attention module. As shown in Figure 6, the results reveal that weight changes under vanilla LoRA exhibit minimal correlation with knowledge localization. In particular, significant weight changes in the lower layers contribute to the decline in generalization capabilities, a phenomenon observed after LoRA fine-tuning. In contrast, L2-LoRA aligns weight changes with knowledge localization. The lower layers show minimal parameter updates, while the higher layers exhibit more substantial adjustments. This indicates that L2-LoRA effectively penalizes changes in the lower layers while allowing more flexibility in the higher layers. Our results demonstrate that vanilla LoRA fails to adapt to knowledge localization, reinforcing the need for layer-aware regularization to preserve pre-trained knowledge while enhancing task-specific adaptation.



(a) Vanilla LoRA Fine-tuning



(b) L2-LoRA Fine-tuning

Figure 6: Layer-wise weight changes after fine-tuning with vanilla LoRA and L2-LoRA. The blue and red curves denote quadratic polynomial fits of the change magnitudes for the Query and Value projection matrices, respectively. Compared to vanilla LoRA, L2-LoRA suppresses weight updates in lower layers while permitting larger updates in higher layers.

Effectiveness of NAIE$_{(D,\ell)}$ score Based on the analysis of the NAIE$_{(D,\ell)}$ score, we observe that the lower layers of the model primarily store pre-trained knowledge, with minimal adaptation to downstream tasks. Conversely, higher layers predominantly store task-specific knowledge, with less emphasis on pre-trained knowledge. This layer-specific distribution is crucial for fine-tuning strategies. To assess the effectiveness of the NAIE$_{(D,\ell)}$ score, we conducted experiments on mathematical reasoning. Initially, we froze all layers and progressively unfroze them starting from layer 0, evaluating the model's performance at each stage. The results, shown in Table 5, confirm that the NAIE$_{(D,\ell)}$ score accurately reflects the amount of knowledge retained in each layer.

Starting	Evaluation	Number of Trainable Layers			
		0	2	4	6
Layer 0	MMLU	43.95	40.37	41.45	41.71
	AddSub	3.29	3.29	4.05	11.14

Table 5: Performance on general benchmarks and downstream tasks. Layer 0 is the starting layer, and the number of trainable layers gradually increases from lower to higher layers. Fine-tuning only the lowest layers leads to about a 3 point decrease on MMLU, while yielding negligible gains on AddSub.

Starting	Evaluation	Number of Trainable Layers			
		0	2	4	6
Layer 31	MMLU	43.95	43.85	43.68	43.88
	AddSub	3.29	13.42	14.68	17.22

Table 6: Performance on general benchmarks and downstream tasks. Layer 31 is the starting layer, and trainable layers are gradually increased from higher to lower layers. As more layers are made trainable, MMLU performance remains largely stable, whereas AddSub improves more rapidly.

Notably, layers 0 and 1 retain minimal task-specific knowledge, and fine-tuning these layers alone results in negligible improvements on downstream tasks. Similarly, we unfroze layers progressively from the highest layer backward, assessing performance as the number of trainable layers increased. As shown in Table 6, higher layers contain little pre-trained knowledge, so fine-tuning them has minimal impact on generalization performance. However, since these layers capture more task-specific knowledge, fine-tuning higher layers results in faster improvements on downstream tasks compared to fine-tuning the lower layers.

Conclusion

In this work, we present a fine-grained, layer-wise analysis of knowledge localization in LLMs, introducing the Normalized Average Indirect Effect (NAIE) to quantify the task-specific knowledge localization in each layer. Our findings reveal that pre-trained world knowledge is predominantly localized in the lower layers, while fine-tuning shifts task-specific knowledge to deeper layers. Motivated by these insights, we propose L2-LoRA, a novel variant of Low-Rank Adaptation that applies layer-specific L_2 regularization to LoRA parameters. L2-LoRA enhances the efficiency of fine-tuning by imposing stronger regularization on lower layers to preserve general world knowledge, while allowing more flexibility in higher layers to adapt to downstream tasks.

References

AI@Meta. 2024. Llama 3 Model Card.

Bafghi, R. A.; Bagwell, C.; Ravichandran, A.; Shrivastava, A.; and Raissi, M. 2025. Fine tuning without catastrophic forgetting via selective low rank adaptation. *arXiv preprint arXiv:2501.15377*.

Biderman, D.; Portes, J.; Ortiz, J. J. G.; Paul, M.; Greengard, P.; Jennings, C.; King, D.; Havens, S.; Chiley, V.; Frankle, J.; et al. 2024. Lora learns less and forgets less. *arXiv preprint arXiv:2405.09673*.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165.

Chen, Y.; Cao, P.; Chen, Y.; Liu, K.; and Zhao, J. 2024a. Journey to the Center of the Knowledge Neurons: Discoveries of Language-Independent Knowledge Neurons and Degenerate Knowledge Neurons. In Wooldridge, M. J.; Dy, J. G.; and Natarajan, S., eds., *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, 17817–17825. AAAI Press.

Chen, Y.; Cao, P.; Chen, Y.; Liu, K.; and Zhao, J. 2024b. Knowledge Localization: Mission Not Accomplished? Enter Query Localization! arXiv:2405.14117.

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. arXiv:1803.05457.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168.

Dai, D.; Dong, L.; Hao, Y.; Sui, Z.; Chang, B.; and Wei, F. 2022. Knowledge Neurons in Pretrained Transformers. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, 8493–8502. Association for Computational Linguistics.

Ding, N.; Qin, Y.; Yang, G.; Wei, F.; Yang, Z.; Su, Y.; Hu, S.; Chen, Y.; Chan, C.-M.; Chen, W.; et al. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3): 220–235.

Geva, M.; Caciularu, A.; Wang, K. R.; and Goldberg, Y. 2022. Transformer Feed-Forward Layers Build Predictions by Promoting Concepts in the Vocabulary Space. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, 30–45. Association for Computational Linguistics.

Geva, M.; Schuster, R.; Berant, J.; and Levy, O. 2021. Transformer Feed-Forward Layers Are Key-Value Memories. In Moens, M.; Huang, X.; Specia, L.; and Yih, S. W., eds., *EMNLP 2021, Virtual Event / Punta Cana, Dominican Re-*

public, 7-11 November, 2021, 5484–5495. Association for Computational Linguistics.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Han, Z.; Gao, C.; Liu, J.; Zhang, J.; and Zhang, S. Q. 2024. Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey. arXiv:2403.14608.

Hase, P.; Bansal, M.; Kim, B.; and Ghandeharioun, A. 2023. Does Localization Inform Editing? Surprising Differences in Causality-Based Localization vs. Knowledge Editing in Language Models. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *NeurIPS 2023*.

Hayou, S.; Ghosh, N.; and Yu, B. 2024. LoRA+: Efficient Low Rank Adaptation of Large Models. arXiv:2402.12354.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021a. Measuring Massive Multitask Language Understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021b. Measuring Mathematical Problem Solving With the MATH Dataset. arXiv:2103.03874.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685.

Hu, Z.; Wang, L.; Lan, Y.; Xu, W.; Lim, E.; Bing, L.; Xu, X.; Poria, S.; and Lee, R. K. 2023. LLM-Adapters: An Adapter Family for Parameter-Efficient Fine-Tuning of Large Language Models. In Bouamor, H.; Pino, J.; and Bali, K., eds., *EMNLP 2023, Singapore, December 6-10, 2023*, 5254–5276. Association for Computational Linguistics.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; Lavaud, L. R.; Lachaux, M.-A.; Stock, P.; Scao, T. L.; Lavril, T.; Wang, T.; Lacroix, T.; and Sayed, W. E. 2023. Mistral 7B. arXiv:2310.06825.

Jiang, C.; Qi, B.; Hong, X.; Fu, D.; Cheng, Y.; Meng, F.; Yu, M.; Zhou, B.; and Zhou, J. 2024. On Large Language Models' Hallucination with Regard to Known Facts. arXiv:2403.20009.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7: 453–466.

Lee, Y.; Chen, A. S.; Tajwar, F.; Kumar, A.; Yao, H.; Liang, P.; and Finn, C. 2023. Surgical Fine-Tuning Improves Adaptation to Distribution Shifts. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Lei, G.; and Cooper, S. J. 2025. Layerwise Recall and the Geometry of Interwoven Knowledge in LLMs. arXiv:2502.10871.

Liang, Y.-S.; and Li, W.-J. 2024. InfLoRA: Interference-Free Low-Rank Adaptation for Continual Learning. arXiv:2404.00228.

Lin, Y.; Ma, X.; Chu, X.; Jin, Y.; Yang, Z.; Wang, Y.; and Mei, H. 2024. LoRA Dropout as a Sparsity Regularizer for Overfitting Control. arXiv:2404.09610.

Liu, S.-Y.; Wang, C.-Y.; Yin, H.; Molchanov, P.; Wang, Y.-C. F.; Cheng, K.-T.; and Chen, M.-H. 2024. DoRA: Weight-Decomposed Low-Rank Adaptation. arXiv:2402.09353.

Luo, Y.; Yang, Z.; Meng, F.; Li, Y.; Zhou, J.; and Zhang, Y. 2024. An Empirical Study of Catastrophic Forgetting in Large Language Models During Continual Fine-tuning. arXiv:2308.08747.

Meng, F.; Wang, Z.; and Zhang, M. 2024. PiSSA: Principal Singular Values and Singular Vectors Adaptation of Large Language Models. arXiv:2404.02948.

Meng, K.; Bau, D.; Andonian, A.; and Belinkov, Y. 2022. Locating and Editing Factual Associations in GPT. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in neural information processing systems*.

Meng, K.; Sharma, A. S.; Andonian, A.; Belinkov, Y.; and Bau, D. 2023. Mass-Editing Memory in a Transformer. arXiv:2210.07229.

Petroni, F.; Rocktäschel, T.; Riedel, S.; Lewis, P. S. H.; Bakhtin, A.; Wu, Y.; and Miller, A. H. 2019. Language Models as Knowledge Bases? In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 2463–2473. Association for Computational Linguistics.

Pu, G.; Jain, A.; Yin, J.; and Kaplan, R. 2023. Empirical Analysis of the Strengths and Weaknesses of PEFT Techniques for LLMs. arXiv:2304.14999.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Rao, A. T.; and Bhandari, R. N. 2025. Interpreting Internal Representations of Syntax and Semantics in LLMs.

Ren, W.; Li, X.; Wang, L.; Zhao, T.; and Qin, W. 2024. Analyzing and Reducing Catastrophic Forgetting in Parameter Efficient Tuning. arXiv:2402.18865.

Roziere, B.; Gehring, J.; Gloeckle, F.; Sootla, S.; Gat, I.; Tan, X. E.; Adi, Y.; Liu, J.; Sauvestre, R.; Remez, T.; et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

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

Tenney, I.; Das, D.; and Pavlick, E. 2019. BERT Rediscovers the Classical NLP Pipeline. arXiv:1905.05950.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale,

S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Valipour, M.; Rezagholizadeh, M.; Kobyzev, I.; and Ghodsi, A. 2023. DyLoRA: Parameter-Efficient Tuning of Pre-trained Models using Dynamic Search-Free Low-Rank Adaptation. In Vlachos, A.; and Augenstein, I., eds., *EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, 3266–3279. Association for Computational Linguistics.

Wang, H.; Liu, T.; Li, R.; Cheng, M.; Zhao, T.; and Gao, J. 2024a. RoseLoRA: Row and Column-wise Sparse Low-rank Adaptation of Pre-trained Language Model for Knowledge Editing and Fine-tuning. arXiv:2406.10777.

Wang, H.; Xiao, Z.; Li, Y.; Wang, S.; Chen, G.; and Chen, Y. 2024b. MiLoRA: Harnessing Minor Singular Components for Parameter-Efficient LLM Finetuning. arXiv:2406.09044.

xAI. 2025. Grok 3.

Xu, L.; Xie, H.; Qin, S.-Z. J.; Tao, X.; and Wang, F. L. 2023. Parameter-Efficient Fine-Tuning Methods for Pre-trained Language Models: A Critical Review and Assessment. arXiv:2312.12148.

Yang, A. X.; Robeyns, M.; Wang, X.; and Aitchison, L. 2024a. Bayesian Low-rank Adaptation for Large Language Models. arXiv:2308.13111.

Yang, Y.; Li, X.; Zhou, Z.; Song, S. L.; Wu, J.; Nie, L.; and Ghanem, B. 2024b. CorDA: Context-Oriented Decomposition Adaptation of Large Language Models. *arXiv preprint arXiv:2406.05223*.

Young, A.; Chen, B.; Li, C.; Huang, C.; Zhang, G.; Zhang, G.; Li, H.; Zhu, J.; Chen, J.; Chang, J.; Yu, K.; Liu, P.; Liu, Q.; Yue, S.; Yang, S.; Yang, S.; Yu, T.; Xie, W.; Huang, W.; Hu, X.; Ren, X.; Niu, X.; Nie, P.; Xu, Y.; Liu, Y.; Wang, Y.; Cai, Y.; Gu, Z.; Liu, Z.; and Dai, Z. 2024. Yi: Open Foundation Models by 01.AI. *CoRR*, abs/2403.04652.

Yu, L.; Jiang, W.; Shi, H.; Yu, J.; Liu, Z.; Zhang, Y.; Kwok, J. T.; Li, Z.; Weller, A.; and Liu, W. 2023. MetaMath: Bootstrap Your Own Mathematical Questions for Large Language Models. *arXiv preprint arXiv:2309.12284*.

Zhang, Q.; Chen, M.; Bukharin, A.; Karampatziakis, N.; He, P.; Cheng, Y.; Chen, W.; and Zhao, T. 2023. AdaLoRA: Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning. arXiv:2303.10512.

Zhang, W.; Janson, P.; Aljundi, R.; and Elhoseiny, M. 2024. Overcoming Generic Knowledge Loss with Selective Parameter Update. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 24046–24056.