The Scaling Law of LoRA Based on Mutual Information Upper Bound

Anonymous ACL submission

Abstract

LoRA (Low-Rank Adaptation) is a widely used LLM fine-tuning method. During the fine-tuning process, the Scaling Law can guide the selection of the optimal model scale and data complexity to balance model performance and fine-tuning costs. Although existing methods frequently rely on external metrics (e.g., cross-entropy or perplexity) to evaluate model performance, the scaling law may exhibit instability during testing, which is largely attributed to the generalization gap between training and testing. To address this issue, we propose the Mutual Information Upper Bound (MIUB) metric between base modules and LoRA modules, to investigate the Scaling Law in the large-scale LoRA fine-tuning context. The metric gauges the dependency between the general knowledge obtained during pre-training and the task-specific knowledge acquired through LoRA adaptation. In doing so, the metric pays more attention to the distribution changes within the LoRA architecture, so as to evaluate the Scaling Law more robustly. In our experiments, we validated this approach on benchmark datasets, using the Llama3-8B and Phi3-3B models. The results show that the proposed MIUB metric aligns more accurately and stably with the scaling law of LoRA fine-tuning compared to cross-entropy, perplexity and more metrics.

1 Introduction

Pre-trained on vast amounts of data, large language models like GPT-X (Achiam et al., 2023) and LLaMA3 (Dubey et al., 2024) have achieved remarkable results in general domains. However, to address various personalized needs, especially under the pressure of inference deployment costs, fine-tuning serves as an effective method, enhancing the model's personalization and multi-tasking capabilities with relatively small datasets (Kim et al., 2024; Wang et al., 2023; Ge et al., 2023). Among these, LoRA (Low-Rank Adaptation) (Hu et al., 2021; Yang et al., 2024) fine-tuning leverages the idea of low-rank approximation. By freezing the parameters of the large model, it only uses a small number of newly added low-rank parameter matrices to learn the specific knowledge in the new data.

There is no doubt that whether it is LLMs pre-training or fine-tuning, how to controllably balance computing resources and model effects has always been a widely concerned issue. Some work has proposed that there is a scaling law for large model pre-training (Kaplan et al., 2020; Wei et al., 2024), that is, as the size of the LLM increases and the amount of pre-training data increases, the effect of pre-training usually changes regularly. In the pre-training stage, external metrics such as Cross Entropy and Perplexity are usually used to construct Scaling Law for evaluating the model. The existing evaluation metrics primarily focus on assessing the overall distribution of the model. Some work also shows that evaluations based on external metrics are sometimes not stable (Wei et al., 2024).

In the LoRA architecture, the factors that affect the effect of model fine-tuning mainly include the model size, the rank size of LoRA, the amount of data, etc. In addition, there is a natural generalization gap (e.g., distribution shift and knowledge conflict) between training and testing (Xiao, 2024). When there is a large difference between the amount of data and the model size, the impact caused by this generalization gap is not obvious. However, the base module is frozen, and the size of the LoRA module changes relatively little, and the variation in the amount of fine-tuning data is also limited. Therefore, the evaluation of Scaling Law based on external metrics will be disturbed by the generalization gap, leading to instability.

In order to solve the above problems, we shift our perspective to the interior of the LoRA framework. From a general perspective, the effect after fine-tuning is mainly related to two parts of knowledge, one is the meta-knowledge relied on from

the large language model, and the other is the generalized knowledge learned by the newly added parameters (Mao et al., 2024; Jovanovic and Voss, 2024). Some work has shown that when fine-tuning large models, there will be conflicts between new and old knowledge (Shi et al., 2024). Therefore, inspired by the above research, we propose to use the Mutual Information Upper Bound (MIUB) between base modules and LoRA modules to evaluate the Scaling Law in LoRA fine-tuning. MIUB quantitatively analyzes the upper bound of the internal distribution's dependency relationship. This helps reduce the interference of various generalization gaps on the evaluation of the scaling law.

By leveraging the structural advantages of LoRA, it can efficiently calculate the MIUB metric between the output distribution of the large model and that of LoRA. Experimental results show that the MIUB decreases as the size of the large model, the LoRA rank, and the data size (length or complexity) increase. Additionally, the MIUB adheres to the scaling law, is more stable than traditional external evaluation metrics, and better reflects the actual performance trends of the model (such as accuracy). This implies that the MIUB metric enables a more precise selection of the optimal rank and model size configuration, striking a balance between task performance and resource consumption. Furthermore, this paper compares the MIUB size patterns under different prompt templates and contrasts them with the patterns used in fine-tuning.

- An internal metric, the Mutual Information Upper Bound (MIUB), is proposed for the LoRA architecture. By quantitatively analyzing the dependency relationship between the base modules and LoRA modules, it mitigates the instability of the testing Scaling Law caused by the generalization gap.
- Theoretical analysis demonstrates that the Scaling Law derived from MIUB enable more stable assessment of distributional discrepancies between base and LoRA modules, enhancing the stability of performance evaluation in LoRA fine-tuning.
- Empirical results reveal that MIUB not only aligns with the Scaling Law across model sizes and data complexities but also achieves superior robustness and stability compared to traditional metrics like Cross-Entropy, Perplexity (PPL) and more.

2 Related Works

As the scale of large models continues to increase, LoRA is widely used as a lightweight fine-tuning method. However, similar to how the Scaling Law govern pretraining paradigms, establishing LoRA-specific Scaling Law has become crucial for optimizing resource reducing trial-and-error costs. Therefore, this section systematically reviews the research advancements in both Low-Rank Adaptation techniques and Scaling Law theories.

2.1 Scaling Law

The Scaling Law have been a persistent topic in both nature and science (Gan et al., 2021), and in recent years, they have also shown strong guiding capabilities in the field of Large Language Models (LLMs). In the field of neural networks, the Scaling Law critically demonstrate how model performance scales with increases in computational resources, data, and model parameters. (Hu et al., 2021) first introduced the "scaling law" for neural language models, indicating that larger models trained on more data tend to perform better. (Zhang et al., 2024) comprehensively tested the Scaling Law of fine-tuning frameworks under existing evaluation metrics. In the information retrieval domain, (Fang et al., 2024) proposed using contrastive log-likelihood as a metric to assess whether retrieval models adhere to the Scaling Law. In the compression domain, (Wei et al., 2024) introduced the information-theoretic Matrix entropy to measure the performance of large models, showing that Matrix entropy is more accurate and stable compared to the unstable CE (cross-entropy) and PPL (perplexity) metrics. This work has inspired us to investigate the internal relationships within models to evaluate the Scaling Law of LoRA.

2.2 Low-Rank Adaptation

(Hu et al., 2021) was the first to propose the application of LoRA in large models. The core idea of this method is to decompose the weight updates of the model into low-rank matrices, significantly reducing computational costs while maintaining model performance. In recent years, various works have focused on reducing the cost of model fine-tuning and enhancing its generalization capabilities in the design of LoRA structures. (Ding et al., 2023) further reduced the computational cost of LoRA by using gating units to dynamically adjust the intrinsic rank. Additionally, combining LoRA with MoE

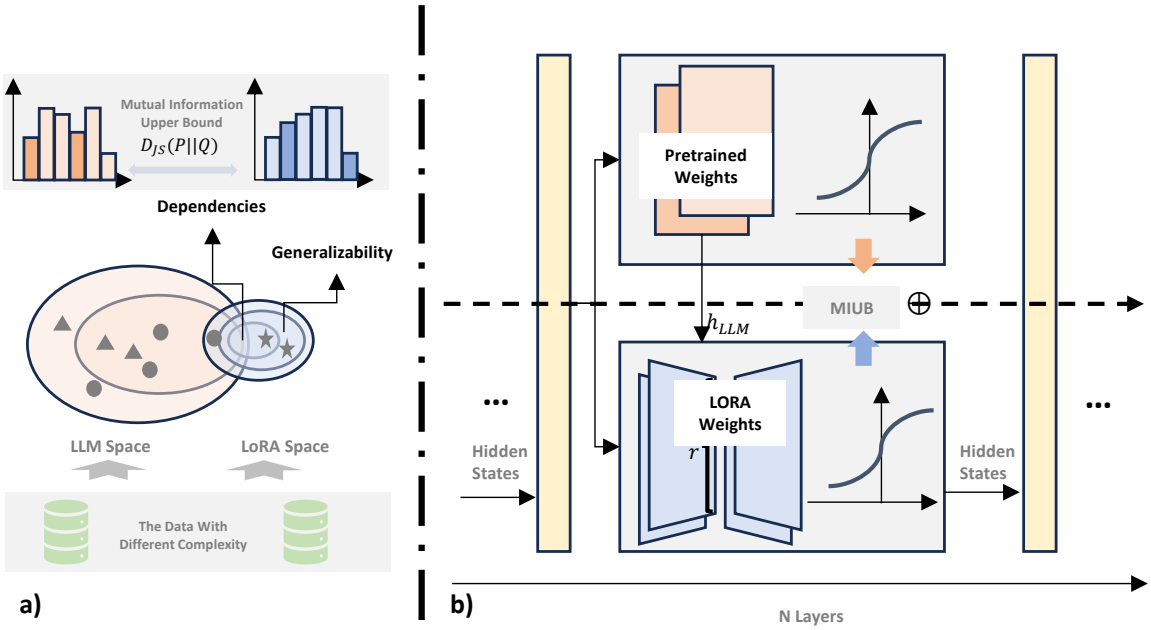

Figure 1: Overall schematic diagram. a) The left figure refers to the dependency between LLM Space and LoRA Space during LoRA fine-tuning. This paper measures this dependency by the upper limit of mutual information, and the generalization of the model will also change accordingly. b) The right figure refers to the LoRA training mode used in this paper and the method of calculating MIUB in this process.

techniques has also provided assurance for enhancing its generalization ability. LoRAHub (Huang et al., 2023) selects different LoRA combinations for task generalization. (Dou et al., 2024) proposed MoELoRA, which utilizes both LoRA and MoE for specific task adjustment and multi-task processing. (Liu et al., 2023) introduced the multimodal learning capabilities of multimodal expert models. During the LoRA fine-tuning process, the parameters of the base modules are frozen. Meanwhile, the LoRA's rank size limits the changes in model size and computational cost. In view of this, the traditional external metrics used to quantify the overall distribution face greater challenges in stably evaluating the Scaling Law. Therefore, by leveraging the architectural advantage of LoRA, we propose a Scaling Law metric that focuses on the internal distribution changes of the model.

3 Methodology

3.1 The Scaling Law of LoRA

For newly added fine-tuning data, without disrupting the feature space of the large model itself (i.e., by freezing the parameters of the large model), LoRA relies on some of the meta-knowledge of the LLM and learns new specific features by adding low-rank parameter weights. Therefore, as shown in Figure 1, there is a natural dependence and generalization relationship between the LLM and LoRA modules. Furthermore, we model the dependency relationship between them as mutual information. which not only measures the information obtained about the distribution of LoRA from the LLM variables but also reveals the extent of their overlap in feature space.

Definition 1 (Mutual Information for LoRA Adaptation)**.** *Let $\boldsymbol{O}$ and $\boldsymbol{L}$ denote the hidden state distributions of the base LLM and LoRA-adapted features, respectively. Their mutual information is defined as:*

$$\mathcal{I}(\boldsymbol{O};\boldsymbol{L}) = \iint p(\boldsymbol{o},\boldsymbol{l}) \log \frac{p(\boldsymbol{o},\boldsymbol{l})}{p(\boldsymbol{o})p(\boldsymbol{l})} \, \mathrm{d}\boldsymbol{o}\mathrm{d}\boldsymbol{l}, \quad (1)$$

where $p(\boldsymbol{o},\boldsymbol{l})$ is the joint distribution, and $p(\boldsymbol{o})$, $p(\boldsymbol{l})$ are marginals. Higher $\mathcal{I}(\boldsymbol{O};\boldsymbol{L})$ indicates stronger dependency between pretrained knowledge and LoRA modules.

In the previous text, we first proposed using mutual information to measure the dependency relationship between the distribution of the base module and that of the Low-Rank Adaptation (LoRA) module. However, the mutual information levels presented by models of different scales vary greatly, making it difficult for us to find a stable dependency pattern. Especially when there are issues such as noise in the feature distribution, the calculation of

mutual information will be significantly affected. In view of this, this paper further designs the Mutual Information Upper Bound (MIUB) as a metric to measure the difference between the distribution of the large model and the LoRA distribution.

Theorem 2 (Mutual Information Upper Bound between Base Modules and LoRA Modules)**.** *Let $\boldsymbol{O}$ and $\boldsymbol{L}$ be random variables representing the output distributions of the base LLM and LoRA module, respectively, with joint distribution $\boldsymbol{P}_{OL}$ and marginal distributions $\boldsymbol{P}_O$ and $\boldsymbol{P}_L$. The mutual information $\mathcal{I}(\boldsymbol{O};\boldsymbol{L})$ is bounded by:*

$$\mathcal{I}(\boldsymbol{O};\boldsymbol{L}) \leq 2 \cdot \mathcal{D}_{JS}(\boldsymbol{P}_{OL} \| \boldsymbol{P}_O \boldsymbol{P}_L), \tag{2}$$

where $\mathcal{D}_{JS}$ denotes the Jensen-Shannon divergence. The proof can be found in Appendix A. As shown in Figure 1, large models acquire rich meta-knowledge, and the knowledge from new data absorbed by the LoRA module inevitably depends on the features already learned by the large model. Specifically, a larger MIUB value implies a stronger dependency relationship between the base modules and the LoRA modules, and a higher degree of overlap in the distribution space. This often indicates that the LoRA module learns less domain-specific knowledge from the new data, and generally, the actual performance of the model will also deteriorate accordingly.

By introducing the MIUB to measure dependence within the LoRA architecture, it ensures that the dependence will not decrease indefinitely during the data measurement process, but will instead stabilize within a certain range and approach its upper bound. This approach provides a theoretical upper limit for the dependence in the LoRA architecture, while also guaranteeing its convergence, ensuring that the dependence ultimately stabilizes within a finite range and avoiding excessive fluctuations or infinite reduction. Furthermore, we will give two corollaries: one is the Scaling Law based on the MIUB metric, and the other is why MIUB is closer to the actual model performance than other metrics.

Corollary 1. *Here is a scaling law that focuses on the model size, LoRA rank size, and dataset size during LoRA fine-tuning:*

$$\begin{aligned} \mathit{MIUB}(N,R,D) = A\left(\frac{N_0}{N}\right)^{\alpha} + B\left(\frac{R_0}{R}\right)^{\beta} \\ + C\left(\frac{D_0}{D}\right)^{\gamma}. \end{aligned} \tag{3}$$

where $\mathit{MIUB}(N,R,D)$ is the metric as a function of the number of parameters in the large model N, the LoRA rank size R, and the dataset size D. N_0, R_0, D_0 are scaling constants that normalize the respective terms. α, β, γ are scaling exponents that describe how the MIUB scales with respect to the model size, LoRA rank size, and dataset size, respectively. A, B, C are constants that depend on the specific problem and architecture.

Assumption 3. *This paper assumes that the generalization gap G_{gap} is the main reason for the instability of the scaling law in the LoRa framework. G_{gap} arises from the difference between the training error and the testing error for unseen labelswork (Xiao, 2024), which encompasses factors such as distribution shifts and knowledge conflicts. Formally, we can express it as:*

$$G_{gap} = E_{te}^{unseen} - E_{tr} = \sum_{i=1}^{n} I_i + M_{dist}, \tag{4}$$

where I_i denotes the i-th contributing error, and M_{dist} represents the error caused by the distribution evaluation metric.

Corollary 2. *Based on the above hypothesis, the discrepancy between the Scaling Law's evaluation metric M and the model's actual performance Y is defined as the generalization gap G_{gap}, as expressed in the following equation.*

$$Y - M = \sum_{i=1}^{n} I_i + M_{dist}, \tag{5}$$

In the context of LoRA, variations in rank have a relatively limited impact on model size and computational cost. Traditional metrics primarily focus on overall distributional changes, and M_{dist} is influenced by the parameters and knowledge inherited from the base model distribution. On one hand, this influence increases the error, enlarging G_{gap} and further deviating from the model's actual performance. On the other hand, traditional metrics may exhibit insensitivity to distributional shifts specific to the LoRA components.

In contrast, the Mutual Information Upper Bound (MIUB) quantifies the dependencies between internal distributions, allowing for a more effective evaluation of the model's generalization ability when learning new data. Therefore, MIUB serves as a more reliable scaling law metric in LoRA fine-tuning, providing greater stability in assessing model adaptability and performance.

3.2 Calculating MIUB in LoRA Architecture

The paper adds LoRA structures to all the Dense Linear layers in the Attention and FFN modules of a large model. The original parameters of the large model are frozen, and only the LoRA components are trained during fine-tuning. Specifically, the hidden states of the large model are denoted as h^m_{LLM}, and the hidden states of LoRA, h^m_{lora}, are obtained by adding the hidden states of the large model to the output of LoRA.

The hidden states h^m_{LLM} and h^m_{LoRA} are converted into probability distributions using the softmax function. Then, the MIUB between these two probability distributions is calculated, as shown in Figure 1. By summing the MIUBs of all the LoRA components, we obtain the MIUB for a single sample. The average MIUB across all samples gives the final evaluation value:

$$M = \frac{1}{N} \sum_{\omega_{\text{set}}} \sum_{m} D^m_{JS}(P \| Q) \tag{6}$$

where $D^m_{JS}(P||Q)$ represents the Jensen-Shannon divergence between the probability distributions P and Q for the m-th component.

As shown in the Appendix B, we employed prompt learning during the fine-tuning of the large model. Taking a classification task as an example, the Train Prompt instructs the model to select the correct option and serves as a zero-shot template. During testing, in addition to the zero-shot prompt, we also have the option to use a 1-shot template (Test Prompt 1), which includes one positive and one negative example, a few-shot template, and a template that imposes restrictions on the task output (Test Prompt 3). We also evaluate the model's performance across different prompt templates.

4 Experiments

In this section, we will evaluate the proposed model structure on natural language tasks and verify the effectiveness of various measures we use. All experiments were performed on NVIDIA A800 GPU.

4.1 Model and Hyperparameters

We use Llama3 (Dubey et al., 2024) and Phi3-3B (Abdin et al., 2024) as our testing model, Llama3 has 8B parameters and 32 layers and Phi3 has 3B parameters and 32 layers, we use them to test the best model settings on models of different sizes. We use Adam as the optimizer with a learning rate of 4×10^{-5} for fine-tuning downstream tasks and set the batch size to 32.

4.2 Dataset and Metrics

We use our proposed structure on five popular zero-shot generation tasks, including PIQA (Bisk et al., 2020), ARC-Challenge (Clark et al., 2018), ARC-Easy (Clark et al., 2018), Winogrande (Sakaguchi et al., 2021), and HellaSwag (Zellers et al., 2019), with higher accuracy, indicating that Mooe has a stronger parameter fine-tuning ability to handle downstream tasks.

For perplexity verification, we chose two datasets: Wiki2 (Merity et al., 2016) and PTB (Marcus et al., 1994). Lower Perplexity indicates that the compressed model has a stronger ability to maintain the output distribution of the original model.

In addition, we use Metaphor Understanding Challenge (MUNCH) (Tong et al., 2024) dataset. Given that metaphor understanding is significantly challenging for large language models (LLMs), this test can effectively verify the applicability of each metric in evaluating the fine-tuning effect .

Seven metrics, Accuracy (ACC), Cross-Entropy (CE), Perplexity (PPL), Cosine similarity (COS), and Euclidean distance (EU), Mutual Information (MI) and Mutual Information Upper Bound (MIUB), were used for experimental evaluation.

4.3 Scaling Law Setting

- For the scaling settings of the LoRA components, we primarily adjusted the rank to different sizes, specifically 32, 128, 512.

- For the large model, we applied a parameter-sharing compression method to adjust the scaling of the model. To ensure that the model's basic performance is not unfairly affected or that abnormal experimental results do not occur due to compression, we fixed the first 16 layers of the Phi3 and llama3 models and applied different parameter-sharing strategies to the last 16 layers: sharing every eight layers ($share_8$), every four layers ($share_4$), every two layers ($share_2$), and no sharing ($share_1$).

- In the data scaling section, we selected 100 data samples from each of the test sets across multiple tasks, with data lengths in the ranges of [1, 100], [101, 200], and [201, 300].

4.4 Main Results

We conducted experiments on seven benchmark datasets, where AVG refers to the average value of ARC-Easy, ARC-Challenge, HellaSwag, PIQA

Table 1: Experiments comparing the performance of the Scaling Law under MIUB and Cross-Entropy metrics through controlled configurations of LoRA ranks and model sizes. Black arrows indicate the trend of model size scaling for the ACC metric, red arrows signify that the trends of the CE or MIUB metrics deviate from the ACC trend, while green arrows denote alignment with the ACC trend.

Dataset	Model	Metrics	32	128	512	$share_8$	$share_4$	$share_2$	$share_1$
ARC-Easy	Phi3	ACC	0.951	0.951	0.969	0.955	0.949	0.951	0.955
		CE	0.018	0.162	0.007	23.977	9.647	2.514	0.077
		MIUB	1586.0	1566.6	1567.7	1902.6	1712.6	1643.0	1597.3
	LLaMA3	ACC	0.873	0.846	0.862	0.007	0.901	0.894	0.912
		CE	12.924	11.789	13.427	8.741	11.594	13.399	14.339
		MIUB	3420.8	3407.1	3398.8	3831.8	3579.5	3443.7	3405.1
ARC-Challenge	Phi3	ACC	0.859	0.863	0.887	0.849	0.863	0.887	0.873
		CE	0.018	0.159	0.007	24.661	9.606	2.510	0.070
		MIUB	1596.2	1579.1	1578.3	1913.3	1728.5	1657.4	1607.6
	LLaMA3	ACC	0.754	0.782	0.792	0.007	0.154	0.816	0.823
		CE	12.018	10.444	12.342	9.033	13.328	11.295	13.463
		MIUB	3456.9	3447.6	3439.3	3861.9	3613.7	3485.0	3446.6
HellaSwag	Phi3	ACC	0.809	0.796	0.814	0.789	0.809	0.789	0.813
		CE	0.009	0.191	0.014	18.215	14.896	0.057	0.007
		MIUB	1521.4	1518.7	1511.6	1884.9	1685.9	1608.0	1542.1
	LLaMA3	ACC	0.840	0.866	0.875	0.279	0.817	0.872	0.882
		CE	0.013	0.005	0.010	16.376	7.447	2.124	0.014
		MIUB	3287.5	3272.7	3258.3	3765.5	3475.7	3282.1	3282.0
PIQA	Phi3	ACC	0.848	0.849	0.836	0.852	0.859	0.858	0.863
		CE	0.020	0.362	0.012	17.117	13.471	0.494	0.065
		MIUB	1610.5	1590.2	1586.4	1906.6	1742.5	1693.0	1621.0
	LLaMA3	ACC	0.858	0.871	0.900	0.620	0.837	0.891	0.891
		CE	13.441	8.074	12.947	16.547	12.926	12.793	10.299
		MIUB	3509.1	3498.1	3485.9	3882.3	3658.6	3540.3	3497.6
Winogrande	Phi3	ACC	0.815	0.822	0.814	0.817	0.803	0.821	0.830
		CE	0.024	0.242	2.452	23.470	11.055	9.077	0.004
		MIUB	1543.6	1530.5	1531.4	1883.2	1679.8	1616.7	1557.5
	LLaMA3	ACC	0.799	0.822	0.866	0.494	0.725	0.855	0.863
		CE	0.211	1.125	4.984	5.591	9.211	7.702	4.700
		MIUB	3246.5	3232.1	3217.7	3744.3	3407.2	3238.6	3233.0
AVG	Phi3	ACC	0.856	0.856↑	0.864↑	0.852	0.857↑	0.861↑	0.867↑
		CE	0.018	0.251↑	0.602↑	20.110	12.695↓	2.763↓	0.035↓
		MIUB	**1564.6**	**1551.5**↓	**1548.6**↓	**1894.5**	**1706.7**↓	**1642.2**↓	**1579.1**↓
	LLaMA3	ACC	0.825	0.837↑	0.859↑	0.281	0.550↑	0.867↑	0.874↑
		CE	5.519	4.056↓	6.446↑	9.345	10.181↑	7.544↓	5.803↓
		MIUB	**3360.2**	**3346.9**↓	**3334.0**↓	**3803.9**	**3527.8**↓	**3368.6**↓	**3350.2**↓

and Winogrande datasets. The experimental results show that the MIUB has two conclusions:

MIUB changes regularly with the change of model size. The bolded text in Table 1 indicates that MIUB follows the pattern of decreasing as the model size increases. From the the average results (AVG) the ranks of LoRA are set to 32, 128 and 512 respectively. As the ranks increase, that is, the size of LoRA increases, MIUB gradually decreases, which means that LoRA relies less on the features of the large model and has stronger generalization. In the right half of the table, the sizes of the large model are set to $share_8$, $share_4$, $share_3$, $share_1$, corresponding to models with 18 layers, 20 layers, 24 layers, and 32 layers of parameters, respectively. Notably, the computational FLOPs remain constant across all configurations. As the size of the LLM base modules changes, MIUB also decreases. It is worth noting that the change of the rank of the LoRA part has little effect on the model size, so the change of MIUB is small, while the change of the large model size is large, so the change of MIUB is larger. Additionally, as shown in Figure 2, we present the MIUB and PPL with respect to the size of the large model for the PTB and Wiki2 language modeling tasks. The experimental results indicate that, with the increase in the number of parameters, MIUB exhibits a significant decreasing trend, demonstrating that the scaling law holds.

Compared to traditional metrics, MIUB not only better reflects the changing trend of actual effects but also exhibits greater stability in ad-

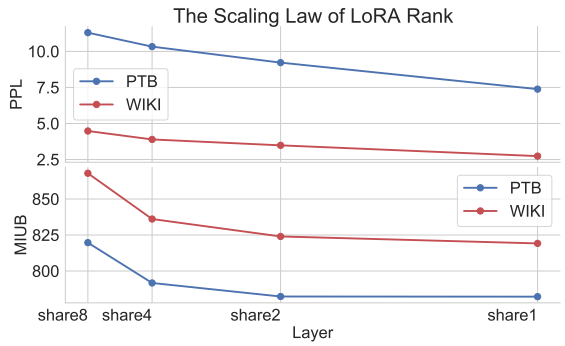

Figure 2: Comparative experiment between MIUB and PPL under LLM size changes.

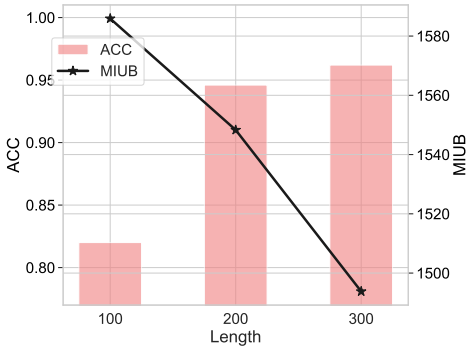

Figure 3: The scaling law of data complexity.

hering to the scaling law. We analyze this from two perspectives: first, by comparing the trends of CE, MIUB, and actual performance (ACC), and second, by conducting a comparative analysis of the Scaling Law based on PPL, CE, and MIUB. As shown in Table 1, in the AVG test based on the Phi3 model, an abnormal increase in CE was observed (indicated by the red arrow) even as ACC improved. In contrast, MIUB consistently decreased as ACC increased, indicating that during fine-tuning, the model's dependency on the larger model weakened, leading to stronger learning of generalized knowledge. Regarding the stability of the scaling law, compared to CE, which exhibited a significant abnormal increase as the rank increased, MIUB consistently maintained a steady decline. As the size of the larger model increased, calculations revealed that the CE value at $share_8$ was 571 times that of $share_1$, while the size of the larger model increased by less than twice. In comparison, MIUB's change was more stable, decreasing by 17%. This effect is more pronounced in Figure 2. The change in large model size is not linear; the increase becomes more significant. Additionally, under limited data conditions, although the model tends to learn more generalized knowledge, the complexity of the large model inevitably increases. Therefore, while MIUB still shows a decreasing trend, the rate of decrease will correspondingly diminish. The reason for the above experimental results is that MIUB measures the changes in the distribution relationship within the LoRA architecture, making it more sensitive to the evaluation of the model's generalization ability (performance) in the context of the Scaling Law.

Additionally, the trend diagrams for the five datasets referenced in Table 1 can be found in Appendix C.

4.5 Data Complexity

To study the data size scaling law based on MIUB, we conducted experiments analyzing data complexity. As described in the Scaling Law Setting section, the data was divided according to length, and the experimental results are shown in Figure 3. The results indicate that as the data length increases, the actual performance (ACC) of the large model improves, suggesting that the large model acquires more comprehensive information from the prompt. At the same time, MIUB exhibits a systematic decrease, indicating that as the new data becomes more complex (larger in scale), the dependency on the large model during fine-tuning diminishes, and there is a greater need for the LoRA module to learn more generalized knowledge.

4.6 Prompt Learning Analysis

This paper uses four types of prompt templates, as shown in Table 2. "Main" refers to the zero-shot prompt used for training, while "Prompt1" "Prompt2" and "Prompt3" are one-shot templates, few-shot templates with positive and negative examples, and output control templates, respectively. The experimental results show that regardless of the template used, MIUB decreases with the increase in LoRA and the size of the large model, demonstrating the stability of MIUB as a scaling law effectiveness metric. Comparing the four prompts, the order is generally: MIUB (Prompt1) $>$ MIUB (Main) $>$ MIUB (Prompt3) $>$ MIUB (Prompt2). Prompt1 has the highest MIUB because it incorporates data from the training set, which enhances the dependency on the large model during the knowledge learning process. In contrast, Prompt2, due to the inclusion of negative examples, has greater uncertainty and thus a smaller MIUB.

Table 2: The results of different prompt on Phi3-3B Model.

Dataset	Matrices	32	128	512	$share_8$	$share_4$	$share_2$	$share_1$
AVG	Main (ACC)	0.856	0.856	0.864	0.852	0.857	0.861	0.867
	Main (MIUB)	**1564.6**	**1551.5**	**1548.6**	**1894.5**	**1706.7**	**1642.2**	**1579.1**
	Prompt1 (ACC)	0.854	0.851	0.854	0.836	0.832	0.853	0.865
	Prompt1 (MIUB)	**1569.7**	**1556.4**	**1537.4**	**1900.5**	**1704.7**	**1652.0**	**1591.0**
	Prompt2 (ACC)	0.855	0.850	0.852	0.838	0.820	0.853	0.865
	Prompt2 (MIUB)	**1533.0**	**1525.7**	**1516.1**	**1889.1**	**1682.9**	**1609.4**	**1552.0**
	Prompt3 (ACC)	0.869	0.866	0.870	0.859	0.856	0.869	0.872
	Prompt3 (MIUB)	**1545.2**	**1541.6**	**1534.7**	**1894.7**	**1711.0**	**1635.5**	**1573.1**

Table 3: Comparative experiments between MIUB and other metrics on the MUNCH dataset

Matrices	32	128	512
ACC	0.712	0.712	0.712
CE	6.299	8.969 ↑	9.832 ↑
COS	255.3	255.5 ↑	255.6 ↑
EU	3982.5	2086.1 ↓	1426.8 ↓
MI(10^{-1})	1435.9	1435.8 ↓	1436.0 ↑
MI(10^{-3})	2036.9	2036.9	2048.0 ↑
MI	4071.4	4071.4	4064.6↓
MIUB(10^{-1})	2835.2	2835.2	2835.2
MIUB(10^{-3})	2899.5	2899.5	2881.2↓
MIUB	4071.4	4071.4	4064.6↓

4.7 Comparison with More Matrices

To evaluate the Scaling Law of relevant datasets where the LLM base model performs poorly, we conduct a comparative analysis of five Scaling Law metrics with different rank sizes, including the Mutual Information Upper Bound (MIUB). In Table 3, cross-entropy (CE) is an external metric, and cosine (COS) similarity, Euclidean distance (EU), and mutual information (MI) are internal metrics for assessing the model's internal distribution changes based on this paper's calculation method. Notably, to fully compare the performance differences between MIUB and MI, we use the method of adding noise to test which metric is more robust.

During the process of changing the rank from 32 to 512, the model's actual accuracy remained unchanged, with only the model size being altered. The external cross-entropy (CE) and internal cosine similarity (COS) metrics both showed an upward trend, which deviated from the model's actual performance. Meanwhile, the Euclidean distance (EU) metric dropped significantly, also failing to mirror the model's true performance. Notably, without noise in the distributions of the base and LoRA modules, the values of the Mutual Information Upper Bound (MIUB) and mutual information (MI) are equal, align with the model's actual effect, and remain basically stable. However, once the distributions are disturbed by noise, regardless of whether the noise level is low (0.001) or high (0.1), the mutual information (MI) exhibits an abnormal increasing trend. In stark contrast, even when the noise level reaches 0.1, the Mutual Information Upper Bound (MIUB) can still conform to the actual performance of the model. In conclusion, it can be concluded that for LoRA fine-tuning, the MIUB is a more robust Scaling Law metric.

5 Conclusion

In order to reduce the trial-and-error cost of LoRA fine-tuning, this paper proposes the Mutual Information Upper Bound (MIUB) metric for evaluating the general LoRA framework, and systematically explores the Scaling Law of MIUB with respect to LLM size, rank size of the LoRA module, and data size. Specifically, by leveraging the structural advantages of LoRA, this paper calculates MIUB based on the output distributions of the LLM's frozen layer and the LoRA module, and quantitatively evaluates the generalization gap between training and testing by quantifying the dependency between the LoRA module and the base model. Experiments on eight benchmark datasets and two general large models, LLaMA3-8B and Phi3-3B, show that the MIUB not only aligns with the Scaling Law , but also provides more robust and stable results compared to traditional general metrics.

6 Limitations

In terms of fine-tuning large models using LoRA, this study, conducted on models of the same scale while controlling for parameters and data complexity, demonstrates that the scaling law based on the mutual information upper bound exhibits more consistent and stable trends compared to other metrics. However, there are several key limitations: the experiments have not been conducted on different scales within the same series of LLMs (e.g., LLaMA3-8B, LLaMA3-14B, etc.), validation on a broader range of LLMs such as DeepSeek, Qwen2.5, GPT, GLM4, and others is required. We plan to address these limitations in future work with more extensive experimental studies.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

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

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, et al. 2024. Loramoe: Alleviating world knowledge forgetting in large language models via moe-style plugin. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1932–1945.

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

Yan Fang, Jingtao Zhan, Qingyao Ai, Jiaxin Mao, Weihang Su, Jia Chen, and Yiqun Liu. 2024. Scaling laws for dense retrieval. *Preprint*, arXiv:2403.18684.

Zhengtao Gan, Orion L Kafka, Niranjan Parab, Cang Zhao, Lichao Fang, Olle Heinonen, Tao Sun, and Wing Kam Liu. 2021. Universal scaling laws of keyhole stability and porosity in 3d printing of metals. *Nature communications*, 12(1):2379.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. 2023. In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945*.

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

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2023. Lorahub: Efficient cross-task generalization via dynamic lora composition. *arXiv preprint arXiv:2307.13269*.

Mladjan Jovanovic and Peter Voss. 2024. Trends and challenges of real-time learning in large language models: A critical review. *arXiv preprint arXiv:2404.18311*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Jeonghoon Kim, Jung Hyun Lee, Sungdong Kim, Joonsuk Park, Kang Min Yoo, Se Jung Kwon, and Dongsoo Lee. 2024. Memory-efficient fine-tuning of compressed large language models via sub-4-bit integer quantization. *Advances in Neural Information Processing Systems*, 36.

Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. 2023. Moelora: An moe-based parameter efficient fine-tuning method for multi-task medical applications. *arXiv preprint arXiv:2310.18339*.

Yuren Mao, Yuhang Ge, Yijiang Fan, Wenyi Xu, Yu Mi, Zhonghao Hu, and Yunjun Gao. 2024. A survey on lora of large language models. *arXiv preprint arXiv:2407.11046*.

Mitch Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The penn treebank: Annotating predicate argument structure. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Dan Shi, Renren Jin, Tianhao Shen, Weilong Dong, Xinwei Wu, and Deyi Xiong. 2024. Ircan: Mitigating knowledge conflicts in llm generation via identifying and reweighting context-aware neurons. *arXiv preprint arXiv:2406.18406*.

Xiaoyu Tong, Rochelle Choenni, Martha Lewis, and Ekaterina Shutova. 2024. Metaphor understanding challenge dataset for LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3517–3536, Bangkok, Thailand. Association for Computational Linguistics.

Jue Wang, Yucheng Lu, Binhang Yuan, Beidi Chen, Percy Liang, Christopher De Sa, Christopher Re, and Ce Zhang. 2023. Cocktailsgd: Fine-tuning foundation models over 500mbps networks. In *International Conference on Machine Learning*, pages 36058–36076. PMLR.

Lai Wei, Zhiquan Tan, Chenghai Li, Jindong Wang, and Weiran Huang. 2024. Large language model evaluation via matrix entropy. *arXiv preprint arXiv:2401.17139*.

Lechao Xiao. 2024. Rethinking conventional wisdom in machine learning: From generalization to scaling. *arXiv preprint arXiv:2409.15156*.

Shu Yang, Muhammad Asif Ali, Cheng-Long Wang, Lijie Hu, and Di Wang. 2024. Moral: Moe augmented lora for llms' lifelong learning. *arXiv preprint arXiv:2402.11260*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. 2024. When scaling meets llm finetuning: The effect of data, model and finetuning method. *arXiv preprint arXiv:2402.17193*.

A The proof of Theorem 2

Proof. Let $\boldsymbol{O}$ represent the output distribution from the base LLM module, and $\boldsymbol{L}$ denote the output distribution introduced by the LoRA adaptation module. The mutual information between these two distributions can be written as the Kullback-Leibler (KL) divergence between their joint distribution $\boldsymbol{P}_{OL}$ and the product of their marginals $\boldsymbol{P}_O \otimes \boldsymbol{P}_L$:

$$\mathcal{I}(\boldsymbol{O};\boldsymbol{L}) = D_{\mathrm{KL}}\left(\boldsymbol{P}_{OL} \| \boldsymbol{P}_O \otimes \boldsymbol{P}_L\right), \quad (7)$$

where $\boldsymbol{P}_O$ captures the knowledge learned by the base modules during pre-training, and $\boldsymbol{P}_L$ represents the task-specific knowledge gained through the LoRA module. Next, the Jensen-Shannon (JS) divergence between the joint distribution (representing the interaction between the base model and LoRA) and the product of independent marginals (representing the decoupling of base and LoRA) is defined as:

$$\begin{aligned}\mathcal{D}_{\mathrm{JS}}\left(\boldsymbol{P}_{OL} \| \boldsymbol{P}_O \otimes \boldsymbol{P}_L\right) = &\frac{1}{2} D_{\mathrm{KL}}\left(\boldsymbol{P}_{OL} \| \boldsymbol{M}\right) \\ &+ \frac{1}{2} D_{\mathrm{KL}}\left(\boldsymbol{P}_O \otimes \boldsymbol{P}_L \| \boldsymbol{M}\right),\end{aligned} \quad (8)$$

where $\boldsymbol{M} = \frac{1}{2}(\boldsymbol{P}_{OL} + \boldsymbol{P}_O \otimes \boldsymbol{P}_L)$ serves as the midpoint distribution. Using the convexity property of KL divergence, we derive the following fundamental inequality:

$$\mathcal{I}(\boldsymbol{O};\boldsymbol{L}) \leq 2 \cdot \mathcal{D}_{\mathrm{JS}}\left(\boldsymbol{P}_{OL} \| \boldsymbol{P}_O \otimes \boldsymbol{P}_L\right). \quad (9)$$

□

B Prompts

Prompt

Train Prompt
Choose the correct answer for the following question: x_i
Answer: y_i
Test Prompt 1
Choose the correct answer for the following question. Here is an example shown below: s_i
The new question is: x_i
Answer: y_i
Test Prompt 2
Choose the correct answer for the following question. Here is an positive example shown below: s_i
Here is an negative example shown below: n_i
The new question is: x_i
Answer:
Test Prompt 3
Please note that you can only choose from A, B, c or D. Choose the correct answer for the following question: x_i
Answer:

C Scaling Law Trend Chart

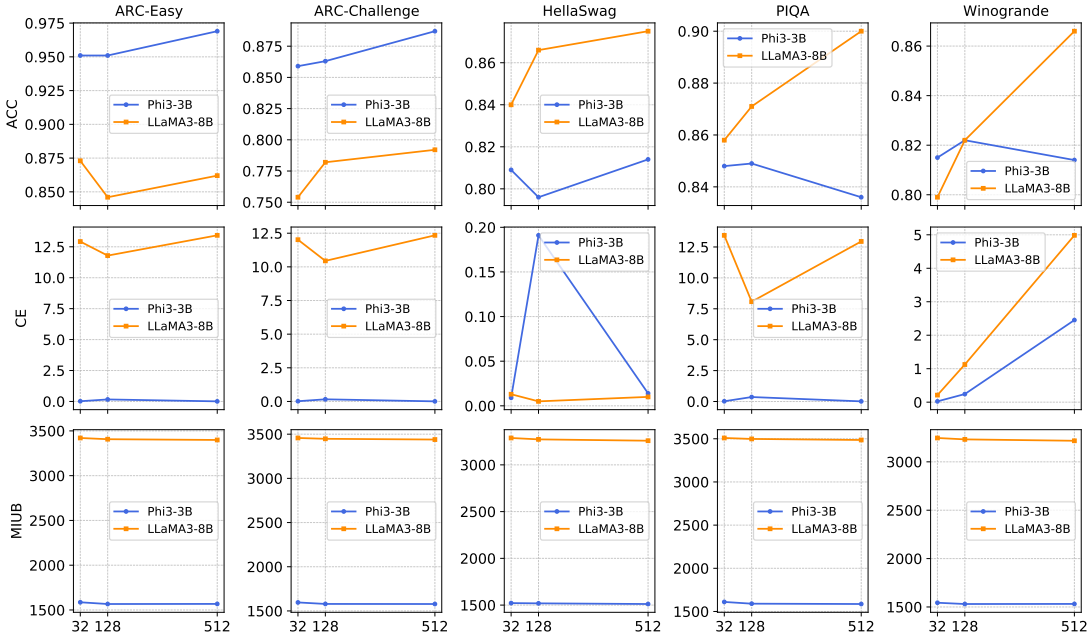

Figure 4: Evaluate the effect of model testing with changing LoRA rank.

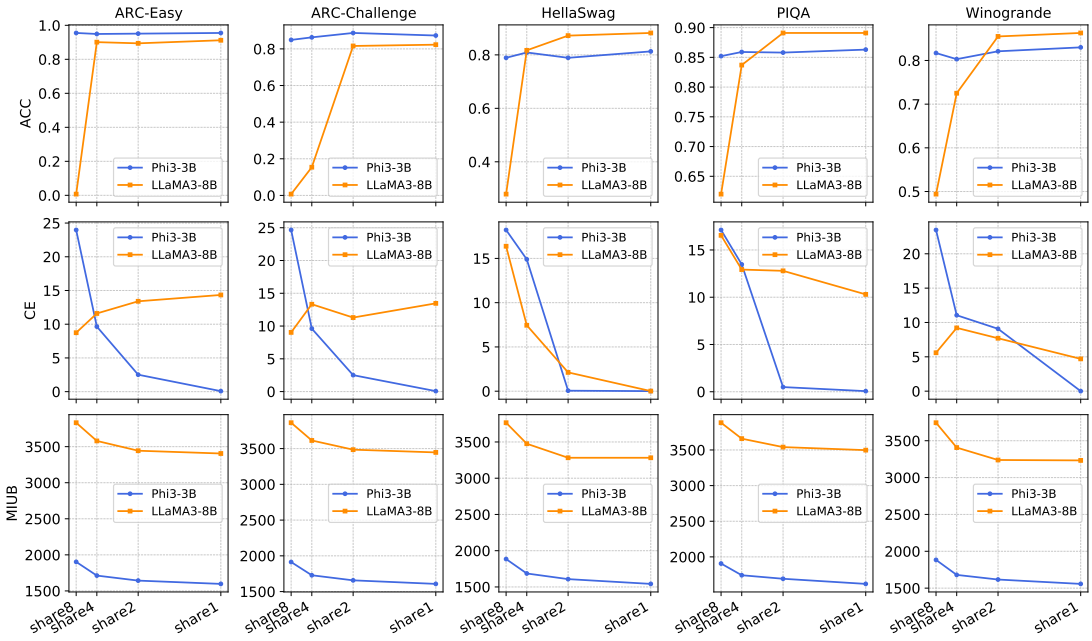

Figure 5: Evaluate the effect of model testing with changing LLM size based on parameter sharing.