# Beyond Full Fine-tuning: Harnessing the Power of LoRA for Multi-Task Instruction Tuning

# Chunlei Xin<sup>1,3</sup>, Yaojie Lu<sup>1</sup>\*, Hongyu Lin<sup>1</sup>, Shuheng Zhou<sup>4</sup>, Huijia Zhu<sup>4</sup>, Weiqiang Wang<sup>4</sup>, Zhongyi Liu<sup>4</sup>, Xianpei Han<sup>1,2</sup>, Le Sun<sup>1,2</sup>

<sup>1</sup>Chinese Information Processing Laboratory <sup>2</sup>State Key Laboratory of Computer Science Institute of Software, Chinese Academy of Sciences, Beijing, China <sup>3</sup>University of Chinese Academy of Sciences, Beijing, China <sup>4</sup>Ant Group

{chunlei2021, luyaojie, hongyu, xianpei, sunle}@iscas.ac.cn {shuheng.zsh, huijia.zhj, weiqiang.wwq, zhongyi.lzy}@antgroup.com

#### **Abstract**

Low-Rank Adaptation (LoRA) is a widespread parameter-efficient fine-tuning algorithm for large-scale language models. It has been commonly accepted that LoRA mostly achieves promising results in single-task, low-resource settings, and struggles to handle multi-task instruction tuning scenarios. In this paper, we conduct a systematic study of LoRA on diverse tasks and rich resources with different learning capacities, examining its performance on seen tasks during training and its cross-task generalization on unseen tasks. Our findings challenge the prevalent assumption that the limited learning capacity will inevitably result in performance decline. In fact, our study reveals that when configured with an appropriate rank, LoRA can achieve remarkable performance in high-resource and multi-task scenarios, even comparable to that achieved through full fine-tuning. It turns out that the constrained learning capacity encourages LoRA to prioritize conforming to instruction requirements rather than memorizing specialized features of particular tasks or instances. This study reveals the underlying connection between learning capacity and generalization capabilities for robust parameter-efficient fine-tuning, highlighting a promising direction for the broader application of LoRA across various tasks and settings.

Keywords: Natural Language Generation, Low-Rank Adaptation, Instruction Tuning

#### 1. Introduction

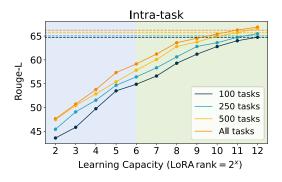
Instruction tuning aims to fine-tune language models on a collection of tasks described via instructions (Wei et al., 2022; Wang et al., 2022; Iyer et al., 2022; Wang et al., 2023; Longpre et al., 2023), which aligns the large-scale language models (LLMs) with human-like responses for the given instruction. Due to the extensive computational resources required to fully fine-tune LLMs, many studies have adopted Low-Rank Adaptation (LoRA, Hu et al. (2022)) methods for instruction tuning (Xu et al., 2023; Cui et al., 2023; Zhang et al., 2023; Wu et al., 2023). Specifically, LoRA introduces trainable rank decomposition matrices into each layer of the Transformer, and freezes the pre-trained model weights, which significantly reducing the number of trainable parameters for downstream tasks.

Due to the reduction of learnable parameters, one common assumption is that the limited learning capacity will lead to performance decline (Ding et al., 2023; Mundra et al., 2023; Zhang et al., 2023; Wu et al., 2023; Sun et al., 2023). Consequently, LoRA is primarily considered effective when data resources are limited and tuning tasks are constrained (Chen et al., 2022; Pu et al., 2023; Fu et al., 2023). However, the performance of LoRA with large learning capacities in scenarios with diverse

tasks and abundant resources remains unexplored. In this paper, we aim to bridge this research gap by conducting a systematic investigation of LoRA in instruction tuning scenarios. We evaluate the performance of LoRA across diverse tasks and rich resources with different learning capacities achieved by adjusting the number of trainable parameters. Specifically, we vary the trainable parameters of LoRA by altering its intrinsic ranks, and control task diversity by changing the number of learning tasks. The effectiveness is evaluated for both intra-task performance on learned tasks and inter-task performance on unseen tasks. The experimental results reveal the following findings:

- LoRA can achieve remarkable performance in high-resource, multi-task settings. As demonstrated in Figure 1, when configured with an appropriate rank, LoRA can achieve performance comparable to or even surpass those of full fine-tuning. This finding challenges the previous assumption that LoRA is primarily suited for low-resource and constrained task scenarios, broadening the applicability of LoRA.
- Besides parameter-efficiency, LoRA may further bring the benefit of implicit regularization.
   We observe that LoRA consistently exhibits improved performance as its learning capacity increases. Even with a high learning capac-

<sup>\*</sup> Corresponding author



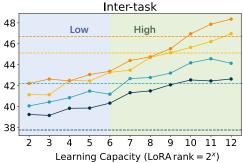


Figure 1: Scaling trends of intra-task (left) and inter-task (right) generalization capabilities of LLaMA-7B as a function of learning capacity across varying task diversity. Models fine-tuned using LoRA are represented by solid lines, while fully tuned models are denoted by dashed horizontal lines. Models are tuned on 100, 250, 500, and 756 tasks from Nlv2 dataset (Wang et al., 2022), respectively.

ity (rank $\geq 2^{10}$ ), LoRA maintains robust crosstask generalization performance compared to full fine-tuning. We assume that there exists implicit regularization within LoRA, which reduces overfitting to seen tasks and improves the generalization ability on unseen tasks.

· Only when LoRA attains sufficient learning capacity can it effectively grasp the semantics of the task instructions. As shown in Figure 1, when the learning capacity is limited, in contrast to the noticeable improvement in the tasksolving capability for seen tasks, increasing the intrinsic rank of LoRA does not significantly enhance performance on unseen tasks. We observe that the constrained learning capacity prompts LoRA to prioritize adhering to the output form specified in the instruction, but LoRA may struggle to accurately solve the task. Therefore, only when the learning capacity is sufficiently large can LoRA effectively understand the semantics of instructions, thereby enhancing the performance on unseen tasks.

In summary, our findings challenge the common assumption that limited learning capacity will result in a decline in performance. We discover that LoRA can achieve both promising intra- and inter-task generalization performance. It appears that with limited learning capacity, LoRA tends to prioritize generating outputs that align with the instruction requirements but may lack accuracy. This study enhances our understandings of the application scope and influence factors of LoRA, therefore raising the opportunity to employ LoRA across a wider range of tasks and settings.

#### 2. Background

#### 2.1. Instruction Tuning

**Instruction Tuning Problem.** Instruction tuning aims to align LLMs with human-like natural lan-

guage instructions. As illustrated in Figure 2, an instruction dataset can be formally defined as  $\mathcal{D}=\{(I^{(t)},x_i^{(t)},y_i^{(t)}),t\in T\},$  where t represents a specific task within the task set  $T,I^{(t)}$  denotes the natural language instruction for task  $t,x_i^{(t)}$  represents the input, and  $y_i^{(t)}$  represents the target associated with input  $x_i^{(t)}$  under task t. The primary objective of instruction tuning is to enable a LLM  $\mathcal M$  to generate the appropriate response given  $(I^{(t)},x_i^{(t)}),$  as depicted in the right part of Figure 2.

Instruction Tuning with LoRA. To achieve comparable performance to full-parameter fine-tuning while significantly reducing the computational requirements, one notable representative method is LoRA(Hu et al., 2022). Inspired by Aghajanyan et al. (2021), LoRA hypothesizes that the updates to the weights have a low "intrinsic rank" during adaptation. Parameter update for a pre-trained weight matrix  $W_0 \in \mathbb{R}^{d \times k}$  in LoRA is decomposed into a product of two low-rank matrices:

$$h = W_0 x + \Delta W x = W_0 x + BAx$$

where  $B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times k}$ , and the rank  $r \ll min(d,k)$ . During training,  $W_0$  is frozen and does not receive gradient updates, while A and B contain trainable parameters. A is initialized using a random Gaussian initialization technique, whereas matrix B is initialized as a zero matrix, so the initial value of  $\Delta W = BA$  is zero at the beginning of training. It's worth noting that  $\Delta Wx$  is scaled by  $\alpha/r$ , and a higher  $\alpha$  value assigns more weight to the LoRA activations.

#### 2.2. Experimental Settings

**Dataset.** We adopt the English portion of the SUPER-NATURALINSTRUCTION (NIv2) dataset (Wang et al., 2022) as our benchmark, which contains 757 training tasks and 119 unseen test tasks covering a diverse range of task types. The default instruction schema of NIv2 is composed of

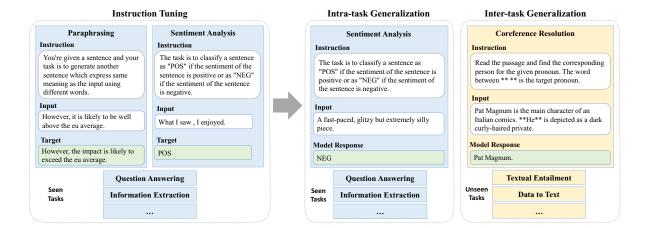


Figure 2: Overview of instruction tuning and model generalization. Instruction tuning aligns large language models with human-like natural language instructions across various tasks. After instruction tuning, the model is expected to adapt to various scenarios across different domains. In this study, we consider the model's ability to perform tasks included in the training stage as intra-task generalization, while its capability to handle tasks not encountered during training is considered as inter-task generalization.

the following components: task definition, positive examples and negative examples. In our experiments, we adopt task definitions alone as the task instructions, without any demonstration examples. Tuning on Varying Training Task. To evaluate the performance of LoRA with varying learning capacities across different levels of task diversity, we construct the training set  $T_{train}$ , which consists of four datasets  $T_{100}$ ,  $T_{250}$ ,  $T_{500}$  and  $T_{all}$ . Specifically, we sample 100/250/500/all tasks from the official training tasks in NIv2 and then select 100 examples from each of these tasks. Task sampling is conducted in proportion to the distribution of task categories within the NIv2 dataset. It's worth noting that the four training datasets are inclusive, meaning that each dataset contains all the training data from the preceding dataset. For instance,  $T_{250}$  includes all the training data present in  $T_{100}$ , and so forth. To ensure the robustness and reliability of our findings, we perform three rounds of sampling, resulting in three distinct training sets  $T_{\text{train}}$ .

Evaluation on Intra- and Inter-tasks. In our experimental setup, the evaluation task set is further divided into two distinct subsets:  $T_{\text{seen}}$  and  $T_{\text{unseen}}$ . The former,  $T_{\text{seen}}$ , includes the tasks that have been encountered during the instruction tuning stage, and is used to evaluate the intra-task performance of the instruction-tuned model  $\mathcal{M}'$ . Formally, the intra-task test set can be denoted as  $\mathcal{D}_{\text{test-seen}} =$  $\{(I^{(t)}, x_i^{(t)}, y_i^{(t)}), t \in T_{\text{seen}}\}$ . It evaluates the memorization capabilities of  $\mathcal{M}^{'}$ , measuring how effectively the model performs on the seen tasks during training. Conversely,  $T_{\text{unseen}}$  includes the tasks that were not encountered during training, serving to measure the cross-task generalization performance of  $\mathcal{M}^{'}$ . The inter-task test set can be denoted as  $\mathcal{D}_{\text{test-unseen}} = \{(I^{(t)}, x_i^{(t)}, y_i^{(t)}), t \in T_{\text{unseen}}\}.$  It measures the ability of  $\mathcal{M}'$  to generalize across different unseen tasks. Specifically, we utilize the official test split of Nlv2, which consists of 119 unseen test tasks, and select 100 balanced test samples for each task to construct the inter-task test set. To construct the intra-task test set, each task within  $T_{100}$  contributes 100 examples, resulting in a total of 10,000 examples. It is ensured that all examples used for testing do not overlap with the corresponding training set.

#### 2.3. Details about Model Fine-tuning

Considering the remarkable performance of LLaMA (Touvron et al., 2023a) on various English benchmarks and the widespread application of its LoRA version (Wu et al., 2023; Cui et al., 2023; Xu et al., 2023; Zhang et al., 2023), we utilize the LLaMA-7B to conduct our experiments. Specifically, we adopt the LoRA fine-tuning code on LLaMA, which is publicly available  $^1$ . We update four weight matrices ( $W_q$ ,  $W_v$ ,  $W_k$ , and  $W_o$ ) within the self-attention module, with varying ranks from 4 to 4096. The value of  $\alpha$  is set to twice the rank, and the dropout rate is consistently maintained at 0.05.

During instruction tuning process, we fine-tune LLaMA for 6 epochs, employing a learning rate of 5e-5 with a linear schedule for LoRA, and a learning rate of 3e-5 with a cosine schedule for full fine-tuning. In both cases, we choose AdamW (Kingma and Ba, 2015) as the optimizer, and set the training batch size to 128. For evaluation, following Wang et al. (2022), we employ both Exact Match accuracy and Rouge-L scores (Lin, 2004) to assess the performance. We report average scores across three

<sup>&</sup>lt;sup>1</sup>https://github.com/tloen/alpaca-lora

			$T_{100}$		$T_{250}$		$T_{500}$		$ T_{all} $	
	Rank	Trainable	EM	R-L	EM	R-L	EM	R-L	EM	R-L
LoRA	4	0.062%	29.86	43.59	30.97	45.43	32.70	47.45	32.56	47.63
	8	0.124%	31.28	45.84	33.91	49.07	35.01	50.41	35.11	50.71
	16	0.248%	34.37	49.75	35.93	51.53	37.13	52.86	37.46	53.77
	32	0.496%	37.81	53.49	38.30	54.66	39.21	55.39	40.99	57.32
	64	0.986%	39.05	54.87	40.35	56.42	41.72	57.83	43.42	59.14
LoRA	128	1.953%	40.94	56.60	42.48	58.33	44.23	60.07	45.73	61.21
	256	3.831%	43.70	59.29	45.17	60.62	47.51	62.80	48.44	63.60
	512	7.379%	45.80	61.19	47.76	62.81	48.82	63.75	49.92	64.52
	1024	13.745%	47.75	62.79	49.04	63.59	50.46	64.94	51.60	65.42
	2048	24.167%	49.46	64.01	50.40	64.73	51.78	65.70	52.66	66.26
	4096	38.927%	50.55	64.74	51.28	65.50	52.91	66.60	53.36	66.87
Full Fine-tuning			51.57	64.70	52.10	65.02	52.78	65.68	53.77	66.22

Table 1: Exact match (EM) and Rouge-L (R-L) scores on the intra-task test set of LoRA with varying ranks and Full Fine-tuning. The intra-task test set consists of examples from tasks included in  $T_{100}$  but not overlapping with the corresponding training set. All reported results represent the average performance across three distinct training sets.

			$T_{100}$		$T_{250}$		$T_{500}$		$ T_{all} $	
	Rank	Trainable	EM	R-L	EM	R-L	EM	R-L	EM	R-L
LoRA	4	0.062%	23.75	39.25	24.98	40.07	25.32	41.14	25.96	42.22
	8	0.124%	23.95	39.16	25.14	40.44	24.99	41.13	25.98	42.63
	16	0.248%	23.77	39.84	25.07	40.85	26.30	42.51	25.69	42.46
	32	0.496%	23.48	39.86	25.30	41.48	25.96	42.45	26.25	43.07
	64	0.986%	24.41	40.33	24.85	41.19	26.52	43.28	26.69	43.37
LoRA	128	1.953%	24.65	41.33	26.15	42.68	26.58	43.48	27.48	44.41
	256	3.831%	24.82	41.50	26.22	42.78	27.95	44.70	28.06	44.75
	512	7.379%	25.40	42.09	26.27	43.20	28.04	45.15	28.79	45.53
	1024	13.745%	25.73	42.55	27.22	44.19	28.45	45.66	29.74	46.96
	2048	24.167%	25.58	42.45	27.94	44.57	29.20	46.22	30.66	47.87
	4096	38.927%	26.38	42.64	27.76	44.14	30.05	46.97	31.12	48.36
Full Fine-tuning			23.77	37.77	26.74	42.21	28.85	45.12	30.58	46.71

Table 2: Exact match (EM) and Rouge-L (R-L) scores on the inter-task test set of LoRA with varying ranks and Full Fine-tuning. The inter-task test set comprises 119 unseen test tasks drawn from the official test split of NIv2. All results reported are the average performance across three distinct training sets.

samplings training sets to ensure the robustness and reliability of our results. All experiments are conducted on NVIDIA A100-80GB GPUs.

# 3. LoRA for Instruction Tuning

In this section, we systematically investigate the relationship between the learning capacity and generalization performance of LLMs. To achieve this, we vary the trainable parameters within LoRA and adjust the number of learning tasks during the instruction tuning process. Then we evaluate both intra-task performance on learned tasks and intertask performance on unseen tasks of instruction-tuned models. Table 1 and Table 2 present the

Exact match (EM) and Rouge-L (R-L) scores on the intra-task test set and inter-task test set, respectively. Furthermore, we analyze the differences in output patterns between LoRA and full fine-tuning on general tasks. Outputs of LoRA and fully fine-tuned models on representative tasks are shown in Figure 3. Exact match scores and the percentage of out-of-options (OOP) outputs on classification tasks within the inter-task test set are illustrated in Figure 4.

#### 3.1. LoRA can Beat Full Fine-tuning

Finding 1: LoRA can achieve remarkable intra-task and inter-task generalization performance in highresource, multi-task settings with an appropriate Instruction: Two analogies that relate items to the associated containers is given in the form "A:B.C:?". "A:B" relates item A to its associated container B. Your task is to replace the question mark (?) with the appropriate *container* for the given item C, following the "A:B" relation.

Input: diapers : box. carrots : ?
Golden Answer: bag

Full Fine-tuning: A: B
LoRA (rank = 16): bag
LoRA (rank = 2048): bag

Word Analogy

**Instruction:** [...]Given a prompt and two responses, identify which response is the type of "yes, and". <u>Indicate your answer by 'Response 1'</u> if the first response is correct, otherwise 'Response 2', [...]

**Input:** Prompt: You know this cockroach? Response 1: I thought he died, so maybe we see his ghost? I guess there's no point in speculating.

Response 2: Yeah, he was here when I founded the place.

Golden Answer: Response 2

Full Fine-tuning: No

incorrect answers are highlighted in red, while the correct part of the output is in blue.

LoRA (rank = 16): Response 1

LoRA (rank = 2048): Response 2

**Instruction:** The task is to write a full sentence or two using all of the given information. Use all of the information provided.

Input: name[xname], cuisine[Chinese],
rating[average], location[city centre]

**Golden Answer:** xname serves average Chinese food in the city centre.

Full Fine-tuning: name[xname], cuisine[Chinese], rating[average], location[city centre]

LoRA (rank = 16): (nothing)

LoRA (rank = 2048): The food at xname

was average Chinese.

Dialogue Act Recognition Data to Text

Figure 3: Case studies for instruction models trained on  $T_{\rm all}$  using Full Fine-tuning and LoRA. Descriptions of the task output form are <u>underscored</u>, and the specific output type is denoted in *italics*. The model's

rank.

Previous research in instruction tuning typically constrained the rank of LoRA within the range of 8 to 64 (Zhang et al., 2023; Wu et al., 2023; Xu et al., 2023; Cui et al., 2023). Meanwhile, one common perspective is that LoRA is primarily effective for low-resource and constrained task scenarios (Chen et al., 2022; Pu et al., 2023; Fu et al., 2023). However, our experimental results deviate from this viewpoint. Surprisingly, we observe that even when fine-tuned on NIv2 that consists of over 75,000 examples across 756 tasks, LoRA with an appropriate rank, such as 2048 and 4096, exhibits remarkable intra-task and inter-task generalization performance. This observation challenges the previous opinion that the effectiveness of LoRA is primarily limited to low-resource and limited tasks scenarios. thereby expanding the applicability of LoRA.

Finding 2: LoRA can achieve performance comparable to or even surpass those of full fine-tuning.

Due to the reduction in the number of learnable parameters, one common assumption is that the constrained learning capacity could lead to a decline in performance (Ding et al., 2023; Mundra et al., 2023; Zhang et al., 2023; Wu et al., 2023). In line with these studies, we observe that when configured with learnable parameters less than 1%, the intra-task generalization performance of LoRA exhibits a decline of near 10 points compared to full fine-tuning. This performance limitation can be attributed to the restricted learning capacity of LoRA, which hinders its ability to effectively recognize and grasp task-specific features.

However, our experimental results also demonstrate that LoRA can achieve comparable or even superior intra-task and inter-task generalization performance compared to full fine-tuning. As shown in Table 1, the generalization capability of LoRA

continues to improve with an increase in its learning capacity. Ultimately, when configured with a rank of 4096, LoRA achieves intra-task generalization performance comparable to that of full fine-tuning. Similarly, when evaluating the inter-task generalization ability, LoRA with a rank of 2048 surpasses full fine-tuning across different levels of task diversity, demonstrating improved Exact Match and Rouge-L scores. It's worth noting that while increasing the rank of LoRA results in a higher computational demand, it is still far less than that required for full fine-tuning. For example, when the rank of LoRA is set to 2048, the model can still be fine-tuned on a single A100-80GB GPU. Therefore, how to select the optimal rank based on task complexity and available resources deserves further investigation.

Moreover, we observe that, for some challenging unseen tasks, LoRA with an appropriate rank exhibits relatively better instruction understanding capabilities compared to full fine-tuning. For instance, as shown in Figure 3 for the Data to Text task, models are expected to generate complete sentences that incorporate all the provided input information. In practice, fully tuned models often duplicate the provided input as their output. However, LoRA with a rank of 2048 makes efforts to construct sentences based on the provided information, resulting in fluent sentences that include expected keywords such as "xname", "average" and "Chinese".

#### 3.2. LoRA is not only Parameter-Efficient

**Finding 1:** LoRA shows strong robustness in terms of generalization ability. As the learning capacity of the model grows, the generalization performance of LoRA is consistently blessed with an enhancement.

As presented in Table 1, when evaluated on the intra-task test set, LoRA consistently exhibits im-

proved performance as the learning capacity increases. Similarly, Table 2 demonstrates that increasing the number of trainable parameters within LoRA consistently enhances its inter-task generalization capability. Ultimately, with a rank of 4096, LoRA outperforms full fine-tuning on both learned tasks and unseen tasks.

Additionally, we observe that the enhanced learning capacity of LoRA usually leads to improved task-solving capabilities across diverse domains. In particular, in classification tasks with multiple options, LoRA with a relatively low rank tends to output the same option regardless of the input. However, when equipped with sufficient learning capacity, as shown in Figure 4, LoRA learns to generate more balanced outputs to improve the accuracy on classification tasks. Moreover, for some challenging generation tasks such as Data to Text in Figure 3, LoRA with limited learning capacity often produce nothing or fragmented sentences. As the intrinsic rank increases, LoRA exhibits enhanced tasksolving capabilities to generate fluent sentences covering more provided information.

Finding 2: LoRA exhibits an implicit regularization effect.

Fully fine-tuned models excels at solving tasks encountered during training, but their performance on unseen tasks, especially those significantly differing from training tasks, is suboptimal. In contrast, as we increase the rank of LoRA, we observe that there is no significant decline in generalization performance across different levels of task diversity. With the increase in learning capacity, LoRA demonstrates effective problem-solving abilities for both seen and unseen tasks. Notably, the generalization performance of LoRA with learnable parameters exceeding 20% remains robust against overfitting, even when the number of learning tasks is relatively limited.

We observe that when dealing with unseen tasks that markedly differ from those encountered during training, fully tuned models may generate inaccurate responses due to a lack of understanding of task instructions. In contrast, models fine-tuned using LoRA demonstrate relatively stronger instruction understanding capabilities and exhibit better generalization performance for such unseen tasks. An illustrative example is the Word Analogy task shown in Figure 3, where fully tuned models struggle to comprehend the instruction to produce words based on the provided set of phrases. Instead, they generate outputs like "A" or "B", or directly copy a word from the input. Conversely, LoRA demonstrates an ability to grasp the task instruction and attempts to generate appropriate words.

These observations imply the implicit regularization within LoRA, which reduces overfitting to the seen tasks and improves the generalization ability

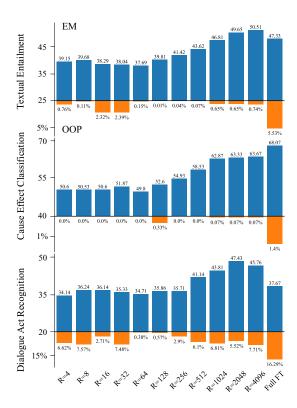


Figure 4: Exact Match (EM) scores and the percentage of out-of-options (OOP) outputs from LoRA with varying ranks and Full Fine-tuning on representative classification tasks within the inter-task test set. Models are fine-tuned on  $T_{\rm all}$  and all results reported are the average performance across three distinct training sets.

on unseen tasks. We hypothesize that the constrained learning capacity in LoRA might prompt it to prioritize capturing task-independent patterns and representations. Consequently, this may equip LoRA with enhanced instruction understanding and following capabilities, enabling it to generalize more effectively across diverse tasks.

# 3.3. Balancing Form and Semantic Accuracy in LoRA

**Finding 1:** Learning to conform to the task output format and learning to accurately solve the task are two tasks with different levels of complexity.

Even when configured with fewer than 1% of the learnable parameters, LoRA exhibits cross-task generalization performance comparable to full finetuning. We observe that this is largely attributed to the fact that fully fine-tuned models sometimes generate responses that are semantically correct but deviate from the prescribed format. In contrast, models fine-tuned using LoRA tend to generate outputs that align with the instruction requirements but may lack accuracy. This observation suggests that learning to conform to the task output format and

learning to accurately solve the task are two tasks with different levels of complexity. The high scores on unseen tasks achieved by LoRA may be largely attributed to the high formal accuracy of its outputs, and this does not necessarily indicate that LoRA has acquired the ability to effectively address the task. This trend is observable in Figure 4, where the percentage of out-of-options outputs from LoRA is significantly lower than that of full fine-tuning on most classification tasks within the inter-task test set. However, the tendency of full fine-tuning models to produce more out-of-option outputs cannot be taken as an indication of overfitting. As illustrated in Figure 4, there is still a noticeable performance gap between the Exact Match scores achieved by full fine-tuning and LoRA with limited learning capacity. This suggests that LoRA prioritizes adhering to the output form required by the instruction, but may struggle to solve the task accurately due to limited learning capacity.

An illustrative example is the Dialogue Act Recognition task shown in Figure 3. When the instruction requires the model to output "Response 1" or "Response 2" to select an appropriate response, fully tuned models often output "yes/no" or simply "1/2", indicating a tendency to provide answers that are not among the given options. Unlike full fine-tuning, LoRA even with a rank of 16 learns to follow the instruction to generate responses between "Response 1" and "Response 2". However, LoRA may often produce incorrect options due to its constrained learning capacity.

**Finding 2:** Only when LoRA attains sufficient learning capacity can it effectively grasp the semantics of the task instructions.

When the learning capacity is limited, in contrast to the noticeable improvement in the task-solving capability for seen tasks, increasing the intrinsic rank of LoRA does not significantly enhance performance on unseen tasks. We assume the primary reason for this phenomenon is that the constrained learning capacity causes LoRA to prioritize adhering to the output form required in the instruction, but it may struggle to accurately solve the task. Therefore, only when LoRA attains sufficient learning capacity can it effectively understand the semantics of the instructions, thereby enhancing the performance on unseen tasks.

This tendency is particularly evident in classification tasks, where LoRA can effectively follow the instructions to generate outputs from the provided options in most scenarios. However, when configured with constrained learning capacity, LoRA tends to produce the same option for different inputs within the same task. As shown in Figure 4, it's only when the learning capacity is sufficiently large that LoRA can effectively understand the semantics of instructions and options, achieving both formal

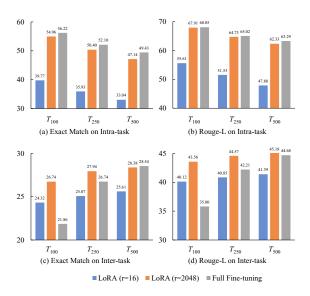


Figure 5: Exact Match and Rouge-L scores of LoRA and Full Fine-tuning on the intra-task test set (upper) and inter-task test set (lower). Models are fine-tuned with a consistent learning data size of 25,000 while varying task diversity.

accuracy and semantic correctness. Therefore, in Textual Entailment and Cause-Effect Classification tasks, the performance improvements of LoRA with increasing rank are not notable until the rank exceeds 128. Similarly, this trend continues until a rank of 512 in Dialogue Act Recognition.

These observations underscore the critical role of learning capacity settings in model training and compression, implying that large models may adopt different paradigms to solve problems at different learning capacities.

### 3.4. Task Scaling

To further investigate the effect of LoRA and full fine-tuning on different levels of task diversity, this section maintains a constant data size while scaling tasks, and subsequently measures their generalization performance.

Specifically, we design control experiments based on the previous experimental settings. To maintain a constant training data size while varying task diversity, we randomly sample 250 examples from each task in  $T_{100}$  and 50 examples from each task in  $T_{500}$  with replacement, resulting in training datasets  $T_{100}^*$  and  $T_{500}^*$  that contain a total of 25,000 instances. Similarly, these training data do not include any test data, and we conduct three samplings to ensure the reliability of the experiments. Figure 5 presents the intra-task and inter-task generalization performance for both full fine-tuning and LoRA, where we consider two representative rank settings: 16 and 2048. All reported results rep-

resent the average performance across the three sampled training sets.

By analyzing the intra-task and inter-task generalization performance of both LoRA and full fine-tuning, we find that:

**Finding 1**: The learning capacity can significantly impact the memorization capabilities of LLMs.

When the dataset size remains consistent while the number of learning tasks increases, having a sufficient learning capacity becomes essential for achieving effective performance on seen tasks. This is evident from the results presented in the upper part of Figure 5, which consistently demonstrate an improvement in intra-task generalization capability with increasing number of learnable parameters. Furthermore, as the complexity of learning objectives escalates, there is a consistent decline in the intra-task performance of models with the same learning capacity. In particular, as the number of learning tasks scales from 100 to 500, there is a noticeable reduction in generalization performance for both LoRA and full fine-tuning on seen tasks. This phenomenon can be attributed to the reduction in available examples for learning each specific task as the number of training tasks grows. Consequently, models with limited learning capacity face challenges in effectively capturing task-specific features and patterns, which subsequently impacts its intra-task generalization ability.

**Finding 2**: The relationship between learning capacity and task diversity has intricate effects on the generalization performance of LLMs.

From the bottom part of Figure 5, We observe that larger learning capacity doesn't necessarily lead to stronger generalization performance, which differs from the commonly held belief. Specifically, the inter-task generalization performance of full fine-tuning lags behind that of LoRA with a rank of 2048, which only possesses 24% of the learnable parameters of full fine-tuning. This phenomenon also highlights the significance of incorporating appropriate regularization techniques during the fine-tuning process to enhance the cross-task generalization capability of LLMs. Additionally, determining the optimal learning capacity represents a valuable direction for future research.

#### 4. Related Work

#### 4.1. Instruction Tuning

In recent years, large language models have demonstrated remarkable capabilities in zero-shot (Wei et al., 2022; Sanh et al., 2022; Kojima et al., 2022) and few-shot learning (Brown et al., 2020; Mishra et al., 2021; Ye et al., 2021). To help the model better understand user intentions and follow instructions more accurately, various efficient

instruction tuning strategies have been proposed (Wei et al., 2022; Wang et al., 2022; Iyer et al., 2022; Wang et al., 2023; Longpre et al., 2023; Touvron et al., 2023b).

Among them, extensive research has shown that scaling the number of learning tasks can significantly enhance the generalization ability of large language models. Specifically, Wei et al. (2022) reveal that the performance of LLMs on unseen tasks improves with the increasing number of instruction tuning task clusters, and that the benefits emerge only with sufficient model scale. Similarly, Sanh et al. (2022) have demonstrated that multi-task prompted training can enable strong zero-shot generalization abilities in language models and scaling the number of training tasks and the number of prompts per task helps boost the zero-shot task generalization performance. Chung et al. (2022) examine the effect of scaling and show that the model performance substantially improved with both a larger model size and more fine-tuning tasks.

# 4.2. Low-Rank Adaptation

Recently, due to the extensive computational resources required to fully fine-tune LLMs, researchers have focused on developing parameter-efficient techniques that strike a balance between computational resources and performance (Houlsby et al., 2019; Lester et al., 2021; Li and Liang, 2021; Ben Zaken et al., 2022; Liu et al., 2023). Among them, one notable representative method is LoRA (Hu et al., 2022), which proposes to decompose the original weight matrix updates into a product of two low-rank matrices.

However, recent works focused on LoRA typically conduct experiments using ranks within the range of 8 to 64. Among them, some works indicate that LoRA generally achieves promising performances in single-task, low-resource settings. For example, Chen et al. (2022) highlight that LoRA tends to outperform full fine-tuning solely on low-resource tasks. Pu et al. (2023) reveal that LoRA outperforms full fine-tuning in low to medium-resource scenarios, while the effectiveness of full fine-tuning tends to improve with increased data availability. Moreover, certain studies indicate that LoRA struggles to achieve comparative performance to full fine-tuning in multi-task instruction tuning scenarios. Zhang et al. (2023) utilize LoRA with a rank of 16 to fine-tune LLaMA through scenario-specific multi-task instruction tuning, uncovering that full fine-tuning achieves superior performance compared to LoRA. Sun et al. (2023) fine-tune LLaMA on Chinese instruction data and show that the performance of LoRA with a rank of 8 lags significantly behind full fine-tuning by approximately 10 points. Wu et al. (2023) show a performance gap of over 10 points between full fine-tuning and LoRA with

a rank of 8 when fine-tuning LLaMA on medical papers.

To the best of our knowledge, the exploration of the relationship between trainable parameters, task diversity, and multi-task learning capabilities for LoRA remains limited. To bridge this research gap, we conduct a comprehensive analysis to explore the intricate connection between the learning capacity of LoRA and its generalization performance on both seen and unseen tasks.

#### 5. Conclusion

In this paper, we conduct a systematic study to examine the intra-task and inter-task generalization capabilities exhibited by LoRA across various ranks and full fine-tuning. Our findings challenge common assumptions that the limited learning capacity will result in performance decline. Specifically, LoRA can achieve remarkable intra-task and inter-task performances in high-resource and multitask scenarios, even comparable to that of full finetuning. We observe that the limitation in learning capacity prompts LoRA to prioritize conforming to the instruction requirements instead of memorizing the specialized features of particular tasks or instances. These observations point to implicit regularization within LoRA, which reduces overfitting to seen tasks and improves the generalization ability on unseen tasks. Our research also uncovers the inherent connection between learning capacity and generalization capabilities, contributing to robust and parameter-efficient fine-tuning.

#### 6. Limitations

We discuss limitations of our work that hopefully could inspire future research in this avenue.

Limited Exploration of Low-Rank Adaptation: Our study primarily investigates the generalization capabilities of models with different learning capacities by varying the rank of LoRA within the attention layers of transformers. Although this aligns with prior research that has predominantly applied LoRA within the attention layers, we leave the exploration of extending the application of low-rank adaptation to the entire transformer architecture as important future work.

Limited Exploration of Base Model: Due to constraints in computational resources and time, our experiments are conducted exclusively on the LLaMA-7B. While this model serves as an illustrative example, it may not fully represent the broader landscape of large language models. Extending our investigation to a wider range of models with varying scales and architectures could provide a more comprehensive understanding of the connection between learning capacity and generalization

performance.

# 7. Acknowledgments

We sincerely thank the reviewers for their insightful comments and valuable suggestions. This work was supported by the Natural Science Foundation of China (No. 62122077, 62106251 and 62306303) and Ant Group Research Fund.

# 8. Bibliographical References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 7319–7328, Online. Association for Computational Linguistics.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient finetuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In Advances in Neural Information Processing Systems, volume 33, pages 1877–1901. Curran Associates, Inc.

Guanzheng Chen, Fangyu Liu, Zaiqiao Meng, and Shangsong Liang. 2022. Revisiting parameter-efficient tuning: Are we really there yet? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2612–2626, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma,

- Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416.
- Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for chinese llama and alpaca. *ArXiv*, abs/2304.08177.
- Ning Ding, Yujia Qin, Guang Yang, Fu Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Haitao Zheng, Jianfei Chen, Y. Liu, Jie Tang, Juanzi Li, and Maosong Sun. 2023. Parameter-efficient finetuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5:220–235.
- Zihao Fu, Haoran Yang, Anthony Man-Cho So, Wai Lam, Lidong Bing, and Nigel Collier. 2023. On the effectiveness of parameter-efficient fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12799–12807.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.
- Srinivas Iyer, Xiaojuan Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, Xian Li, Brian O'Horo, Gabriel Pereyra, Jeff Wang, Christopher Dewan, Asli Celikyilmaz, Luke Zettlemoyer, and Veselin Stoyanov. 2022. Opt-iml: Scaling language model instruction meta learning through the lens of generalization. *ArXiv*, abs/2212.12017.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

- Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 4582–4597, Online. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. Gpt understands, too. *Al Open*.
- S. Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*.
- Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Natural instructions: Benchmarking generalization to new tasks from natural language instructions. *ArXiv*, abs/2104.08773.
- Nandini Mundra, Sumanth Doddapaneni, Raj Dabre, Anoop Kunchukuttan, Ratish Puduppully, and Mitesh M Khapra. 2023. A comprehensive analysis of adapter efficiency. In Workshop on Efficient Systems for Foundation Models @ ICML2023.
- George Pu, Anirudh Jain, Jihan Yin, and Russell Kaplan. 2023. Empirical analysis of the strengths and weaknesses of PEFT techniques for LLMs. In ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models.
- Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey,

M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Xianghui Sun, Yunjie Ji, Baochang Ma, and Xiangang Li. 2023. A comparative study between full-parameter and lora-based fine-tuning on chinese instruction data for instruction following large language model. *ArXiv*, abs/2304.08109.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao. Vedanui Goswami. Naman Goval. Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. ArXiv, abs/2307.09288.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting* 

of the Association for Computational Linguistics (Volume 1: Long Papers), pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen, 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 5085-5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. Pmc-llama: Towards building open-source language models for medicine. *ArXiv*, abs/2304.14454.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. *ArXiv*, abs/2304.01196.

Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. 2021. CrossFit: A few-shot learning challenge for cross-task generalization in NLP. In *Proceedings of the* 2021 Conference on Empirical Methods in Natural Language Processing, pages 7163–7189, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yue Zhang, Leyang Cui, Deng Cai, Xinting Huang, Tao Fang, and Wei Bi. 2023. Multi-task instruction tuning of llama for specific scenarios: A preliminary study on writing assistance. *ArXiv*, abs/2305.13225.