Enhancing Large Language Model Performance in Downstream Tasks: A Balanced Approach with PEFT and RLHF

Xiwei Xu[1], Bangcheng Sun[1], Hongbo Zhao[1], Chenyu Zhou[1], Qiwen Wang[1]

[1]AI Class, Artificial Intelligence Research Institute, Xiamen University
{36920231153251,36920231153231,36920231153265,36920231153268,36920231153235}@stu.xmu.edu.cn

Abstract

This paper addresses the issues of excessive HBM usage and low computational efficiency during the RLHF (Reinforcement Learning from Human Feedback) process of existing large models. It explores how the integration of Parameter-Efficient Fine-Tuning (PEFT) and RLHF can find a balance between resource efficiency and performance, thereby significantly reducing the resource consumption during the RLHF process of existing large models. Initially, the paper discusses the importance of fine-tuning smaller large models for specific applications, emphasizing the balance between performance and resource efficiency. The paper then delves deeper into the mechanisms of PEFT and RLHF, exploring how these methods can be synergistically applied. Finally, it uses the adaptation for Chinese semantic analysis tasks as a case study. Extensive experiments were conducted, demonstrating the effectiveness of this approach, showing that PEFT and RLHF can be efficiently combined in specific downstream tasks. Our research indicates that the strategic application of PEFT and RLHF offers a feasible pathway to optimize smaller large language models for specific downstream tasks, achieving a balance between performance and computational practicality.

1 Introduction

In the rapidly evolving field of artificial intelligence, large language models (LLMs) have emerged as pivotal tools in understanding and processing human language.The advent of language models such as ChatGPT and GPT-4, which exhibit human-like understanding and generation capabilities across various domains, has highlighted the importance of instruction tuning in enabling these models to better comprehend human instructions.Over the past few years, there has been a significant increase in the size of pre-trained language models (PLMs) such as GPT3(Brown et al. 2020),OPT(Zhang et al. 2022), BLOOM(Workshop et al. 2022), and PaLM(Chowdhery et al. 2023), which have billions of parameters. This increase in size has been accompanied by a commensurate increase in the cost of training and deploying large PLMs, with substantial financial and environmental implications.

In the rapidly evolving landscape of artificial intelligence, large language models (LLMs) such as GPT-4 have emerged as groundbreaking tools, demonstrating remarkable proficiency in understanding and generating human language. These models, built upon deep learning algorithms and trained on extensive datasets, have shown great promise in a range of applications, from composing text to providing sophisticated customer service solutions.(Brown et al. 2020) However, the journey from a general-purpose LLM to a model adept at handling specific downstream tasks is fraught with challenges that are as diverse as they are complex.

From a technical perspective, Reinforcement Learning from Human Feedback (RLHF) can significantly enhance the capabilities of a model, but the RLHF process itself requires a substantial amount of VRAM. This necessitates the acquisition of professional-grade computing cards, which can be costly. Additionally, the RLHF process is a delicate task. It requires a deep understanding of the model's learning mechanics, as improper fine-tuning can lead to model degradation rather than improvement. Ensuring that the model achieves high accuracy in specific, often nuanced tasks is a challenge. This is complicated by the need to ensure that the model operates ethically, without bias, and in compliance with existing regulations—requirements that are crucial in sensitive domains such as healthcare or finance. Such processes are often realized through RLHF.

Parameter-Efficient Fine-Tuning (PEFT) is designed to facilitate the efficient adaptation of large pre-trained models for various downstream applications without the need to fine-tune all parameters. The PEFT approach selectively fine-tunes a small number of model parameters while freezing most of the pre-trained LLM's parameters, thereby significantly reducing computational and storage costs. Houlsby et al. (2019) proposed the fine-tuning method for BERT, marking the beginning of research in fine-tuning. This efficient method of PEFT makes fine-tuning on large language models more feasible, aiding in quicker adaptation to specific semantic analysis tasks. (Houlsby et al. 2019) first proposed an efficient fine-tuning method for BERT marked the beginning of research in efficient fine-tuning. The Parameter-Efficient Fine-Tuning (PEFT) approach makes the fine-tuning of large language models more feasible, aiding in quicker adaptation to specific downstream tasks. Reinforcement Learning from Human Feedback(RLHF) addresses the challenge of limited labeled data by incorporating reinforcement learning techniques that leverage human-generated feedback. Instead of relying solely on traditional supervised fine-tuning, RLHF integrates a reward model derived from human-provided feedback to guide the model's learning process. This approach is particularly advantageous in scenarios where acquiring large-scale labeled datasets is impractical or expensive. For instance, (Ouyang et al. 2022) use RLHF to fine-tune GPT3(Brown et al. 2020).

This paper adopts a combination of PEFT and RLHF, specifically through the combination of the PEFT module with the base model, to create various models that significantly reduce the VRAM usage during the RLHF process. To validate the effectiveness of our approach, we conduct the RLHF process on models for Chinese semantic analysis. Due to complex syntax and rich morphology, Chinese language processing presents unique challenges(Chen 2022). These characteristics make semantic analysis of the Chinese language a particularly complex task for language models(Shancheng, Yunyue, and Fuyu 2018).

We first introduce specific datasets related to Chinese semantic analysis, such as tasks involving sentence semantic analysis, sentence implication relationship analysis, and news headline classification. Then, this section details how PEFT is effectively used to perform RLHF on the model's parameter set, enhancing its ability to handle tasks related to Chinese semantic analysis.

Additionally, we explore the role of RLHF in improving the model's performance in Chinese semantic analysis tasks. By integrating feedback from our specially constructed datasets, the reward model learns specific language discrimination capabilities, thereby improving the trained large model's ability to accurately understand and interpret Chinese text during the reinforcement learning process.

Furthermore, this section presents case studies and experimental results demonstrating the efficacy of combining PEFT and RLHF in Chinese semantic analysis. These results highlight significant improvements in tasks such as sentence semantic analysis, analysis of implied relationships in sentences, and news headline classification.

In summary, this paper employs a combination of PEFT and RLHF, significantly reducing resource requirements, and tests the approach on target tasks to seek practical application in real-world scenarios.

2 Related Work

2.1 PEFT

The landscape of Parameter-Efficient Fine-Tuning (PEFT) has been enriched by a multitude of innovative methodologies aimed at reducing the computational and memory burdens associated with fine-tuning large pre-trained models. This section encapsulates some seminal and recent works that have significantly contributed to the domain.

LoRA *LoRA*(Hu et al. 2021) proposed a simple way to perform low-rank fine-tuning. As the new model contains the same size of parameters as the original model, it was a challenge for the models to balance the efficiency with model quality. To deal with this trouble, parameter update for a weight matrix in LoRA is decomposed into a production with two low-rank matrix.That is to say, for the model weight W, it no longer carry out full-parameter fine-tuning training, but add residual form to the weight, and complete the optimization process by training δW. LoRA approximate δW with matrices W_A and W_B

$$W' = W + \delta W \tag{1}$$

$$\delta W = W_A \cdot W_B \tag{2}$$

$$W_A \in \mathbb{R}^{in\times r}, \quad W_B \in \mathbb{R}^{r\times \text{out}} \tag{3}$$

All pre-trained parameters are frozen, and two sparse matrices W_A,W_B was introduced to be trained,gaussian initialization for W_A and zero initialization for W_B.The scaling factor is constant and typically equals $\frac{\alpha}{r}$,α serve as a hyperparameter. They can be integrated into the original weight matrix W by adding the projection of W_A and W_B to W.The overall architecture of LoRA is shown in Figure 1.

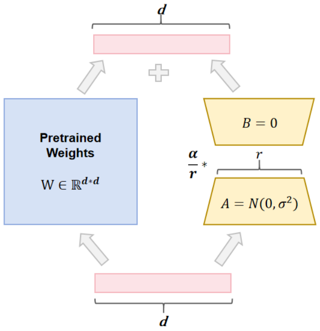


Figure 1: The overall architecture of LoRA.

RepAdapter (Luo et al. 2023) observe that most existing Parameter-efficient transfer learning(PETL) methods can inevitably slow down model inference. For prompt-tuning methods, the inserted tokens greatly increase the computation cost of vision models.In terms of visual adapters, the modules they add also increase the network complexity. So they proposed a parameter-efficient and computationally friendly adapter for giant vision models, called RepAdapter. Specifically, they prove that the adaption modules, even with a complex structure, can be seamlessly integrated into most giant vision models via structural re-parameterization. This property makes RepAdapter zero-cost during inference. The structure of RepAdapter is still different from existing visual adapters, as shown in Fig. 3. They found no performance degradation in the vision model after removing the non-linearity of the adapter.Specifically, the formulation of $f(X;\theta)$ for RepAdapter can be re-written as

$$f(X;\theta) = X + \phi_u(\phi_d(X)). \tag{4}$$

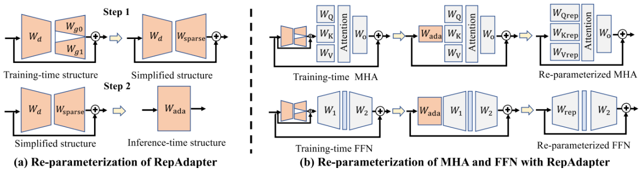
RepAdapter adopts the dense-sparse connections, where ϕ_u is formulated as a group-wise transformation by

$$\phi_u(X) = \left[X'_{g0}W_{g0}, \ldots, X'_{gk}W_{gk}\right] + b. \tag{5}$$

Here, $X'_i \in \mathbb{R}^{n\times\frac{c}{k}}$ is the features splitted from $X \in \mathbb{R}^{n\times c}$, k is the number of groups.$W_i \in \mathbb{R}^{\frac{c}{k}\times\frac{d}{k}}$ is the projection weight matrix and $b \in \mathbb{R}^d$ is the bias term.

能力点	问题	能力点的介绍
信息提取	以下句子中谁去了商店？“昨天，小明和小红一起去了商店买东西。”	从给定文本中识别并提取关键信息的能力，如从叙述中提取特定人物或行动。
信息分析	阅读以下描述并回答问题：“根据最新的统计数据，中国的人口总数已经超过了14亿。”问题：根据描述，中国的人口总数是多少?	理解和分析文本内容以回答特定问题的能力，要求模型能识别文本中的信息并进行逻辑分析。
创意生成	创造一个关于时间旅行的独特故事情节。	使用创造力构思新颖概念或故事情节的能力，要求模型具备创新性和吸引人的内容创造能力。
常识推理	如果一家人围在餐桌旁，摆放了许多菜肴和饮料，你可以推断他们在做什么?	使用日常常识进行逻辑推理的能力，这依赖于模型对常见生活情境的理解。
情景适应	在餐厅用餐时，服务员送来了错误的菜品，你应该如何向服务员表达你的抱歉和更换菜品的要求?	根据具体情境提供适当反应或建议的能力，要求模型对不同社交和环境情境有所理解，并能提出适当建议。

Table 1: Capabilities and Data Introduction



(a) Re-parameterization of RepAdapter

(b) Re-parameterization of MHA and FFN with RepAdapter

Figure 2: Illustration the structural re-prameterization of RepAdapter.

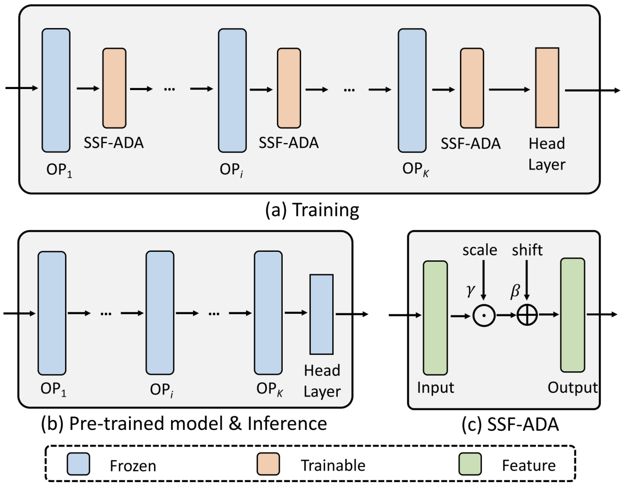
SSF SSF(Lian et al. 2022) is an efficient parameter fine-tuning method that fine-tunes a model by scaling and panning the depth features extracted from a pre-trained model, thus achieving comparable performance to full fine-tuning while requiring fewer tunable parameters. Scaling features is a fundamental step in ensuring that variables share similar scales, preventing one feature from dominating others during the training process. A common scaling method is Min-Max Scaling:

$$X_{\text{scaled}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

This formula normalizes features to a range between 0 and 1, maintaining the relative relationships between data points. Shifting features involves centering them around a common point, often referred to as zero-centering or mean shifting:

$$X_{\text{centered}} = X - \mu$$

By subtracting the mean (μ), this ensures that the data is centered around zero, offering advantages in terms of model interpretability and convergence. Combining scaling and shifting of features creates a robust preprocessing strategy, serving as a new baseline for efficient model tuning. Properly scaled and centered features contribute to quicker convergence during training and reduce sensitivity to hyperparameter choices.



(a) Training

(b) Pre-trained model & Inference

(c) SSF-ADA

Figure 3: SSF Working Principle

2.2 Dataset

analyObjective Dataset In the landscape of downstream tasks for large language models, a collection of diverse datasets is indispensable for benchmarking models' seman-

Dataset	Sentence(s)	Label
EPRSTMT	”外包装上有点磨损，试听后感觉不错”	Positive
	”产品完全不符合描述，非常失望”	Negative
TNEWS	”江疏影甜甜圈自拍，迷之角度竟这么好看， 美吸引一切事物”	News
	”最新研究显示咖啡有助于延缓老化”	Science
OCNLI	”sentence1”:”身上裹一件工厂发的棉大衣,手插在袖筒里” ”sentence2”:”身上至少一件衣服”	Entailment
	”sentence1”:”他出生在北京” ”sentence2”:”他是北京人”	Neutral
BUSTM	”sentence1”:”女孩子到底是不是你” ”sentence2”:”你不是女孩子吗”	1
	”sentence1”:”天气预报说今天会下雨” ”sentence2”:”今天应该会很晴朗”	0

Table 2: Expanded sample entries from each dataset

tic analysis capabilities. We incorporate four distinct Chinese datasets to evaluate performance across various domains: EPRSTMT, TNEWS, OCNLI, BUSTM.

EPRSTMT is structured for sentiment classification in e-commerce product reviews. It measures whether user-generated content reflects positive or negative sentiments. Extracted from Toutiao's news portal, TNEWS categorizes short text news titles into 15 sections, ranging from tourism to finance. As part of the Chinese Language Understanding Evaluation benchmark, OCNLI is the first large-scale dataset for native Chinese natural language inference. Stemming from an AI assistant, BUSTM's goal is to identify the semantic congruence of dialogue text pairs, a key task in intent recognition(Examples are shown in Table 1).

These datasets enable comprehensive assessments, helping discern the proficiency of language models in intricate semantic-related tasks.

2.3 Overview of RLHF

RM and PPO Reinforcement Learning From human Feedback (RLHF)(Ouyang et al. 2022),including reinforcement learning from human preferences, is a technique that trains a Reward Model(RM) directly from human feedback and uses the model as a reward function to optimize an agent's policy using reinforcement learning through an optimization algorithm like Proximal Policy Optimization(PPO).RLHF can improve the robustness and exploration of reinforcement-learning agents,especially when the reward function is sparse or noisy.

The Reward Model(RM) is trained in advance to the policy being optimized to predict if a given output is good (high reward) or bad(low reward).

Proximal Policy Optimization(PPO) is an algorithm in the field of reinforcement learning that trains a computer agent's decision function to accomplish difficult tasks, which is an architecture that improves our agent's training stability by avoiding too large policy updates. For two reasons:

1.Smaller policy updates during training are more likely to converge to an optimal solution.

2.A too big step in a policy update can result in falling “off the cliff” (getting a bad policy) and having a long time or even no possibility to recover.

To do that, it use a ratio that will indicates the difference between current and old policy and clip this ratio to a specific range $[1 - \epsilon, 1 + \epsilon]$, meaning that it remove the incentive for the current policy to go too far from the old one (hence the proximal policy term).PPO was developed by John Schulman in 2017(Schulman et al. 2017), and has become the default reinforcement learning algorithm at American artificial intelligence company OpenAI.

Llama2 and InstructGPT Llama2(Touvron et al. 2023) and InstructGPT(Ouyang et al. 2022) employ distinct methodologies for reward model training, diverging in their strategies to enhance model performance.Their process diagrams are illustrated in Fig. 4 and 5, respectively.

(Touvron et al. 2023) train two separate reward models(RM), one optimized for helpfulness (referred to as Helpfulness RM) and another for safety (Safety RM). They initialize their reward models from pretrained chat model checkpoints, as it ensures that both models benefit from knowledge acquired in pretraining. To train the reward model, they convert their collected pairwise human preference data into a binary ranking label format (i.e., chosen and rejected) and enforce the chosen response to have a higher score than its counterpart. They used a binary ranking loss:

$$\mathcal{L}_{ranking} = -\log\left(\sigma\left(r_\theta\left(x, y_c\right) - r_\theta\left(x, y_r\right)\right)\right) \qquad (6)$$

where $r_\theta\left(x, y\right)$ is the scalar score output for prompt x and completion y with model weights θ. y_c is the preferred response that annotators choose and y_r is the rejected counterpart.Built on top of this binary ranking loss, they further add a margin component in the loss:

$$\mathcal{L}_{ranking} = -\log\left(\sigma\left(r_\theta\left(x, y_c\right) - r_\theta\left(x, y_r\right) - m\left(r\right)\right)\right) \tag{7}$$

where the margin $m\left(r\right)$ is a discrete function of the preference rating.

In contrast, Starting from the SFT model with the final unembedding layer removed, (Ouyang et al. 2022) trained a model to take in a prompt and response, and output a scalar reward.Specifically, the loss function for the reward model is:

$$loss\left(\theta\right) = -\frac{1}{\binom{K}{2}} E_{\left(x, y_w, y_l\right) \sim D}\left[\log\left(\sigma\left(r_\theta\left(x, y_w\right) - r_\theta\left(x, y_l\right)\right)\right)\right] \tag{8}$$

, where $r_\theta\left(x, y\right)$ is the scalar output of the reward model for prompt x and completion y with parametersθ, y_w is the preferred completion out of the pair of y_w and y_l, and D is the dataset of human comparisons.Finally, since the RM loss is invariant to shifts in reward, we normalize the reward model using a bias so that the labeler demonstrations achieve a mean score of 0 before doing reinforcement learning(RL).

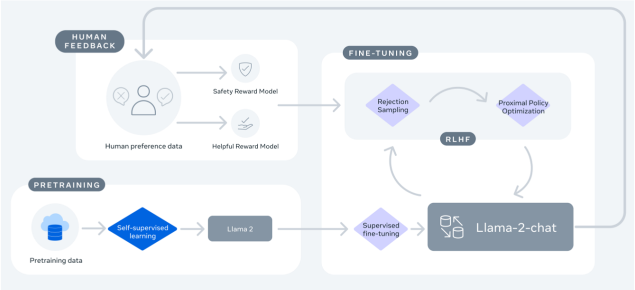


Figure 4: Training of Llama 2-Chat

The substantial resource overhead associated with Reinforcement Learning from Human Feedback (RLHF) can be mitigated by employing Parameter-Efficient Fine-Tuning (PEFT). In the following sections, we will elaborate in detail on this approach.

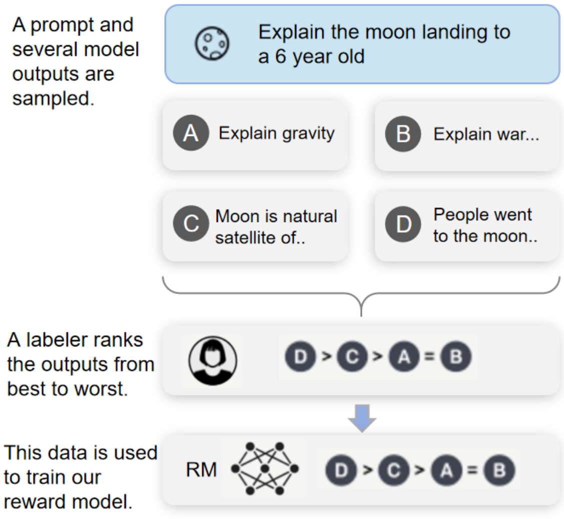

A prompt and several model outputs are sampled.
Explain the moon landing to a 6 year old
A Explain gravity
B Explain war...
C Moon is natural satellite of..
D People went to the moon..
A labeler ranks the outputs from best to worst.
D > C > A = B
This data is used to train our reward model.
RM
D > C > A = B

Figure 5: Training of InstructGPT reward model

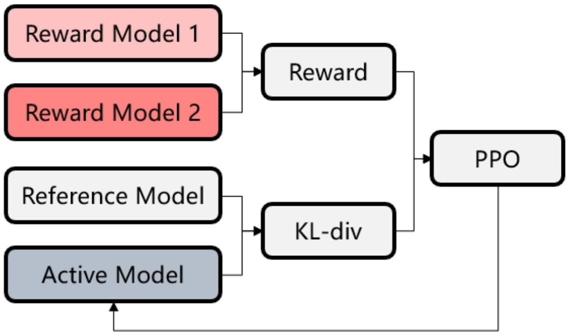


Figure 6: The schematic diagram of the models that need to be stored in VRAM during the RLHF process in Llama2.

3 Methodology

In this section, we discuss how we reduce the resource overhead of Reinforcement Learning from Human Feedback (RLHF) using Parameter-Efficient Fine-Tuning (PEFT)3.1. We will first introduce the PEFT method we employ, followed by a description of the RLHF steps3.3, and how we integrate both approaches.

3.1 PEFT Method

LoRA Given the empirical advantage of LoRA, note that the low-rank structure not only lowers the hardware barrier to entry which allows us to run multiple experiments in parallel, but also gives better interpretability of how the update weights are correlated with the pre-trained weights.

In principle, we can apply LoRA to any subset of weight matrices in a neural network to reduce the number of trainable parameters. We limit our study to only adapting the attention weights for downstream tasks and freeze the MLP modules (so they are not trained in downstream tasks) both for simplicity and parameter-efficiency.We determine which

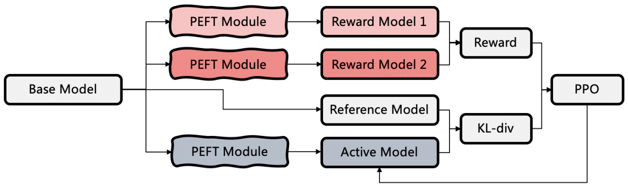


Figure 7: The schematic diagram of integrating PEFT within the RLHF process.

layers need to be fine-tuned using LoRA and replace it with the LoRA layer. The LoRA layer actually adds a bypass on the basis of the original layer, simulates the update of the parameters through low-rank decomposition, freezes the original parameters, and makes fine tuning to freeze the original parameters and only update the parameters of the LoRA layer.

SSF In our study, we have implemented the SSF (Scaling and Shifting Features) method, an efficient parameter fine-tuning approach introduced by Lian et al. in their paper ”Scaling & Shifting Your Features: A New Baseline for Efficient Model Tuning”(Lian et al. 2022). This method fine-tunes a model by scaling and shifting the depth features extracted from a pre-trained model, thereby achieving performance comparable to full fine-tuning with fewer tunable parameters.

RepAdapter Firstly, there are no activation functions in RepAdapter, and this is a critical condition for re-parameterization. Secondly, RepAdapter applies a dense-to-sparse structure. Its sparse part is constructed by group-wise transformation. Thirdly, RepAdapter adopts the pre-inserted placement to maximize its benefit to vision models.

All existing adapters follow the non-linear structure to adapt downstream tasks. However, (Luo et al. 2023) find no performance degradation in the vision model after removing the non-linearity of the adapter. In this case, the effects of non-linearity will become not obvious in vision models. Meanwhile, removing non-linearity also offers the distinct advantage of allowing the adapter to be re-parameterized after training.

In Eq. 4, ϕ_u and ϕ_d are usually two fully connected layers in existing adapters. In contrast, RepAdapter adopts the dense-sparse connections. The sparse structure of RepAdapter makes it more lightweight than conventional visual adapters(Chen et al. 2022). In practice, they also find that this sparse structure can improve performance, which prevents the model from overfitting on downstream tasks with limited training samples.

Empirically, they find that deploying RepAdapter before the neural modules can lead to better performance, which is also feasible for reparameterization. Meanwhile, they also observe that it is more beneficial to apply RepAdapter to both MHA and FFN in ViT. Therefore, the deployment of RepAdapter in Transformer can be formulated by

$$X_l^{'} = MHA\left(f\left(LN\left(X_{l-1}\right);\theta\right)\right) + X_{l-1} \tag{9}$$

$$X_l = FFN\left(f\left(LN\left(X_l^{'}\right);\theta\right)\right) + X_l^{'} \tag{10}$$

3.2 The Efficient Combination of PEFT and RLHF

Efficient PEFT Method LoRA, SSF, and RepAdapter, three Parameter-Efficient Fine-Tuning (PEFT) methods, each have their unique characteristics. LoRA’s low-rank structure reduces the required VRAM usage and accelerates training speed. SSF applies scaling and shifting only to each feature. RepAdapter, on the other hand, utilizes re-parameterizable Adapters, allowing inference processes to occur without additional computational overhead through re-parameterization. As these PEFT methods involve fewer trainable parameters, they enable us to reduce VRAM requirements for gradients and optimizer variables in supervised fine-tuning, while still delivering good performance. The accuracy of most of these methods is comparable to full-model fine-tuning, and in some datasets and tasks, they even surpass full-model tuning. This makes these PEFT methods particularly useful in efficient supervised fine-tuning.

Combination of PEFT and RLHF On the other hand, the re-parameterizable nature of these PEFT methods allows for the transformation of PEFT parameters and the base model into various models, and this transformation is reversible. This is especially useful for our RLHF process, as it allows for significant memory reduction through the combination of the base model and PEFT modules.

3.3 Experimental Procedure

RM model training We employed the hh_rlhf_cn dataset for pre-training the Linksoul-llama2-7b model, resulting in the pre-trained reinforcement model, llama2_RM_base. Subsequent steps involved processing the educhat dataset, an open-question dataset comprising various high school examination questions. We specifically filtered out the multiple-choice questions from this dataset. For the purpose of training the Reinforcement Model (RM), each multiple-choice question was transformed into three different formats based on the following structures: {“chosen”: correct option with explanation, “reject”: correct option only}, {“chosen”: correct option with explanation, “reject”: incorrect option with explanation}, and {“chosen”: correct option, “reject”: incorrect option with explanation}. The rationale behind this data construction was to train the model not only to provide the correct answer to open-ended questions but also to furnish reliable explanations. Fine-tuning of the llama2_RM_base model was conducted using this dataset to derive the final RM model, llama2_RM.

PPO training In Section 3.1, we employed three different Parameter-Efficient Fine-Tuning (PEFT) methods for training on four objective datasets: EPRSTMT, TNEWS, OCNLI, and BUSTM. The detailed results of this training are presented in Table 4.

	RepAdapter_1gpu	**SSF_1gpu**	**LoRA_1gpu**	**Full-Parameter_4gpu**
Trainable Parameter Quantity	14804736	2277376	5799936	6753220352
Trainable Parameter Ratio	0.2%	0.03%	0.08%	100%
Training Time	5h	3.5h	3.5h	10h

Table 3: A Comparison of PEFT and Full-Parameter methods in terms of the Number of Model Parameters and Training Time.

	EPRSTMT	**TNEWS**	**OCNLI**	**BUSTM**	**Average**
LoRA	90.00%	52.77%	51.15%	68.22%	65.54%
SSF	**90.49**%	50.70%	49.40%	64.00%	63.65%
RepAdapter	89.51%	**53.23**%	50.44%	69.58%	65.69%
Full Fine-Tuning	90.00%	52.40%	**51.20**%	**72.60**%	66.55%
base	88.36%	29.80%	44.37%	66.48%	57.25%

Table 4: Comparison of different methods on objective questions.

	EPRSTMT	**TNEWS**	**OCNLI**	**BUSTM**	**Average**
linksoul_llama	88.36%	29.80%	44.37%	66.48%	57.25%
Llama_2_70B_chat	89.70%	45.00%	50.60%	63.00%	62.08%
linksoul_SFT	**90.49%**	52.19%	51.11%	66.70%	65.12%
linksoul_RLHF	90.16%	**52.94%**	**51.94%**	**73.87%**	**67.23%**
ChatGLM3-6B	71.80%	42.60%	38.20%	70.70%	55.83%
Baichuan2-7B-chat	81.20%	39.50%	51.20%	63.10%	58.75%

Table 5: Comparison of results between the optimal method and other existing models on different datasets.

4 Experiments

4.1 Experimental Setup

System Configuration The experiments were conducted on a system equipped with eight NVIDIA GeForce RTX 3090 GPUs, ensuring high-performance computing capabilities for model training and evaluation.

Model Input Handling To address the constraints of the model architecture, a maximum token length of 2048 was set for input sequences. In instances where input sequences exceeded this limit, a truncation process was applied to maintain compatibility with the model. The model's generation process was configured using the following parameters:

- **Maximum New Tokens (`max_new_tokens`):** The value of `nt` determined the maximum number of tokens generated for different datasets. For subjective question datasets, `nt` was set to 128, while for objective question datasets (EPRSTMT, TNEWS, OCNLI, BUSTM), `nt` was set to 1.
- **Number of Beams (`num_beams`):** A value of 1 was chosen to restrict the generation process to a single beam, promoting deterministic output.
- **Top-p (`top_p`):** Set to 0.9, this parameter controlled the nucleus sampling probability, allowing for a diverse yet constrained generation.
- **Temperature (`temperature`):** A low value of 0.1 was chosen to focus the generation process, reducing randomness and enhancing output coherence.

These parameter configurations aimed to strike a balance between model expressiveness and computational efficiency, facilitating meaningful experiments within the specified token constraints. The adaptability of this approach to different question types was emphasized by the specific `nt` values assigned to each dataset, laying the groundwork for subsequent analyses in the following sections.

4.2 Ablation Study

Resource Consumption Comparison Experiment We conduct an ablation study in order to explore the resource consumption of PEFT and Full-Parameter methods.As shown in Table 2, it can be found that compared with full-parameter fine-tuning, the number of training parameters of PEFT is greatly reduced, and using full-parameter tuning requires about 2-3 times the training time cost, demonstrating that PEFT methods reduce resource overhead.

PEFT performs better on objective questions As shown in Table 3,it can be found that PEFT performs better on objective questions.

Comparison of different methods on different datasets In Table. 4, we further compare the model fine-tuned using the PEFT method with other existing models, of which results reveal that the fine-tuned model achieves superior performance.

4.3 Results

As shown in the Table 5, we tested the performance of the model on objective questions using different fine-tuning methods across datasets such as EPRSTMT, TNEWS, OCNLI, and BUSTM. The 'final model' refers to the model that has undergone RLHF (Reinforcement Learning from Human Feedback) technology. It can be observed that models fine-tuned using LoRA and RepAdapter methods can maintain the efficiency of model fine-tuning while achieving accuracy close to full fine-tuning. Furthermore, the performance of the model improves further after applying RLHF. The llama2_RepAdapter model, which achieved the highest average precision, was chosen as the base model. The llama2_rm was utilized as the Reinforcement Model (RM). Training was then conducted on the hh_rlhf_cn test dataset using Proximal Policy Optimization (PPO), culminating in the final model, llama2_RLHF. For subjective questions, we conducted separate tests on models before and after using parameter-efficient fine-tuning (PEFT) and reinforcement learning from human feedback. From Table 6, it can be observed that models using PEFT methods can provide answers to long textual content. In contrast, models using RLHF can further articulate the reasoning behind the answers, enhancing interpretability.

5 Conclusion

Our method, integrating Parameter-Efficient Fine-Tuning (PEFT) with Reinforcement Learning from Human Feedback (RLHF), has demonstrated remarkable efficiency within the RLHF framework. Furthermore, it achieved impressive outcomes on the datasets we utilized. In the RLHF phase, our approach not only met but also exceeded our expectations, providing answers and rational explanations for the targeted questions, thereby successfully fulfilling the objectives of our experiment.

Datasets	Question	Base	SFT	RLHF
BUSTM	请根据提供的中文句子，判断它们是否属于同一语义。 文本: 文本1:那你给我聊个天 文本2:那你给我跳个舞吧 选项：A. 属于 B. 不属于	A. 属于	A. 属于	A. 属于。 文本1和文本2都是请求对方做某事的请求，即“聊天”和“跳舞”。它们的意思相似，都是在请求对方做某件有趣的事情。因此，它们属于同一语义。
OCNLI	请根据下面的前提和假设句子进行推理，选择正确的关系。文本: 前提: 一月份跟二月份肯定有一个月份有. 假设: 肯定有一个月份有 选项：A. 矛盾 B. 中立 C. 蕴含	C. 蕴含	C. 蕴含	C. 蕴含 解释：根据前提，“一月份跟二月份肯定有一个月份有”，可以得出结论：“肯定有一个月份有”。因此，选项C“蕴含”是正确的关系，因为前提表达了一个结论，而假设则是这个结论的表述。假设是前提的一部分，它表达了前提的内容，因此它与前提是蕴含关系。选项A“矛盾”表示前提和假设之间存在矛盾，但是这两个句子并没有表达出矛盾的关系，因此不符合实际情况。选项B“中立”表示前提和假设之间没有明显的关系，但是根据前提和假设的含义，它们确实存在一定的关系，因此选项B也不准确。因此，选项C“蕴含”是正确的关系，因为前提和假设之间存在明显的关系，前提表达了一个结论，而假设则是这个结论的表述，它们之间存在蕴含关系。
EPRSTMT	在此任务中，判断下列评价是好评还是差评 评价: 大小合适，对孔比较准，手感不错，就是价格偏高，不赖脏。 答案:	好评	好评	评价：大小合适，对孔比较准，手感不错，就是价格偏高，不赖脏。答案：评价中提到了产品的大小合适，对孔比较准，手感不错，这些都是好评，但是价格偏高，不赖脏，这些都是差评。因此，这个评价既有好评也有差评，不能简单地归为一个类别。如果要简单总结，可以说这个评价是中立的，因为它既有优点也有缺点，没有明显的偏向。但是，如果要根据评价中提到的优点和缺点来分析，可以得出这个产品的价格较高，但是它的大小合适，对孔比较准，手感不错，这些都是优点，可以让消费者考虑购买。但是，价格较高可能会影响消费者的购买决策，因此，在购买前需要考虑自己的预算和需求。
TNEWS	待分类的新闻标题:女儿被婆婆带一天脚痛，我脱下她鞋子一看，回来我就将婆婆赶出门 待分类的新闻关键字: 选项: A:家庭新闻 B:娱乐新闻 C:体育新闻 D:游戏新闻 E:军事新闻 ...	待分类的新闻标题:女儿被婆婆带一天脚痛，我脱下她鞋子一看，回来我就将婆婆赶出门 待分类的新闻关键字: 选项: A:家庭新闻	F	A: 家庭新闻 关键字：女儿、婆婆、脚痛、鞋子、赶出门。 这则新闻涉及到一个家庭的矛盾，女儿的脚痛引起了婆婆的帮助，但是女儿的母亲发现婆婆并没有为女儿穿上合适的鞋子，并且婆婆的行为让女儿感到受到了侵犯，最终导致母亲赶出了婆婆。这是一则家庭新闻，涉及到家庭成员之间的关系和争端。

Table 6: Comparison of responses of SFT model and RLHF model

References

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Chen, S.; Ge, C.; Tong, Z.; Wang, J.; Song, Y.; Wang, J.; and Luo, P. 2022. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35: 16664–16678.

Chen, Z. 2022. Research on Intelligent Semantic Recognition and Self-Organizing Feature Mapping of Chinese Linguistics Under Big Data Informationization. In *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 1123–1126.

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240): 1–113.

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

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Lian, D.; Zhou, D.; Feng, J.; and Wang, X. 2022. Scaling & Shifting Your Features: A New Baseline for Efficient Model Tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Luo, G.; Huang, M.; Zhou, Y.; Sun, X.; Jiang, G.; Wang, Z.; and Ji, R. 2023. Towards Efficient Visual Adaption via Structural Re-parameterization. *arXiv preprint arXiv:2302.08106*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shancheng, T.; Yunyue, B.; and Fuyu, M. 2018. A Semantic Text Similarity Model for Double Short Chinese Sequences. In *2018 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS)*, 736–739.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Workshop, B.; Scao, T. L.; Fan, A.; Akiki, C.; Pavlick, E.; Ilić, S.; Hesslow, D.; Castagné, R.; Luccioni, A. S.; Yvon, F.; et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.; Li, X.; Lin, X. V.; et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.