PAFT: Prompt-Agnostic Fine-Tuning

Anonymous EMNLP submission

Abstract

Fine-tuning large language models (LLMs) often causes overfitting to specific prompt wording, where minor phrasing variations drastically reduce performance. To address this, we propose *Prompt-Agnostic Fine-Tuning* (PAFT), a method that enhances robustness through dynamic prompt variation during training. PAFT first generates diverse synthetic prompts, then continuously samples from this set to construct training instances, forcing models to learn fundamental task principles rather than surface-level patterns. Across systematic evaluations using both supervised fine-tuning (SFT) and reinforcement learning fine-tuning (RLFT), PAFT consistently demonstrates improved performance on benchmarks for question answering, mathematical reasoning, and tool use. It achieves 7% higher generalization accuracy on unseen prompts than standard methods with similar training efficiency. Notably, models trained with PAFT attain 3.2× faster inference speeds due to reduced prompt sensitivity. Ablation studies further validate effectiveness of PAFT, while theoretical analysis reveals that PAFT can effectively enhance the cross-domain generalization ability of LLM.

1 Introduction

Large language models (LLMs) have demonstrated remarkable success across diverse natural language processing (NLP) tasks (Zhao et al., 2024; Xu et al., 2023). To further enhance the performance of LLMs on specific downstream tasks, supervised fine-tuning (SFT) (Ouyang et al., 2022; Devlin et al., 2019) and reinforcement learning fine-tuning (RLFT) (Wang et al., 2024) has emerged as a widely adopted strategy. These methods typically augment input data with task-specific instructions and construct dialogue datasets with expected outputs, enabling models to learn task-specific patterns. Empirical studies have shown that SFT and RLFT can substantially improve model performance on downstream tasks (Raffel et al., 2023; Hu et al., 2023b; Wei et al., 2022).

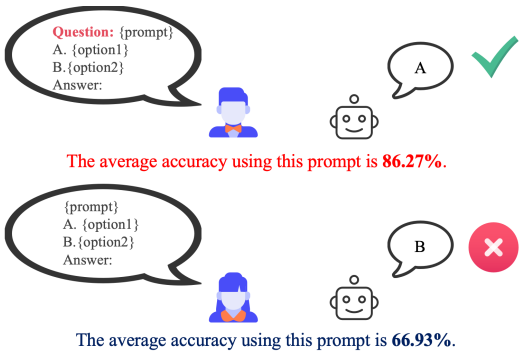

Figure 1: This figure shows how minor prompt changes drastically impact model accuracy. For instance, a one-word alteration to a prompt for the same user question reduced dataset accuracy from 86.27% to 66.93%. This highlights severe performance swings in models lacking prompt robustness.

However, as shown in Figure 1, a critical limitation of current fine-tuning methods is their lack of prompt robustness, as further detailed in Sec. 3. Reliance on fixed instruction prompts (Mishra et al., 2022; Chung et al., 2022) often leads to overfitting on specific prompts patterns (Zhang et al., 2024; Kung and Peng, 2023). Consequently, models become brittle: minor deviations between user and training prompts can significantly degrade inference performance (Mialon et al., 2023; Raman et al., 2023). This brittleness manifests, for example, as substantial accuracy drops in QA tasks with altered prompt phrasing (Wei et al., 2024), or as poor instruction following in chatbots and AI agents when commands deviate from those encountered during training (Hong et al., 2024; Sahoo et al., 2025). Such sensitivity also raises fairness and reliability concerns in algorithmic comparisons (Voronov et al., 2024). This vulnerability is particularly acute when users, unfamiliar with specific SFT prompt structures, provide highly divergent inputs, potentially causing fine-tuned models to perform near random guessing levels (Polo

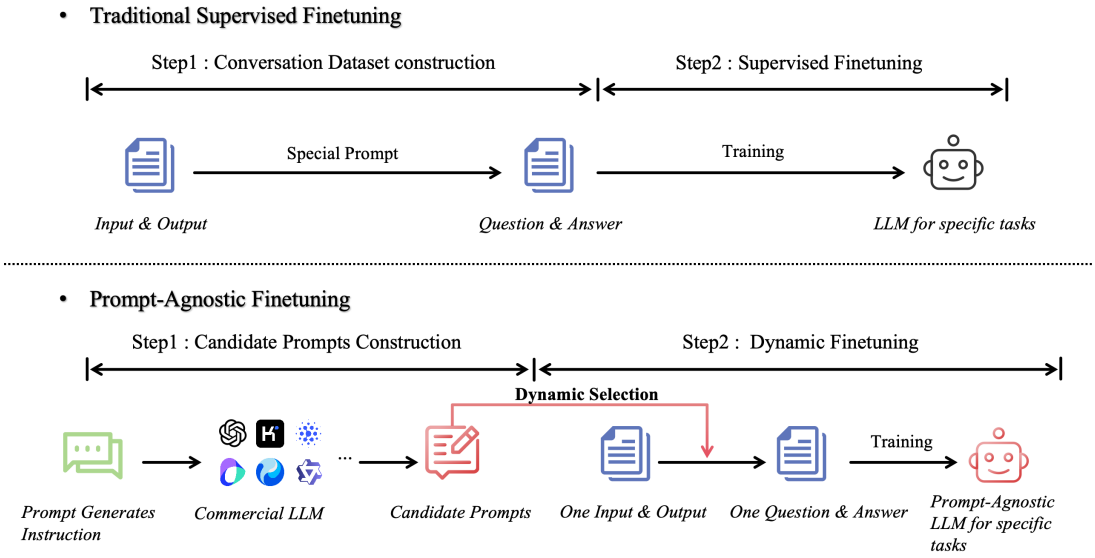

Figure 2: An overview of PAFT: This figure contrasts SFT with PAFT. While SFT relies on fixed datasets and predefined prompts—limiting robustness and cross-prompt generalization—PAFT employs dynamic prompt selection during training, significantly enhancing prompt robustness and generalization capabilities. By leveraging commercial LLMs to generate diverse candidate prompts, PAFT delivers a more scalable and generalizable solution for large language model adaptation.

et al., 2024). Notably, prompt robustness in SFT has received limited attention, with most existing work focusing on in-context learning and prompt tuning (Shi et al., 2024; Ishibashi et al., 2023).

To address this critical gap, we introduce PAFT, a novel framework that dynamically adapts to diverse training prompts. To our knowledge, PAFT is the first systematic approach to improving prompt robustness in both SFT and RLFT, a vital but underexplored area. Unlike traditional methods prone to overfitting specific prompt patterns, PAFT enables models to grasp underlying task semantics, ensuring robust performance across varied human-written prompts. PAFT operates in two phases (Figure 2): first, constructing a diverse set of high-quality synthetic prompts that capture essential task semantics with linguistic variability (Sec. 4.1); second, employing dynamic fine-tuning by sampling from this curated set to expose the model to various formulations (Sec. 4.2). Extensive evaluations demonstrate that PAFT significantly boosts model robustness and generalization to diverse prompts, maintains state-of-the-art downstream performance, and can potentially improve inference speed while preserving training efficiency. These findings highlight PAFT as a promising direction for developing more robust, user-friendly language models.

Our key contributions are as follows: (a) We highlight that fine-tuning with fixed prompts results in poor generalization to unseen prompts and severe performance degradation (Sec. 3). (b) We propose PAFT, a novel framework incorporating candidate prompt construction and dynamic fine-tuning, to enhance the prompt robustness of fine-tuned models (Sec. 4). (c) We empirically demonstrate the consistent and robust performance of PAFT across diverse downstream tasks, fine-tuning algorithms, and varied test prompts, including those unseen during training (Sec. 5). (d) We provide theoretical evidence that PAFT effectively enhances the cross-domain generalization of LLMs (Sec. 6).

2 Related Work

Prompt Optimization. Prompt engineering critically influences LLM performance, driving numerous prompt optimization approaches (Chang et al., 2024; Li, 2023; Diao et al., 2023; Sun et al., 2022). Notable methods include INSTINCT (Lin et al., 2024), which leverages neural network bandits with LLM embeddings for search efficiency. ZOPO (Hu et al., 2024), which employs localized search strategies. BATprompt (Shi et al., 2024), which integrates robustness through natural language perturbations. While these approaches excel at identifying single high-performance prompts, models fine-tuned on such prompts remain vulner-

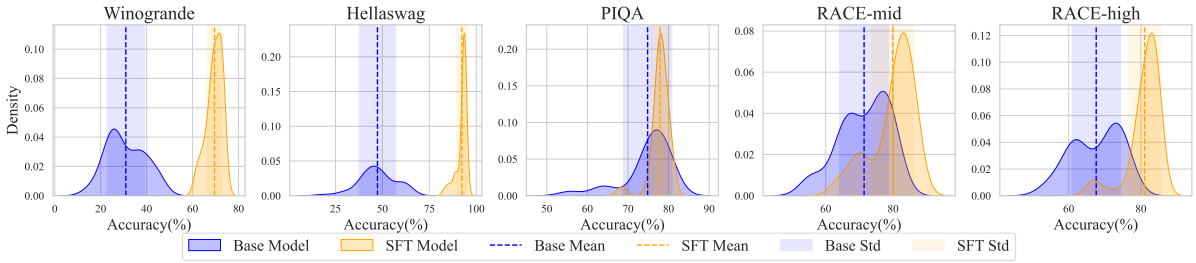

Figure 3: This figure presents experimental results across four datasets comparing base and SFT model performance on 450 diverse prompts (both human-written and LLM-generated). Probability distribution plots reveal that despite SFT's overall accuracy improvements, substantial performance variability persists—certain prompts yield markedly lower accuracy, with high standard deviations indicating significant prompt-dependent fluctuations. These findings underscore crucial impact of prompt and demonstrate the necessity for prompt-agnostic fine-tuning approaches.

able to prompt variations. Our work, in contrast, addresses this limitation by simultaneously enhancing prompt robustness and optimizing performance across the entire prompt space rather than focusing on isolated optimal prompts.

Fine-tuning (FT). SFT and RLFT constitute the predominant paradigms for adapting LLMs, prized for their efficiency. These approaches split into two categories: soft prompt tuning, which optimizes continuous input vectors while preserving base model parameters (Li and Liang, 2021; Liu et al., 2022), and full/parameter efficient fine-tuning (PEFT) (Shu et al., 2024; Ouyang et al., 2022; Liu et al., 2021; Lester et al., 2021). Among PEFT techniques, Low-Rank Adaptation (LoRA) (Hu et al., 2022) predominates by freezing pre-trained weights while introducing trainable low-rank matrices, with recent variants enhancing generalization and reducing overfitting (Chen et al., 2023; Si et al., 2024; Wei et al., 2024). Instruction tuning (Sanh et al., 2022) further improves ability of model to follow diverse task-specific instructions. However, existing methods—particularly soft prompt tuning—still exhibit limited prompt robustness, leaving models vulnerable to prompt variations. Our work addresses this critical limitation while maintaining computational efficiency.

3 Preliminaries

To systematically study the impact of prompt variations on fine-tuned models, we conducted comprehensive experiments across multiple downstream tasks using LLaMA3-8B (Meta, 2024) with LoRA fine-tuning. We constructed a comprehensive set of over 450 prompts (both human-written and LLM-generated), covering a wide range of language styles, task-specific instructions, and formatting variations. Figure 3 presents a statistical analysis of the accuracy distribution for both the base and SFT models across these prompts, revealing a key finding: the formulation of the prompt dramatically influences the performance of the model regardless of the type of task, with only 10% of the prompts producing near-optimal results. Minor prompt modifications (e.g., rephrasing, punctuation, reordering) induce substantial fluctuations.

For example, the addition of "Question" improves accuracy by 20% (Figure 1). This sensitivity highlights the fragility of current fine-tuning methods and their strong dependence on specific prompt formulations. These findings align with prior work (He et al., 2024; Voronov et al., 2024; Salinas and Morstatter, 2024; Min et al., 2022; Gao et al., 2021b). This widespread sensitivity demonstrates a fundamental limitation in current fine-tuning approaches, extending findings from previous research across diverse task domains. Based on these insights, we propose PAFT, which decouples model performance from specific prompt formulations, ensuring consistent results across prompt variations and enhancing practical applicability in real-world scenarios.

4 The PAFT Framework

In this section, we introduce PAFT in detail. As shown in Figure 2, the PAFT framework consists of two key stages: candidate prompt construction (Sec. 4.1) and dynamic fine-tuning (Sec. 4.2).

4.1 Candidate Prompt Construction

To ensure the robustness and effectiveness of PAFT across diverse prompts, we design a comprehensive prompt construction framework that aims to generate diverse and meaningful candidate prompts efficiently, enabling the model to generalize across

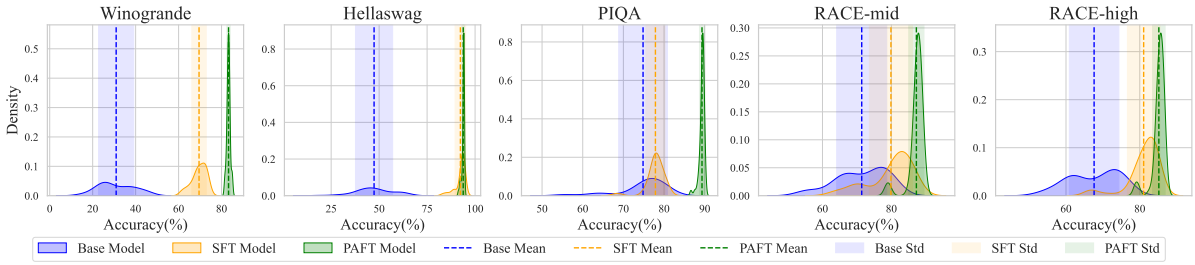

Figure 4: As a visual comparison to Figure 3, we present performance distributions of base models, SFT models, and PAFT across multiple reasoning and reading comprehension tasks. The probability distribution plots illustrate performance on unseen test prompts (both human-written and LLM-generated) not used during PAFT training. Results clearly demonstrate PAFT consistently achieves higher accuracy and lower variance across all tasks, confirming its effectiveness in enhancing prompt robustness.

different prompt formats. Our approach leverages the powerful generative capabilities of LLMs (Kohl et al., 2024) and comprises three key phases.

Diverse LLM Ensemble. We employ 10 mainstream LLMs with varied generation capabilities (OpenAI et al., 2024; Bai et al., 2023; Ouyang et al., 2022) to capture the inherent variability in task interpretation stemming from differences in pre-training data, architectures, and optimization objectives (Minaee et al., 2024; Zhao et al., 2024). This diversity ensures comprehensive coverage of prompt formulations across linguistic styles and instructional approaches, effectively mitigating single-model generation biases.

Dual Prompting Strategy. We combine few-shot and zero-shot techniques to balance quality and diversity. Few-shot prompting leverages in-context learning with curated human examples to generate task-aligned, semantically coherent prompts. Zero-shot prompting encourages diverse linguistic styles and structural variations without explicit examples. By generating 20 prompts with each strategy, we create a comprehensive set spanning high-quality and varied formulations, exposing the model to realistic prompt quality distributions and enhancing robustness to real-world scenarios. See Appendix C for details.

Rigorous Evaluation Design. We randomly partition generated prompts into training and test sets (8:1 ratio), ensuring completely distinct prompts in each set. This approach exposes the model to diverse prompt styles during training while providing a robust testbed for assessing generalization to novel formulations. By evaluating on entirely unseen prompts, we confirm that performance improvements reflect genuine ability to handle diverse prompt formulations rather than overfitting to specific patterns. This framework ensures PAFT learns task semantics independently of prompt phrasing, enabling effective generalization across real-world scenarios while providing a scalable, cost-effective solution for improving prompt robustness.

Algorithm 1 The PAFT Framework

1: **Input:** Generate a good candidate prompt training set $\mathbb{P}$; A task-specific dataset $\mathbb{D}$; The number of training epochs T; The number of same prompt training K; Initialized trainable parameters θ_0^0; Learning rate η_θ
2: **Output:** Fine-tuned model parameters θ^*.
3: **for** each epoch $t = 0$ **to** $T - 1$ **do**
4: $p \leftarrow$ RandomlySample($\mathbb{P}$) // Randomly select a prompt from the candidate set
5: $k \leftarrow 0$ // Initialize the step counter
6: **for** each data point $(x, y) \in \mathbb{D}$ **do**
7: $\mathbf{I} \leftarrow$ InputConstruction(x, p) // Construct input using prompt p and data x
8: $\theta_t^{k+1} \leftarrow \theta_t^k - \eta_\theta \nabla_\theta \ell(\theta, \mathbf{I})|_{\theta=\theta_t^k}$
9: $k \leftarrow k + 1$ // Increment the step counter
10: **if** $k \mod K == 0$ **then**
11: $p \leftarrow$ RandomlySample($\mathbb{P}$)
12: **end if**
13: **end for**
14: $\theta_{t+1}^0 \leftarrow \theta_t^k$
15: **end for**
16: **return** $\theta^* = \theta_T$

4.2 Dynamic Fine-Tuning

Dynamic Fine-Tuning Algorithm. Our PAFT framework enhances the robustness of LLMs through systematic prompt diversification. As shown in Algorithm 1, each training epoch t randomly samples a prompt p from synthetic candidates $\mathbb{P}$ (line 4), exposing the model to varied linguistic styles. For each data point $(x, y) \in \mathbb{D}$ (line 6), the selected prompt is reused for K consecutive steps (lines 7-9), constructing inputs via $\mathbf{I} = \text{InputConstruction}(x, p)$ (line 7) and updating parameters θ using gradient-based optimization like SGD (Sra et al., 2011) or AdamW (Loshchilov and Hutter, 2019) (line 8). After K steps, a new

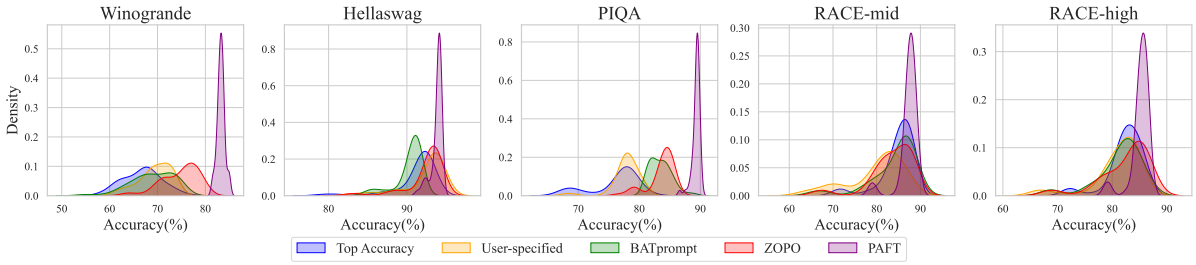

Figure 5: The performance of TopAccuracy, User-specified, BATprompt, ZOPO, and PAFT models is compared on multiple reasoning and reading comprehension tasks. Results are reported in terms of their correct distribution. The tests are conducted on a test set of 50 unseen prompts, different from the ones used in training. The PAFT model shows superior performance compared to other baselines, achieving higher accuracy and lower variance in all tasks.

prompt is sampled (lines 10-11), ensuring multiple prompt exposures per epoch. Each epoch initializes with final parameters from the previous one: $\theta_{t+1}^0 = \theta_t^K$ (line 12), maintaining learning continuity until final parameters $\theta^* = \theta_T$ are achieved after T epochs (line 16).

Benefits of Dynamic Fine-Tuning. Dynamic fine-tuning in PAFT significantly enhances LLM robustness and generalization by exposing the model to diverse prompts during training. This approach mitigates overfitting to fixed prompts, fostering the learning of more generalizable representations less sensitive to specific formulations. Consequently, PAFT achieves consistent performance across varied and unseen prompts, crucial for real-world applications with diverse user input. By reducing reliance on manual prompt engineering, dynamic fine-tuning offers an efficient and scalable solution for improving LLM adaptability.

5 Empirical Results

We evaluate our PAFT framework through comprehensive experiments. Sec. 5.1 describes datasets and experimental setup, Sec. 5.2 analyzes key findings, and Sec. 5.3 presents ablation studies examining critical framework components.

5.1 Datasets and Setup

Benchmark Selection. As the pioneering work addressing prompt robustness in large language models (LLMs) through training, we generate task-specific candidate prompts for each downstream task. Following Hu et al. (2023a); Wei et al. (2024), we selected diverse benchmarks for evaluation: PIQA (Bisk et al., 2019) for commonsense QA, Winogrande (Sakaguchi et al., 2019) and Hellaswag (Zellers et al., 2019) for commonsense reasoning, RACE (Lai et al., 2017) for reading comprehension, and T-eval (Chen et al., 2024) for tool use capabilities in our SFT experiments. Additionally, we employed GSM8K (Cobbe et al., 2021) for mathematical reasoning in our RLFT experiments.

Experimental Setup and Baseline Comparisons. As described in Section 4.1, we generate a diverse set of 400 training prompts only using LLMs and create a separate test set of 50 prompts comprising both human-written and LLM-generated instructions. This separation ensures rigorous evaluation of model generalization to unseen prompt formulations (details in Appendix C). We establish five baselines to isolate prompt engineering's impact on fine-tuning performance: the original pre-trained model (Base Model); the model fine-tuning with human-designed prompts (User) following Wei et al. (2024); the model fine-tuning with the highest accuracy of training prompts (TopAccuracy); the model fine-tuning with BATprompt (Shi et al., 2024) most robust prompt (BATprompt); and fine-tuning with ZOPO (Hu et al., 2024) optimal prompt selection (ZOPO). All models, including baselines, are evaluated on identical test prompts, enabling direct comparison of performance consistency across methods. Notably, we leverage Group Relative Policy Optimization (Shao et al., 2024) (GRPO) as an exemplary case to demonstrate capabilities of PAFT within the RLFT paradigm. Our implementation leverages the Llama-factory (Zheng et al., 2024) and is evaluated using the Opencompass (Contributors, 2023). Detailed experimental configurations are provided in Appendix A. All experiments are conducted on NVIDIA A100, V100, 4090, and L40 GPUs to ensure efficient and scalable evaluation.

5.2 Main Results

Prompt Robustness. As demonstrated across Tables 1 and Figures 4, 5, 7, and 8, PAFT exhibits remarkably low variance across all evaluation tasks,

Table 1: Performance comparison of different fine-tuning methods on the test prompt sets across various reasoning and reading comprehension tasks using the LLaMA3-8B (Meta, 2024) with LoRA rank 8. Results are reported as average accuracy, standard deviation. PAFT demonstrates superior performance, achieving the highest accuracy and lowest variance across all tasks. The last rows show the comparison of PAFT with the second-best performing method (underlined). The Top column indicates the percentage of test prompts with a correct rate of 90% for Hellaswag, 80% for Winogrande, and 85% for other datasets.

Methods	**Hellaswag**			**PIQA**			**Winogrande**			**RACE-mid**			**RACE-high**			**Average**		
Metric	Mean	Std	Top	Mean	Std	Top	Mean	Std	Top	Mean	Std	Top	Mean	Std	Top	Mean	Std	Top
Base Model	47.36	±9.78	0%	74.68	±6.24	0%	45.15	±11.78	0%	71.39	±7.33	0%	67.62	±6.78	0%	61.24	±8.38	0%
User	92.35	±2.78	0%	77.87	±2.36	0%	78.16	±7.97	0%	79.88	±6.32	22%	81.05	±4.45	4%	81.86	±4.78	5%
TopAccuracy	91.27	±2.79	86%	75.96	±3.89	0%	66.77	±3.94	0%	84.81	±4.06	59%	82.45	±3.26	14%	80.25	±3.63	32%
BATprompt	90.30	±1.79	78%	83.41	±1.74	16%	69.01	±4.45	0%	83.92	±5.38	65%	81.33	±4.21	12%	81.56	±3.51	34%
ZOPO	92.46	±2.43	86%	83.52	±2.23	27%	74.75	±3.81	0%	83.50	±5.05	51%	82.36	±4.53	35%	83.32	±3.61	40%
PAFT	**93.83**	**±0.70**	**100%**	**89.33**	**±0.63**	**100%**	**82.09**	**±0.81**	**100%**	**87.26**	**±2.23**	**94%**	**85.17**	**±1.71**	**73%**	**87.57**	**±1.57**	**94%**
↪ Improv.	+1.37	-1.09	14%	+5.81	-1.11	73%	+3.93	-3.00	100%	+2.45	-1.83	29%	+2.72	-1.55	38%	+4.25	-1.94	54%

Table 2: Comparison of inference time (in hours) for different fine-tuning methods. PAFT shows better inference efficiency than other methods. The last line shows the multiple of PAFT improvement.

Inference time/h	**Hellaswag**	**PIQA**	**Winogrande**	**RACE**	**Average**
Base Model	3.97	1.35	1.72	6.24	3.32
User	6.52	0.98	3.27	8.23	4.75
TopAccuracy	5.75	1.13	2.76	7.56	4.30
BATprompt	4.57	1.57	3.14	7.98	4.32
ZOPO	5.12	0.87	3.23	8.28	4.38
PAFT	**1.19**	**0.39**	**0.45**	**2.08**	**1.02**
↪ Improv.	×3.3	×2.23	×3.82	×3.00	×3.25

indicating superior prompt robustness. This enhanced stability stems from our dynamic prompt selection strategy (Sec. 4.2), which continuously adjusts prompts during training, compelling the model to learn essential task features rather than overfitting to specific prompt formats. In contrast, baseline approaches face significant limitations: user prompts rely on manual design with inconsistent quality; TopAccuracy and ZOPO tend to overfit to high-performing training prompts with poor generalization; and while BATprompt addresses robustness, it remains less effective than our method. The low variance of PAFT translates to more stable performance and stronger generalization across diverse prompts, enabling development of more user-friendly QA systems, format-independent agent systems, and directly evaluate the true ability of LLMs by better decoupling the ability from the prompting engineering. Notably, PAFT achieves acceptable performance across most prompts, significantly outperforming all baselines (Table 1, Top column) while maintaining high training efficiency (detailed in Appendix B).

SOTA Performance. As demonstrated in Table 1 and Figures 4, 5, 7, and 8, PAFT consistently achieves superior performance across all evaluated tasks. The model significantly outperforms existing baselines on diverse natural language processing challenges, establishing a new state of the art. This exceptional performance can be primarily attributed to the prompt robustness inherent in PAFT—a capability that enables the model to extract the fundamental essence of each task regardless of prompt variations. The cornerstone of these improvements lies in our innovative architectural design, which effectively decouples prompt formulation from task representation. This decoupling allows PAFT to focus exclusively on learning essential task-specific features rather than becoming entangled in prompt-specific nuances.

Inference Efficiency. PAFT enhances inference efficiency by improving the model's understanding of core task semantics, enabling concise, accurate responses with fewer tokens. Our measurements across all test prompts and datasets (Table 2) demonstrate that PAFT consistently achieves faster inference speeds than baseline methods. This efficiency stems from the model's prompt robustness—performance remains stable regardless of prompt wording variations, eliminating the need for prompt-specific adaptation. Our training with diverse prompt formulations prevents performance degradation when handling unexpected inputs, making PAFT particularly valuable for real-world applications requiring rapid responses, such as dialogue systems and AI agents, while simultaneously reducing computational resource requirements. See Appendix B.2 for more detail.

5.3 Ablation Studies

Hyperparameter Robustness. This ablation study demonstrates the robustness of PAFT to the hyperparameters K (iterations per prompt) and T (epochs). As shown in Table 3, PAFT achieves stable performance across a broad range of K (1

Table 3: Performance comparison of PAFT with varying hyperparameters K (number of iterations per prompt) and T (number of epochs) across multiple reasoning and reading comprehension tasks. Results are reported as mean accuracy (± standard deviation) on the Hellaswag, PIQA, Winogrande, RACE-mid, and RACE-high datasets. The best results for each metric are highlighted in bold.

# K and T	Hellaswag	PIQA	Winogrande	RACE-mid	RACE-high	Average
$K = 1, T = 3$	93.58 (± 1.47)	**89.33**(± 0.63)	81.78 (± 1.11)	86.30 (± 2.73)	84.35 (± 2.24)	87.07 (± 1.64)
$K = 2, T = 3$	93.59 (± 1.24)	88.37 (± **0.49**)	**82.09** (± **0.81**)	86.30 (± 2.64)	84.02 (± 2.24)	86.87 (± 1.48)
$K = 4, T = 3$	**93.83**(± 1.10)	89.07 (± 0.53)	81.96 (± 1.15)	**87.26** (± **2.23**)	**85.17** (± 1.71)	**87.46** (± **1.34**)
$K = 8, T = 3$	**93.83** (± **0.70**)	88.99 (± 0.59)	82.69 (± 0.97)	86.25 (± 2.75)	84.36 (± 2.06)	87.22 (± 1.41)
$K = 1, T = 6$	93.37 (± 1.47)	88.32 (± 0.68)	81.05(± 3.44)	84.40 (± 2.30)	83.34(± **1.66**)	86.10 (± 1.91)

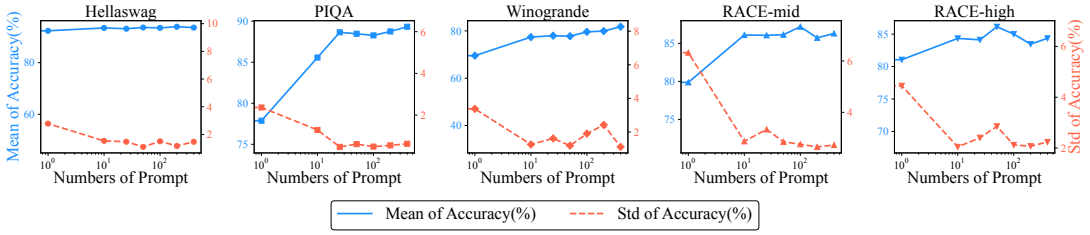

Figure 6: Scaling Law of Training Prompt Numbers: Mean and Standard Deviation of Accuracy Across Different Datasets. The x-axis represents the number of prompts on a logarithmic scale, while the y-axis shows the mean accuracy (left) and standard deviation of accuracy (right) for each dataset.

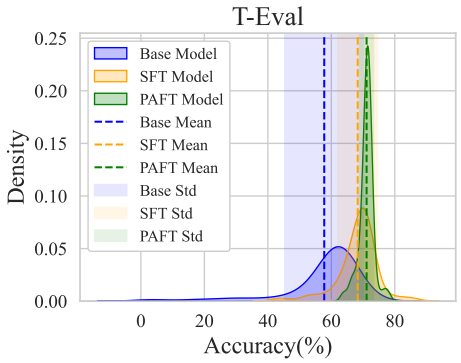

Figure 7: The performance of base model, SFT model, and PAFT model is compared on T-Eval.

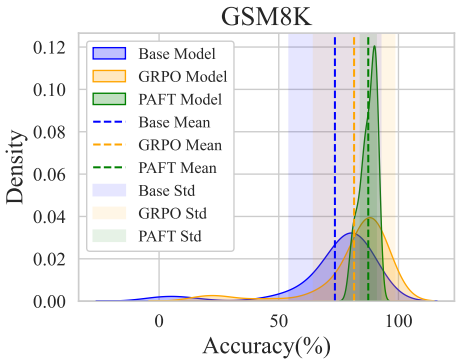

Figure 8: The performance of base model, GRPO model, and PAFT model is compared on GSM8K.

to 8) and T (3 to 6) values, with minimal fluctuations in accuracy and variance. Notably, PAFT achieves near-optimal performance with default settings ($K = 4$, $T = 3$), attaining an average accuracy of 87.46%(±1.34) across all tasks. This robustness reduces the need for extensive hyperparameter tuning, making PAFT a practical and efficient solution for real-world applications.

Impact of Training Prompt Quantity. We conduct an ablation study to investigate the impact of varying numbers of training prompts on model performance, thus validating the effectiveness of PAFT. The experimental results, shown in Figure 6, demonstrate that as the number of prompts increases, the average accuracy of the model significantly improves, while the standard deviation decreases, indicating more stable and reliable performance. However, the performance gains diminish as the number of prompts increases, with only marginal improvements observed beyond a certain threshold. This suggests that while adding prompts can enhance performance, PAFT achieves competitive results with a minimal number of prompts, rendering excessive prompts unnecessary. In most cases, PAFT achieves strong performance with as few as 10 high-quality prompts, and further increases yield only marginal gains. The efficiency of PAFT is particularly notable, as it delivers excellent performance with a minimal number of prompts, making it highly suitable for resource-constrained scenarios where computational efficiency is critical. These findings underscore the

practicality and efficiency of PAFT, offering a robust and efficient solution for real-world applications.

6 Theoretical Insights

The capability of PAFT to generalize effectively to unseen prompt formulations can be rigorously understood through the lens of domain adaptation theory (Ben-David et al., 2006, 2010). In this theoretical construct, the collection of training prompts $\mathcal{P}_{\text{train}}$ along with the task-specific training data $\mathcal{D}_{\text{train}}$ delineates the source domain. Besdies, the set of novel test prompts $\mathcal{P}_{\text{test}}$, paired with $\mathcal{D}_{\text{test}}$, represents the target domain. PAFT aims to learn a model $f^* \in \mathcal{H}$, where $\mathcal{H}$ denotes the hypothesis class, by minimizing the empirical risk computed over instances (x, p_i, y) where each prompt p_i is sampled from $\mathcal{P}_{\text{train}}$.

A foundational result from domain adaptation theory (Ben-David et al., 2010) provides an upper bound on the expected risk of f^* on the target prompt distribution $\mathcal{R}_{\mathcal{P}_{\text{test}}}(f^*)$ with $\min_{f\in\mathcal{H}}(\mathcal{R}_{\mathcal{P}_{\text{train}}}(f) + \mathcal{R}_{\mathcal{P}_{\text{test}}}(f))$:

$$\mathcal{R}_{\mathcal{P}_{\text{test}}}(f^*) \leq \text{Disc}(\mathcal{P}_{\text{train}}, \mathcal{P}_{\text{test}}) + \mathcal{C}(\mathcal{H}, N) + \hat{\mathcal{R}}_{\mathcal{P}_{\text{train}},N}(f^*) + \lambda^* . \quad (1)$$

Here, $\hat{\mathcal{R}}_{\mathcal{P}_{\text{train}},N}(f^*)$ is the empirical risk on N training prompts. The term $\mathcal{C}(\mathcal{H}, N)$ signifies model complexity (e.g., related to Rademacher complexity (Yin et al., 2020)), which typically diminishes as the number of distinct training prompts N increase; this term captures the generalization gap on the source domain. The divergence between the training and test prompt distributions is quantified by $\text{Disc}(\mathcal{P}_{\text{train}}, \mathcal{P}_{\text{test}})$. Finally, λ^* encapsulates the optimal joint error achievable by a hypothesis in $\mathcal{H}$ on both domains. The key of PAFT is strategically designed to optimize this bound for improved generalization.

Complexity Control. By employing a substantial number of distinct training prompts N, PAFT inherently works to reduce the complexity term $\mathcal{C}(\mathcal{H}, N)$. This ensures that the model performance observed on the training prompts becomes a more faithful estimator of its true performance across the entire $\mathcal{P}_{\text{train}}$ distribution, fostering more stable learning. This effect is empirically supported by our ablation studies in Section 5.3 (Figure 6), which demonstrate improved stability with more prompts.

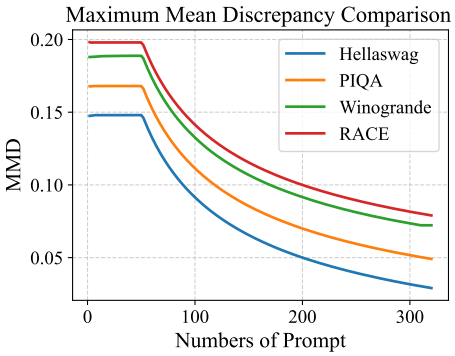

Figure 9: This figure demonstrates the change in MMD for different numbers of $\mathcal{P}_{\text{train}}$ and the same $\mathcal{P}_{\text{test}}$.

Domain Alignment. Minimizing the domain discrepancy term $\text{Disc}(\mathcal{P}_{\text{train}}, \mathcal{P}_{\text{test}})$ is critically dependent on constructing a diverse and comprehensive set of candidate prompts $\mathcal{P}_{\text{train}}$ (see Section 4.1 for details). A more diverse $\mathcal{P}_{\text{train}}$ is more likely to effectively cover or closely approximate the diverse and unseen distribution of test prompts $\mathcal{P}_{\text{test}}$. This approximation reduces the divergence between the training and test prompt distributions and enhances the transferability of knowledge learned from $\mathcal{P}_{\text{train}}$ to $\mathcal{P}_{\text{test}}$. We can quantify this domain difference using Maximum Mean Discrepancy (MMD) (Gao et al., 2021a). As illustrated in Figure 9, an increasing number of diverse training prompts cover a wider semantic space, bringing $\mathcal{P}_{\text{train}}$ closer to $\mathcal{P}_{\text{test}}$. This proximity reduces the upper bound of the target prompt distribution.

Generalization Guarantee. By minimizing the empirical risk $\hat{R}_{\mathcal{P}_{\text{train}},N}(f^*)$ across a sufficiently large and varied corpus of prompts, PAFT encourages the model to internalize the underlying task semantics, rather than merely memorizing superficial prompt structures. This principled approach is key to improving the model performance $R_{\mathcal{P}_{\text{test}}}(f^*)$ when confronted with novel and unencountered prompts.

7 Conclusion

PAFT offers a compelling solution for enhancing the prompt robustness of LLMs. By dynamically adjusting prompts during fine-tuning, PAFT significantly improves model generalization and performance across diverse prompt formulations. Notably, PAFT boosts inference speed with maintained training cost. This approach paves the way for more reliable and efficient LLM deployment in real-world applications.

Limitations

In this section, we discuss potential limitations of PAFT and outline promising directions for future research. While PAFT demonstrates significant progress in enhancing the prompt robustness of Large Language Models (LLMs), certain aspects warrant further investigation. A key area for improvement lies in the dynamic prompt selection strategy employed during fine-tuning. Currently, PAFT utilizes a random sampling approach, which, while exposing the model to a diverse range of prompts, may not be the most efficient or effective method. Exploring more sophisticated sampling techniques, such as curriculum learning or importance sampling, could potentially optimize the training process and further enhance robustness. For instance, prioritizing prompts that induce higher loss or those that are more representative of the overall prompt distribution could lead to faster convergence and improved generalization. Furthermore, integrating adversarial learning into the dynamic fine-tuning phase presents a compelling avenue for future work. Generating adversarial prompts on-the-fly, perhaps through gradient-based updates, could further challenge the model and encourage it to learn more robust task representations. This approach could be particularly beneficial in mitigating the impact of maliciously crafted or unexpected prompts. However, the well-known instability of adversarial training remains a significant hurdle. Stabilizing the training process, perhaps through techniques like robust optimization or regularization, is crucial for realizing the full potential of this approach. Investigating different adversarial prompt generation strategies and their impact on model robustness would be a valuable contribution.

Ethics Statement

We have manually reevaluated the dataset we created to ensure it is free of any potential for discrimination, human rights violations, bias, exploitation, and any other ethical concerns.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2006. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems 19*, pages 137–144.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language.

Kaiyan Chang, Songcheng Xu, Chenglong Wang, Yingfeng Luo, Xiaoqian Liu, Tong Xiao, and Jingbo Zhu. 2024. Efficient prompting methods for large language models: A survey.

Tianyi Chen, Tianyu Ding, Badal Yadav, Ilya Zharkov, and Luming Liang. 2023. Lorashear: Efficient large language model structured pruning and knowledge recovery.

Zehui Chen, Weihua Du, Wenwei Zhang, Kuikun Liu, Jiangning Liu, Miao Zheng, Jingming Zhuo, Songyang Zhang, Dahua Lin, Kai Chen, and Feng Zhao. 2024. T-eval: Evaluating the tool utilization capability of large language models step by step. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9510–9529, Bangkok, Thailand. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro

Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems.

OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Shizhe Diao, Zhichao Huang, Ruijia Xu, Xuechun Li, LIN Yong, Xiao Zhou, and Tong Zhang. 2023. Black-box prompt learning for pre-trained language models. *Transactions on Machine Learning Research*.

Ruize Gao, Feng Liu, Jingfeng Zhang, Bo Han, Tongliang Liu, Gang Niu, and Masashi Sugiyama. 2021a. Maximum mean discrepancy test is aware of adversarial attacks.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021b. Making pre-trained language models better few-shot learners. 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 3816–3830, Online. Association for Computational Linguistics.

Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franklin X Wang, and Sadid Hasan. 2024. Does prompt formatting have any impact on llm performance?

Sirui Hong, Yizhang Lin, Bang Liu, Bangbang Liu, Binhao Wu, Ceyao Zhang, Chenxing Wei, Danyang Li, Jiaqi Chen, Jiayi Zhang, Jinlin Wang, Li Zhang, Lingyao Zhang, Min Yang, Mingchen Zhuge, Taicheng Guo, Tuo Zhou, Wei Tao, Xiangru Tang, Xiangtao Lu, Xiawu Zheng, Xinbing Liang, Yaying Fei, Yuheng Cheng, Zhibin Gou, Zongze Xu, and Chenglin Wu. 2024. Data interpreter: An llm agent for data science.

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*.

Wenyang Hu, Yao Shu, Zongmin Yu, Zhaoxuan Wu, Xiaoqiang Lin, Zhongxiang Dai, See-Kiong Ng, and Bryan Kian Hsiang Low. 2024. Localized zeroth-order prompt optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. 2023a. LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5254–5276, Singapore. Association for Computational Linguistics.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. 2023b. LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Yoichi Ishibashi, Danushka Bollegala, Katsuhito Sudoh, and Satoshi Nakamura. 2023. Evaluating the robustness of discrete prompts. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2373–2384, Dubrovnik, Croatia. Association for Computational Linguistics.

Jens Kohl, Luisa Gloger, Rui Costa, Otto Kruse, Manuel P. Luitz, David Katz, Gonzalo Barbeito, Markus Schweier, Ryan French, Jonas Schroeder, Thomas Riedl, Raphael Perri, and Youssef Mostafa. 2024. Generative ai toolkit – a framework for increasing the quality of llm-based applications over their whole life cycle.

Po-Nien Kung and Nanyun Peng. 2023. Do models really learn to follow instructions? an empirical study of instruction tuning.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations.

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.

Yinheng Li. 2023. A practical survey on zero-shot prompt design for in-context learning. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 641–647, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Xiaoqiang Lin, Zhaoxuan Wu, Zhongxiang Dai, Wenyang Hu, Yao Shu, See-Kiong Ng, Patrick Jaillet, and Bryan Kian Hsiang Low. 2024. Use your INSTINCT: INSTruction optimization for LLMs using neural bandits coupled with transformers. In *Forty-first International Conference on Machine Learning*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Meta. 2024. Introducing meta llama 3: The most capable openly available LLM to date. *Meta Blog*.

Grégoire Mialon, Roberto Dessi, Maria Lomeli, Christoforos Nalmpantis, Ramakanth Pasunuru, Roberta Raileanu, Baptiste Roziere, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented language models: a survey. *Transactions on Machine Learning Research*. Survey Certification.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5316–5330, Dublin, Ireland. Association for Computational Linguistics.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya,

Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Felipe Maia Polo, Ronald Xu, Lucas Weber, Mírian Silva, Onkar Bhardwaj, Leshem Choshen, Allysson Flavio Melo de Oliveira, Yuekai Sun, and Mikhail Yurochkin. 2024. Efficient multi-prompt evaluation of llms. *arXiv preprint arXiv:2405.17202*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Mrigank Raman, Pratyush Maini, J Zico Kolter, Zachary Chase Lipton, and Danish Pruthi. 2023. Model-tuning via prompts makes NLP models adversarially robust. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2025. A systematic survey of prompt engineering in large language models: Techniques and applications.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale.

Abel Salinas and Fred Morstatter. 2024. The butterfly effect of altering prompts: How small changes and jailbreaks affect large language model performance.

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*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models.

Zeru Shi, Zhenting Wang, Yongye Su, Weidi Luo, Fan Yang, and Yongfeng Zhang. 2024. Robustness-aware automatic prompt optimization.

Yao Shu, Wenyang Hu, See-Kiong Ng, Bryan Kian Hsiang Low, and Fei Richard Yu. 2024. Ferret: Federated full-parameter tuning at scale for large language models. In *International Workshop on Federated Foundation Models in Conjunction with NeurIPS 2024*.

Chongjie Si, Zhiyi Shi, Shifan Zhang, Xiaokang Yang, Hanspeter Pfister, and Wei Shen. 2024. Unleashing the power of task-specific directions in parameter efficient fine-tuning.

Suvrit Sra, Sebastian Nowozin, and Stephen J Wright. 2011. *Optimization for machine learning*, page 351–368. Mit Press.

Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. 2022. Black-box tuning for language-model-as-a-service. In *Proceedings of ICML*.

Anton Voronov, Lena Wolf, and Max Ryabinin. 2024. Mind your format: Towards consistent evaluation of in-context learning improvements. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6287–6310, Bangkok, Thailand. Association for Computational Linguistics.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2024. Openchat: Advancing open-source language models with mixed-quality data. In *The Twelfth International Conference on Learning Representations*.

Chenxing Wei, Yao Shu, Ying Tiffany He, and Fei Richard Yu. 2024. Flexora: Flexible low-rank adaptation for large language models. In *NeurIPS 2024 Workshop on Fine-Tuning in Modern Machine Learning: Principles and Scalability*.

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*.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment.

Dong Yin, Kannan Ramchandran, and Peter Bartlett. 2020. Rademacher complexity for adversarially robust generalization.

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

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2024. Instruction tuning for large language models: A survey.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2024. A survey of large language models.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

A Experimental setting

In the main experiment, we compared PAFT with the baseline. The datasets and experimental parameters are as follows:

A.1 Dataset

In this section, we introduce the statistics of the dataset. The statistics of the dataset are shown in Table 4.

Table 4: Number of samples in the train, validation, and test datasets for various dateset.

Number of samples	train dataset	validation dataset	test dataset
Hellaswag	39900	10000	10000
PIQA	16000	2000	3000
Winogrande	40398	1267	1767
RACE	87866	4887	4934

A.2 Specific experimental parameters

Based on the LLaMA3-8B model configuration, several adjustments were made to optimize model performance. In the baseline model experiment, generation parameters were adjusted to ensure the correct output. In the LoRA experiment, adjustments to the generation parameters were retained, and LoRA-related parameters were adjusted. In the PAFT experiment, the size of the validation set was adjusted to control the time required to search for the optimal layer. For specific experimental parameters, see the table 5.

Table 5: Detailed experimental parameters. This table lists the specific parameters we used in the experiments for various methods. These parameters include the target module of LoRA (Lora Target), the maximum sequence length (Max Length), the number of samples for supervised fine-tuning (SFT Samples), the learning rate (LR), the number of training prompts (Training Prompts). Epoch(Epoch) represents the epoch of training. All other parameters not listed here remain consistent across all experiments.

Methods	LoRA Target	Max Length	SFT Samples	LR	Training Prompts	Epoch
LoRA	q & v Proj	1024	20000	0.0001	1	3
PAFT	q & v Proj	1024	20000	0.0001	400	3

B Training cost and inference time

B.1 Training cost

PAFT Maintains Training Efficiency. We now turn our attention to the training efficiency of PAFT. A critical consideration for any practical fine-tuning approach is its impact on training time. Introducing complex mechanisms or additional computational overhead can significantly hinder the training process, especially when dealing with large language models and extensive datasets. Therefore, it is essential to demonstrate that PAFT does not introduce such burdens.

To rigorously evaluate the training time implications of PAFT, we conducted a series of experiments, using Low-Rank Adaptation (LoRA) (Hu et al., 2022) as a representative example of a parameter-efficient fine-tuning method. LoRA has gained popularity due to its ability to adapt pre-trained models with minimal computational cost, making it a suitable baseline for our analysis. Our experiments, the results of which are presented in Table 6, directly compare the training time required for traditional LoRA fine-tuning with the training time required for PAFT integrated with LoRA.

The key finding from our analysis is that PAFT does not introduce any noticeable increase in training time. The data in Table 6 clearly demonstrates that the training duration remains virtually identical

Table 6: Training Time Comparison of Different Fine-tuning Methods on the Test Prompt Sets Across Various Reasoning and Reading Comprehension Tasks Using the LLaMA3-8B(Meta, 2024) Model with LoRA Rank 8. Experiments were conducted on an NVIDIA RTX 4090 GPU. Results are reported as training time in hours. **LoRA + TopAccuracy prompt** prompt refers to the prompt with the highest accuracy in the training set, **LoRA + user-specified prompt** (Wei et al., 2024) refers to fine-tuning with human-designed prompts, **LoRA + BATprompt** (Shi et al., 2024) uses the most robust prompt generated by BATprompt, and **LoRA + ZOPO prompt** (Hu et al., 2024) employs the optimal prompt selected by ZOPO from the training prompt set.

Training time/h	**Hellaswag**	**PIQA**	**Winogrande**	**RACE**	**Average**
LoRA + user-specified prompt	3.01	2.35	3.27	3.95	3.15
LoRA + TopAccuracy prompt	3.00	2.29	2.98	3.93	3.05
LoRA + BATprompt	3.02	**2.23**	3	3.93	3.05
LoRA + ZOPO prompt	**2.97**	2.3	**2.97**	3.83	**3.02**
PAFT	2.98	2.32	3.38	**3.81**	3.12

whether we employ standard LoRA or incorporate PAFT's dynamic prompt selection mechanism. This crucial observation underscores the efficiency of PAFT. The dynamic prompt selection process, which is central to PAFT's ability to enhance prompt robustness, is implemented in a way that does not add significant computational overhead. This is because the selection process is lightweight and seamlessly integrated into the existing training loop. Rather than requiring complex computations or extensive data manipulations, PAFT efficiently chooses from a diverse set of prompts, allowing the model to experience a wider range of input formulations without incurring a substantial time penalty. This efficient dynamic prompt selection is critical for the practical applicability of PAFT, ensuring that it can be readily deployed without compromising training efficiency. Furthermore, this efficiency allows for more extensive experimentation and exploration of different prompt variations, ultimately leading to more robust and generalizable models.

Efficient Candidate Prompt Generation. A key aspect of PAFT's effectiveness lies in its ability to generate a diverse and high-quality set of candidate prompts efficiently. The process of constructing these candidate prompts involves leveraging the capabilities of external large language models (LLMs), which naturally raises the question of associated costs. Specifically, we sought to quantify the token usage required for candidate prompt generation, as this directly translates to the expense incurred when interacting with commercial LLM APIs.

To address this, we conducted a detailed analysis of the token consumption during the candidate prompt generation phase of PAFT. Our investigation, the results of which are summarized in Table 1, focuses on the number of tokens required to produce a sufficient variety of prompts suitable for subsequent selection and fine-tuning. We meticulously tracked the token usage across various prompts generated for different tasks, considering factors such as prompt length, complexity, and diversity.

The findings presented in Table 7 demonstrate that PAFT requires remarkably few tokens to generate a substantial pool of candidate prompts. This efficiency stems from PAFT's strategic approach to prompt engineering. Rather than relying on brute-force generation or computationally intensive search methods, PAFT employs a carefully designed prompting strategy that encourages the external LLMs to produce a wide range of prompt formulations with minimal token consumption. This is achieved through techniques such as few-shot prompting with carefully chosen examples, targeted instructions that guide the LLM towards desired prompt characteristics, and potentially iterative refinement of prompts based on preliminary evaluation. The low token count is crucial for practical applications, as it minimizes the cost associated with using commercial LLM APIs. Moreover, this efficiency enables the exploration of a broader range of potential prompts within a fixed budget, increasing the likelihood of discovering highly effective prompts that contribute to improved model robustness. This efficient prompt generation process is a significant advantage of PAFT, enabling it to achieve superior performance without incurring prohibitive costs.

Table 7: Token Usage for Candidate Prompt Generation. This table shows the number of tokens used to generate approximately 400 candidate prompts for each task. The average token usage is 11.75k. The number of generated prompts can be adjusted based on the scaling law observed in Figure 6 to control costs.

Tokens	Hellaswag	PIQA	Winogrande	RACE	Average
Total Tokens	11.7k	12.1k	10.9k	12.3k	11.75k

B.2 Inference Times

In this section, we show some special cases to demonstrate that PAFT improves reasoning efficiency by improving the model's understanding of the core task semantics, enabling concise and accurate responses with fewer tokens.

cases study

```
Hellaswage: {
    "prompt": "By analyzing the setup in A man is sitting on a roof.
    He, deduce which outcome would serve as the best fit:
    A. is using wrap to wrap a pair of skis.\n
    B. is ripping level tiles off.\n
    C. is holding a rubik's cube.\n
    D. starts pulling up roofing on a roof.\n
    You may choose from 'A', 'B', 'C', 'D'.\n Answer:",
    "SFT prediction": "Best answer: D, as "pulling up roofing on a roof" is the most plausible action
    in this context, while the others are either irrelevant or unclear.",
    "PAFT prediction": "D",
    "gold": "D"
},
PIQA: {
    "origin_prompt": "For the situation described by ice box, consider which solution aligns more
    closely with how things usually work in real life:
    A. will turn into a cooler if you add water to it\n
    B. will turn into a cooler if you add soda to it.
    Use logical reasoning to guide your choice. Answer:",
    "SFT prediciton": "The correct answer is A. will turn into a cooler if you add water to it,
    because: A traditional ice box (or cooler) works by using ice (frozen water) to keep items cold.
    Adding water to an ice box would help maintain cooling as the ice melts,
    whereas soda (a sugary, non-freezing liquid) would not improve cooling efficiency.",
    "PAFT prediciton" : "A",
    "gold": "A"
},
Winograde: {
    "origin_prompt": "
    In the context of Sarah was a much better surgeon than
    Maria so _ always got the harder cases.\n
    which word best completes the sentence? Choose:
    A. Sarah\n
    B. Maria\n
    Answer:",
    "SFT prediciton": "Maybe Sarah or Maria I might need more information to answer this question.
    I guess the final answer is B.",
    "PAFT prediciton" : "A",
    "gold": "A"
},
```

C Prompt

In this section, we present a selection of training and test prompts to illustrate the efficacy of our prompt construction algorithm and to provide a clearer understanding of operational process of PAFT. Due to space constraints, we only list 10 prompts as examples. Section C.1 showcases examples of training prompts, Section C.2 highlights test prompts, and Section C.3 outlines the prompts utilized by the baseline method.

C.1 Train prompt

In this section, we present the prompts generated using the method outlined in Section 4.1 across various datasets. All prompts listed here are utilized for training purposes.

Train Prompt of Hellaswag

```
1. Based on the given context {ctx}, which of the following options correctly predicts the outcome?
Choose the correct letter option.\n A. {A}\nB. {B}\nC. {C}\nD. {D}\n Answer:
2. Considering the scenario described in {ctx}, identify the most accurate prediction of the
final result:Select the correct letter.\n A. {A}\nB. {B}\nC. {C}\nD. {D}\n Answer:
3. Given the information in {ctx}, which option best forecasts the correct ending?Provide the
correct letter choice.\n A. {A}\nB. {B}\nC. {C}\nD. {D}\n Answer:
4. From the context {ctx}, which of the following options accurately predicts the conclusion?Write
down the correct letter.\n A. {A}\nB. {B}\nC. {C}\nD. {D}\n Answer:
5. Using the details provided in {ctx}, select the option that correctly predicts the final outcome:
Enter the correct letter.\n A. {A}\nB. {B}\nC. {C}\nD. {D}\n Answer:
6. Based on the context {ctx}, which option is the most accurate prediction of the ending?Choose the
correct letter option.\n A. {A}\nB. {B}\nC. {C}\nD. {D}\n Answer:
7. Given the scenario in {ctx}, identify the option that correctly forecasts the outcome:Select the
correct letter.\n A. {A}\nB. {B}\nC. {C}\nD. {D}\n Answer:
8. Considering the details in {ctx}, which option best predicts the correct conclusion?Provide the
correct letter choice.\n A. {A}\nB. {B}\nC. {C}\nD. {D}\n Answer:
9.Analyze the context {ctx} and determine the correct prediction of the outcome:Indicate the
correct letter.\n A. {A}\nB. {B}\nC. {C}\nD. {D}\n Answer:
10. Analyze the given context {ctx} and determine the most accurate prediction of the final result:
Indicate the correct letter.\n A. {A}\nB. {B}\nC. {C}\nD. {D}\n Answer:
```

Train Prompt of PIQA

```
1.In order to {goal}, which of the following options is the most logical choice based on common
knowledge?\nA. {sol1}\nB. {sol2}\nAnswer:
2.Consider the scenario where you need to {goal}. Which option would be the most appropriate
according to general understanding?\nA. {sol1}\nB. {sol2}\nAnswer:
3.When trying to {goal}, which of the following would be the best course of action based on everyday
reasoning?\nA. {sol1}\nB. {sol2}\nAnswer:
4.To achieve {goal}, which option aligns best with common sense?\nA. {sol1}\nB. {sol2}\nAnswer:
5.Based on typical knowledge, which of the following is the correct choice to {goal}?
\nA. {sol1}\nB. {sol2}\nAnswer:
6.If you want to {goal}, which of these options would be the most sensible according to common
reasoning?\nA. {sol1}\nB. {sol2}\nAnswer:
7.Using general knowledge, determine the best option to {goal}.\nA. {sol1}\nB. {sol2}\nAnswer:
8.To {goal}, which of the following choices is the most reasonable based on common sense?
\nA. {sol1}\nB. {sol2}\nAnswer:
9.When considering how to {goal}, which option would be the most logical based on everyday knowledge?
\nA. {sol1}\nB. {sol2}\nAnswer:
10.According to common reasoning, which of the following is the best way to {goal}?
\nA. {sol1}\nB. {sol2}\nAnswer:
```

Train Prompt of Winogrande

```
1.Choose the correct answer to complete the sentence.{ctx}
\nA. {only_option1}\nB. {only_option2}\nAnswer:
2.elect the appropriate option to fill in the blank.{ctx}
\nA. {only_option1}\nB. {only_option2}\nAnswer:
3.Fill in the blank with the correct answer.{ctx}
\nA. {only_option1}\nB. {only_option2}\nAnswer:
4.Identify the correct choice to complete the statement.{ctx}
\nA. {only_option1}\nB. {only_option2}\nAnswer:
5.Choose the right answer to fill in the gap .{ctx}
\nA. {only_option1}\nB. {only_option2}\nAnswer:
6.Select the correct option to complete the sentence.{ctx}
\nA. {only_option1}\nB. {only_option2}\nAnswer:
7.Fill in the blank with the correct answer.{ctx}
\nA. {only_option1}\nB. {only_option2}\nAnswer:
8.Identify the correct choice to complete the sentence.{ctx}
\nA. {only_option1}\nB. {only_option2}\nAnswer:
9.Choose the right answer to fill in the blank. {ctx}
\nA. {only_option1}\nB. {only_option2}\nAnswer:
10.Select the appropriate option to complete the statement.{ctx}
\nA. {only_option1}\nB. {only_option2}\nAnswer:
```

Train Prompt of RACE

```
1.Carefully read the following article and answer the question by selecting the correct option.
Respond with A, B, C, or D.\n\nArticle:\n{article}\n\n
Q: {question}\n\nA. {A}\nB. {B}\nC. {C}\nD. {D}\nAnswer:
2.Read the passage below and choose the best answer to the question.
Reply with the letter A, B, C, or D.\n\nArticle:\n{article}\n\n
Q: {question}\n\nA. {A}\nB. {B}\nC. {C}\nD. {D}\nAnswer:
3.After reading the article, answer the following question by selecting the correct option.
Please respond with A, B, C, or D.\n\nArticle:\n{article}\n\n
Q: {question}\n\nA. {A}\nB. {B}\nC. {C}\nD. {D}\nAnswer:
4.Examine the article provided and answer the question by choosing the most appropriate option.
Reply with A, B, C, or D.\n\nArticle:\n{article}\n\n
Q: {question}\n\nA. {A}\nB. {B}\nC. {C}\nD. {D}\nAnswer:
5.Read the following text and answer the question by selecting the correct letter.
Respond with A, B, C, or D.\n\nArticle:\n{article}\n\n
Q: {question}\n\nA. {A}\nB. {B}\nC. {C}\nD. {D}\nAnswer:
6.Carefully read the article and choose the best answer to the question.
Reply with A, B, C, or D.\n\nArticle:\n{article}\n\n
Q: {question}\n\nA. {A}\nB. {B}\nC. {C}\nD. {D}\nAnswer:
7.Read the passage and answer the question by selecting the correct option.
Respond with A, B, C, or D.\n\nArticle:\n{article}\n\n
Q: {question}\n\nA. {A}\nB. {B}\nC. {C}\nD. {D}\nAnswer:
8.After reading the article, choose the correct answer to the question.
Reply with A, B, C, or D.\n\nArticle:\n{article}\n\n
Q: {question}\n\nA. {A}\nB. {B}\nC. {C}\nD. {D}\nAnswer:
9.Read the provided text and answer the question by selecting the best option.
Respond with A, B, C, or D.\n\nArticle:\n{article}\n\n
Q: {question}\n\nA. {A}\nB. {B}\nC. {C}\nD. {D}\nAnswer:
10.Examine the article and answer the question by choosing the correct letter.
zReply with A, B, C, or D.\n\nArticle:\n{article}\n\n
Q: {question}\n\nA. {A}\nB. {B}\nC. {C}\nD. {D}\nAnswer:
```

C.2 Test prompt

In this section, we present the prompts generated using the method outlined in Section 4.1 across various datasets. All prompts listed here are utilized for testing purposes, and they are not visible during training.

Test Prompt of Hellaswag

```
1.Based on the information provided, please select the most probable conclusion: {ctx}
\n A. {A}\nB. {B}\nC. {C}\nD. {D}\n
Remember to consider the implications of each option. Answer:
2.In the scenario described by {ctx}, there is only one correct way the story or situation could end.
When predicting the right ending, consider the cause-and-effect relationships established within
the context.An option that logically follows from the preceding events is likely the correct one.
\n A. {A}\nB. {B}\nC. {C}\nD. {D}\n You may choose from 'A', 'B', 'C', 'D'.\n Answer:
3.Based on the given context {ctx}, which of the following options correctly predicts the outcome?
Choose the correct letter option.
\n A. {A}\nB. {B}\nC. {C}\nD. {D}\n Answer:
4.To solve this problem based on {ctx}, weigh the significance of each potential ending:
A. {A}\nB. {B}\nC. {C}\nD. {D}\n You may choose from 'A', 'B', 'C', 'D'.\n Answer:
5.Analyzing the context of {ctx}, think about the relationships and conflicts presented.
Which option is most likely to resolve these issues and lead to a satisfying ending?
\n A. {A}\nB. {B}\nC. {C}\nD. {D}\n Answer:
6.{ctx}\nQuestion: Taking into account the context, which outcome is the most expected?
\n A. {A}\nB. {B}\nC. {C}\nD. {D}\n Answer:
7.From the detailed description provided, choose the option that best completes the scenario:{ctx}\
n A. {A}\nB. {B}\nC. {C}\nD. {D}\n
Consider all aspects of the scenario to make an informed decision on the correct ending.\n Answer:
8.Given the scenario described in {ctx}, which of the following conclusions seems most plausible?
Consider all the details and clues provided to make an informed guess.
\n A. {A}\nB. {B}\nC. {C}\nD. {D}\n Answer:
9.To unlock the hidden treasure in {ctx}, you need to choose the correct key.
Which option will open the treasure chest?
A. {A} B. {B} C. {C} D. {D}\n You may choose from 'A', 'B', 'C', 'D'.\n Answer:
10.{ctx}\nQuestion: Reflecting on the emotional stakes and the structure of the narrative,
which conclusion feels the most genuine?
\n A. {A}\nB. {B}\nC. {C}\nD. {D}\n Answer:
```

Test Prompt of PIQA

```
1.Solve the following single-choice question by using your common sense reasoning skills.
Choose the correct option and reply with the corresponding letter.
\nQuestion: {goal}\nA. {sol1}\nB. {sol2}\nAnswer:
2.For the situation described by {goal}, consider which solution aligns more closely with how things
usually work in real life: A. {sol1}\nB. {sol2}. Use logical reasoning to guide your choice. Answer:
3.Given the context of the question, choose the answer that demonstrates the best common
sense reasoning: {goal}\nA. {sol1}\nB. {sol2}\n Answer format: A/B \nAnswer:
4.In considering the aim set forth in {goal}, visualize the potential consequences of each action
as if you were directly involved. This visualization can help you identify the better choice:\n
Question: {goal}\nA. {sol1}\nB. {sol2}\nAnswer:
5.Which solution fits the goal based on common sense?
{goal}\n A. {sol1}\nB. {sol2}\n Answer format: A/B \nAnswer:
6.Analyze the following scenario and select the answer that reflects logical reasoning: {goal}
\nA. {sol1}\nB. {sol2}\n Answer format: A/B \nAnswer:
7.Identify the most logical outcome for the situation described: {goal} A. {sol1} B. {sol2}
Answer format: A/B Remember, the trick is to apply your general knowledge to the scenario. Answer:
8.According to common reasoning, which of the following is the best way to {goal}?
\nA. {sol1}\nB. {sol2}\nAnswer:
9.Which solution best fits the goal based on your general knowledge? {goal}
\n A. {sol1}\nB. {sol2}\n Answer format: A/B \nAnswer:
10.You are about to answer a question that relies on your understanding of basic logic.
Please respond with A or B to indicate your choice.
\nQuestion: {goal}\nA. {sol1}\nB. {sol2}\nAnswer:
```

Test Prompt of Winogrande

```
1.In the context of {prompt}, which word best completes the sentence?
Choose: A. {only_option1}. B. {only_option2}.\nAnswer:.
2.When analyzing {prompt}, think about the overall theme. What fits best?
A. {only_option1}. B. {only_option2}.\nAnswer:.
3.For {prompt}, consider the emotional tone. Which option resonates more?
A. {only_option1}. B. {only_option2}.\nAnswer:.
4.Reflect on {prompt}. Which word logically fills the gap?
A. {only_option1}. B. {only_option2}.\nAnswer:.
5.In {prompt}, which choice aligns with the preceding ideas?
A. {only_option1}. B. {only_option2}.\nAnswer:.
6.When faced with {prompt}, think about the context. What completes it best?
A. {only_option1}. B. {only_option2}.\nAnswer:.
7.For {prompt}, identify the word that maintains the flow of the sentence.
Choose: A. {only_option1}. B. {only_option2}.\nAnswer:.
8.In the case of {prompt}, which option best conveys the intended meaning?
A. {only_option1}. B. {only_option2}.\nAnswer:.
9.Analyze {prompt} for clues. Which word fits the context?
A. {only_option1}. B. {only_option2}.\nAnswer:.
10.When considering {prompt}, which option enhances the clarity of the statement?
A. {only_option1}. B. {only_option2}.\nAnswer:.
```

Test Prompt of RACE

```
1.After reading the article, analyze the question and choose the best answer
based on the details and themes discussed. Look for clues within the text that
align with one of the options.\nArticle:\n{article}\n\nQuestion:
{question}\nOptions: \nA. {A}\nB. {B}\nC. {C}\nD. {D}\nAnswer:
2.Article:\n{article}\nAfter reading the passage, please answer the following question:
\n{question}\nA. {A}\nB. {B}\nC. {C}\nD. {D} \nAnswer:
3.Carefully read the following article and answer the question by selecting the correct option.
Respond with A, B, C, or D.\n\nArticle:\n{article}\n\n
Q: {question}\n\nA. {A}\nB. {B}\nC. {C}\nD. {D}\nAnswer:
4.Read the text carefully and answer the question by choosing the most appropriate option.
Evaluate the relevance of each choice to the main points discussed.
\nArticle:\n{article}\n\nQuestion: {question}\nOptions: \nA. {A}\nB. {B}\nC. {C}\nD. {D}\nAnswer:
5.Describe the setting of the article.
{question}\n{article}\nA. {A}\nB. {B}\nC. {C}\nD. {D} \nAnswer:
6.While reading the {article}, highlight or make mental notes of significant details.
The {question} is asking [describe the specific query].
Now evaluate the options:\nA. {A}\nB. {B}\nC. {C}\nD. {D} \nAnswer:
7.After carefully analyzing {article}, determine which of the following options best
answers the question:
{question}. A. {A}\nB. {B}\nC. {C}\nD. {D} \nAnswer:
8.Read {article} with a focus on answering {question}. Choose the most suitable option.
Article: {article} Question:{question} Options: A. {A} B. {B} C. {C} D. {D}
Trick: Be cautious of answer choices that seem too extreme. Your answer is just one letter. Answer:
9.Article:\n{article}\nFrom the information in the article, identify the correct
answer to the following question: \n{question}\nA. {A}\nB. {B}\nC. {C}\nD. {D} \nAnswer:
10.When {article} mentions {question}, which option best describes the author's attitude?
\nA. {A}\nB. {B}\nC. {C}\nD. {D} \n// Pay attention to the tone of the author.
Look for words that convey emotions or opinion to determine the attitude.Answer:
```

C.3 Baseline prompt

In this section, we present the best prompts generated or filtered using the baseline for training.

Prompt of Hellaswag

```
TopAccuracy prompt:
Given the context {ctx}, predict the correct ending by choosing the most logical option.
\n A. {A}\nB. {B}\nC. {C}\nD. {D}\n You may choose from 'A', 'B', 'C', 'D'.\n Answer:

User-specified prompt:
{ctx}\n Question: {Question}\n A. {A}\nB. {B}\nC. {C}\nD. {D}\n
You may choose from 'A', 'B', 'C', 'D'.\n Answer:

BATprompt :
Given the context below, predict the most logical ending by choosing the correct option
from the provided choices. Ensure your choice aligns with the context and is the most coherent
conclusion. \n Context: {ctx}\n
Question: Which ending makes the most sense?\n A. {A}\nB. {B}\nC. {C}\nD. {D}\n
You may choose from 'A', 'B', 'C', 'D'.\n Answer:

ZOPO prompt:
Based on {ctx}, which option is the most likely correct ending?
Consider the overall context, character motivations, and any foreshadowing.
Trick: Analyze the consistency of each option with the established details.
A. {A}\nB. {B}\nC. {C}\nD. {D}\n You may choose from 'A', 'B', 'C', 'D'.\n Answer:
```

Prompt of PIQA

```
TopAccuracy prompt:
Use both common sense and logical reasoning to determine the correct solution for the goal:
{goal}\n A. {sol1}\nB. {sol2}\n Answer format: A/B \nAnswer:

User-specified prompt:
There is a single choice question. Answer the question by replying A or B.'\n
Question: {goal}\nA. {sol1}\nB. {sol2}\nAnswer:

BATprompt :
You should use both common sense and logical reasoning to determine the most appropriate
solution for the following goal. Carefully evaluate the provided options and choose the
one that best aligns with the goal. Goal: {goal}\nA. {sol1}\nB. {sol2}\nAnswer:

ZOPO prompt:
To solve this common sense reasoning question, consider which of the two options seems
more plausible based on everyday knowledge and logic.
\nQuestion: {goal}\nA. {sol1}\nB. {sol2}\n
Think about the practical implications of each choice to determine the correct answer.\nAnswer:
```

Prompt of Winogrande

```
TopAccuracy prompt:
Question: {prompt}\nA. {only_option1}\nB. {only_option2}\nAnswer:

User-specified prompt:
There is a single choice question,  you need to choose the correct option to fill in the blank.
Answer the question by replying A or B.\n
Question:{prompt}\nA. {only_option1}\nB. {only_option2}\nAnswer:

BATprompt :
Complete the following sentence by selecting the most contextually appropriate option.
Carefully consider the meaning and context of the sentence to make your choice.
Question: {prompt}\nA. {only_option1}\nB. {only_option2}\nAnswer:

ZOPO prompt:
Question: Choose the correct modal verb: {prompt}\nA. {only_option1}\nB. {only_option2}\nAnswer:.
```

Prompt of RACE

```
TopAccuracy prompt:
Read the following article carefully: {article}. After reading, answer the question: {question}.
Choose the correct option from the choices provided:
\nA. {A}\nB. {B}\nC. {C}\nD. {D} \n
Trick: Focus on the main idea and supporting details in the article.
Output: Only the letter of the correct answer.\nAnswer:

User-specified prompt:
Article:\n{article}\nQuestion:\n{question}\nA. {A}\nB. {B}\nC. {C}\nD. {D} \nAnswer:

BATprompt :
Please read the passage carefully, focusing on the main ideas and supporting details.
Answer the question that follows by choosing the best option from the choices provided.
Ensure your response is based solely on the information in the passage. Output only the
letter of the correct answer. Article:\n{article}
\nQuestion:\n{question}\nA. {A}\nB. {B}\nC. {C}\nD. {D} \nAnswer:

ZOPO prompt:
A reading comprehension question is before you. Read the article and answer the question
by selecting A, B, C, or D.\n\nArticle:\n{article}\n\n
Q: {question}\n\nA. {A}\nB. {B}\nC. {C}\nD. {D}\nAnswer:
```