Self-Renewal Prompt Optimizing with Implicit Reasoning

Zihan Liang[1,*] **Ben Chen**[1,*,†] **Zhuoran Ran**[1,3,*] **Zihan Wang**[1,*]
HuangYu Dai[1] **Yufei Ma**[2] **Dehong Gao**[2,4] **Xiaoyan Cai**[3] **Libin Yang**[2]
[1]Alibaba Group, Hangzhou, China
[2]Northwestern Polytechnical University, School of Cybersecurity, Xi'an, China
[3]Northwestern Polytechnical University, School of Automation, Xi'an, China
[4]Binjiang Institute of Artificial Intelligence, ZJUT, Hangzhou, China

Abstract

The effectiveness of Large Language Models (LLMs) relies on their capacity to understand instructions and generate human-like responses. However, aligning LLMs with complex human preferences remains a significant challenge due to the potential misinterpretation of user prompts. Current methods for aligning LLM behaviors fall into two categories: output optimization (such as RLHF, RLAIF, and DPO) and input optimization (like OPRO and BPO). While both approaches aim to guide LLMs towards generating responses that align with desired objectives, the labor-intensive and intentions-inconsistent data annotation, as well as the strict and tedious training supervision, make them struggle to yield optimal results across all models. To address these shortcomings, we introduce a novel self-renewal approach called Prompt Optimization with Implicit Reasoning (POIR). It consists of two key components: 1) a model-specific and self-recirculating data collection method that leverages self-evaluation to enhance prompts in accordance with the model's intrinsic logits, and 2) a prompt rewrite schema that injects implicit reasoning for direct preference learning. Through self-renewal optimization, POIR refines LLM outputs to better align with human preferences across various LLMs and tasks, without relying on supervised fine-tuning. Extensive experiments on a range of LLMs and tasks demonstrate POIR's superior performance. We believe this advancement offers a novel paradigm for developing LLMs that are more attuned to user intentions.

1 Introduction

Recent advancements in Natural Language Processing (NLP) have been primarily driven by the development of Large Language Models (LLMs) (Chowdhery et al., 2023; Zeng et al., 2022; Brown et al., 2020; Touvron et al., 2023; Zhang et al., 2022). Research has shown that LLMs exhibit a remarkable ability to understand and follow instructions (Zhao et al., 2023), leading to the emergence of influential applications like ChatGPT (Achiam et al., 2023). However, aligning LLMs with human preferences remains a complex challenge due to potential discrepancies between the intended meaning of user prompts and the LLMs' interpretation. Addressing this issue requires a nuanced understanding of the interplay between human language, context, and the limitations of models, which is essential for realizing the full potential of LLMs and ensuring their outputs align with users' objectives and values.

Common methods for guiding LLMs to align with human preferences can be broadly categorized into two approaches. The first focuses on output alignment, which involves supervised fine-tuning (SFT) followed by reinforcement learning such as RLHF (Ouyang et al., 2022), RLAIF (Lee et al., 2023; Bai et al., 2022), and DPO (Rafailov et al., 2024). These methods aim to optimize model generation by leveraging preference-laden paired data, typically sourced from experts and LLMs. However, this process is labor-intensive and struggles to maintain consistency due to differing annotator focuses, potentially confusing the LLMs during training and hindering the alignment process.

The second approach, typically like OPRO, PromptAgent, and BPO (Yang et al., 2023a; Wang et al., 2024; Cheng et al., 2023), focuses on input alignment through prompt optimization. By refining input prompts, incorporating additional context, and reformulating illogical or ambiguous components, these methods try to bridge the gap between user intentions and LLMs' interpretations. However, most of them require multiple rounds of prompt rewriting, which is time-consuming during the inference (Ye et al., 2023; Guo et al., 2023; Pryzant et al., 2023). Moreover, OPRO (Yang

[*]Equal Contribution.
[†]Corresponding Author.

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 3030–3041

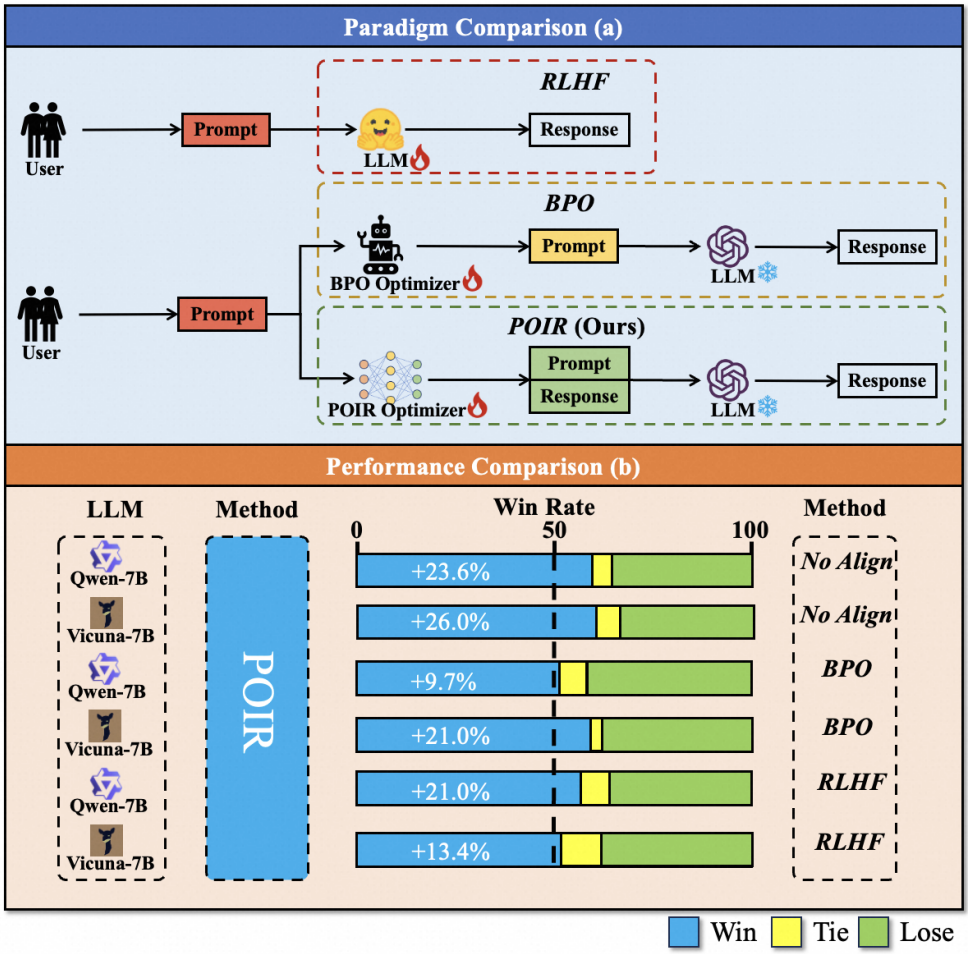

Figure 1: Comparisons of three distinct approaches (RLHF, BPO, and POIR) for aligning LLMs with human preferences. On various models, our POIR facilitates more effective prompt optimization and leads to a better response.

et al., 2023a) needs the task-specific searches, fixed rewriting prompts, and model-specific optimization, which hinder its generalizability and adaptability across diverse tasks and language models. BPO (Cheng et al., 2023) uses GPT-4 to rewrite the original prompts, and train one specific model to generate optimized prompts. Though simple and easy to use, its generalization to unseen text is limited due to excessive rigid supervision. Furthermore, a single general optimized prompt may not produce optimal results across all models.

To address these, here we introduce a novel self-renewal approach called Prompt Optimization with Implicit Reasoning (POIR). It contains two components: 1) a model-specific and self-recirculating data collection method that leverages self-evaluation to enhance prompts, aligning them by the model's intrinsic logits. and 2) a prompt rewrite schema that injects implicit reasoning for direct preference learning. Specifically, we employ Direct Preference Optimization (DPO) (Rafailov et al., 2024), instructing the model to output both optimized prompts and corresponding responses. This design enables the model to implicitly reason about the prompt-response relationship, facilitating more effective prompt optimization.

POIR addresses the limitations of BPO by employing a model-specific data collection method that generates optimized prompts tailored to different model families. Moreover, POIR employs DPO instead of SFT, reducing the need for high-quality annotated data. By injecting implicit reasoning, POIR facilitates more effective prompt optimization and encourages a deeper understanding. Extensive experiments across five models and four tasks demonstrate POIR's superior performance, with win rates increasing by 9.2% to 36.2% compared to models using BPO and DPO. Notably, the effect

increases with model size, with POIR's win rate increasing by 9.2% on the Qwen-7B model and 25.6% on the 14B model compared to BPO.

2 Related Work

Aligning LLMs with human preferences has been a focus of extensive research. Two main approaches have emerged: output alignment, which employs SFT followed by RLHF to optimize model outputs, and input alignment, which focuses on rewriting input prompts. This section provides a detailed discussion of these methods.

2.1 Output Alignment

SFT followed by RLHF is an active area of research aimed at improving final responses by aligning outputs. During the SFT stage, LLMs are trained on high-quality datasets to strengthen their foundational instruction-following abilities (Achiam et al., 2023). Then RLHF is employed to incorporate scalable human feedback into the instruction process. One classic approach (Ouyang et al., 2022) trains a reward model using paired preference data and subsequently instructs LLMs based on this learned critic using the Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017). Despite its effectiveness in aligning LLMs with human intents, this approach faces several challenges, including the scarcity of high-quality preference data and difficulties in ensuring preference consistency due to annotator variations (Lee et al., 2023; Rafailov et al., 2024). Reinforcement Learning from AI Feedback (RLAIF) addresses the reliance on human annotation by using larger LLMs (Chowdhery et al., 2023) to judge preferences (Lee et al., 2023). Direct Preference Optimization (DPO)(Rafailov et al., 2024) proposes an intuitive contrastive loss for directly training LLMs, eliminating the need for a reward model and reducing training complexity and cost. However, the quality of the generated text is intrinsically constrained by the quality of input prompt, emphasizing the need for further research into enhancing prompt efficacy and robustness.

2.2 Input Alignment

Prompt optimization aims to enhance performance by rewriting prompts to better align with the underlying human intent. OPRO (Yang et al., 2023a) leverages LLMs to iteratively generate new prompts by learning from the optimization trajectory of previously explored prompts. Similarly, ProTeGi (Pryzant et al., 2023) employs LLMs to generate "textual gradients" that identify and rectify prompt deficiencies by editing prompts in semantically opposite directions. EVOPROMPT (Guo et al., 2023) utilizes evolutionary algorithms and differential evolution to guide LLMs in evolving prompt populations via evolution operators. TRAN (Yang et al., 2023b) focuses on generating rules from observed errors to establish a rule set that prevents recurrent mistakes in LLM outputs. APE (Zhou et al., 2022) advances prompt optimization by assessing prompts on limited datasets and refining them based on the resulting feedback. Despite these advancements, the multiple iterations of prompt revision and dependency on additional data render these methods time-intensive, constraining their efficiency and practicality, especially in scenarios demanding swift inference.

The most recent method, BPO (Cheng et al., 2023), tries to address this by training a seq2seq model. It first uses GPT-4 to rewrite prompts with given responses from multiple publicly available preference datasets, then employs supervised fine-tuning (SFT) to train a LLaMA-7B-Chat model as a prompt optimizer, generating optimized prompts using only the original prompt as input. However, BPO has several limitations. Firstly, training data is derived from multiple preference datasets with potentially conflicting preferences. Secondly, SFT requires high-quality annotated data, which can be costly to obtain through manual annotation or using advanced models like GPT-4. Moreover, various LLMs are trained on data with different distributions, and a single general optimized prompt may not produce optimal results across all model families.

3 Method

The objective of prompt optimization is to construct a mapping that transitions the original prompt to its optimized version. BPO (Cheng et al., 2023) trains a seq2seq prompt optimizer through SFT. However, as mentioned earlier, SFT has some drawbacks, including its reliance on high-quality data and the use of a strict cross-entropy (CE) loss that supervises the generation of LLMs on a token-by-token basis. This approach introduces a susceptibility to overfitting.

For instance, an optimized prompt like "What is the Pythagorean theorem?" holds the same intent as "Please provide a detailed explanation of

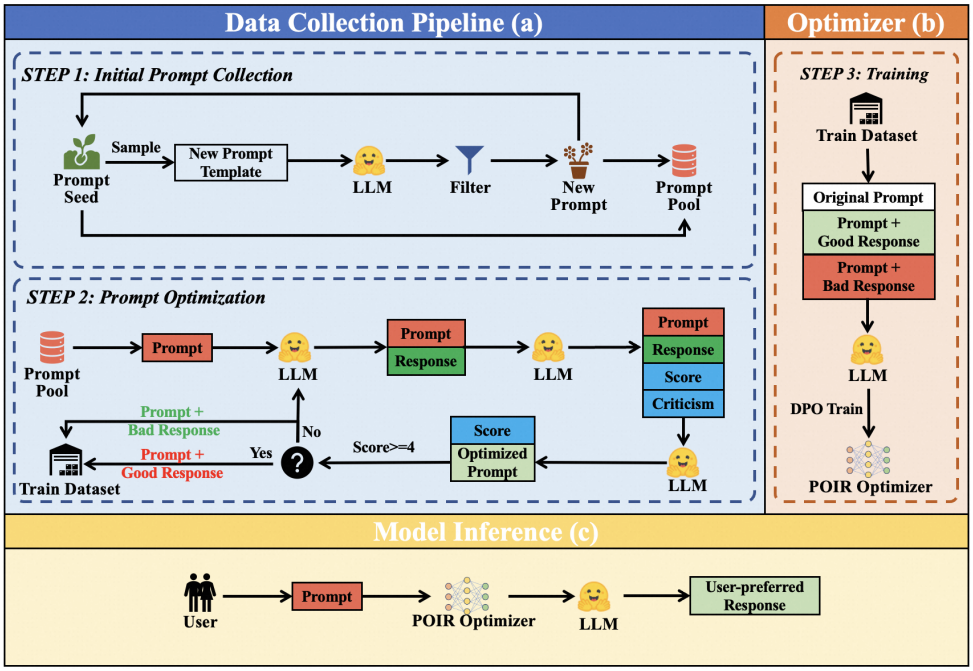

Figure 2: An overview of POIR. The data collection pipeline includes: (1) Generate diverse prompts through initial prompt seeds; (2) Optimize prompts through iterative prompt optimization and self-evaluation by the LLM; (3) Construct preference data by pairing high-scoring and low-scoring optimized prompts. In the training stage, a prompt optimizer is trained by DPO for generating an optimized {prompt, response} pair given the original prompt. During the inference stage, the prompt optimizer is used to generate optimized prompts that can lead to better-aligned responses.

the Pythagorean theorem." Despite their semantic equivalence, the rigid supervision might lead to a substantial loss for such variations due to its emphasis on matching specific token sequences. This overfitting problem may restrict the flexibility of the model in handling diversely expressed prompts, detracting from the adaptability and utility of the optimized prompts in more generalized applications. As a result, this may potentially constrain the model's overall effectiveness.

To address these limitations, POIR employs DPO (Rafailov et al., 2024) to train the prompt optimizer, thereby mitigating the issues caused by SFT. The comprehensive workflow of POIR is illustrated in Figure 2. We first leverage the model's inherent capabilities for prompt generation, response evaluation, and prompt optimization to produce paired data for the upcoming prompt optimizer training. This pipeline is delineated in Section 3.1. By utilizing the model's inherent capabilities, POIR reduces its reliance on expensive and potentially inconsistent annotated data, addressing another limitation of existing approaches.

After acquiring these data, we employ DPO with an implicit reasoning approach to train the prompt optimizer, as detailed in Section 3.2. This approach enables the model to implicitly reason about the relationship between the prompt and the response, facilitating more effective prompt optimization without the need for strict token-level supervision. By incorporating the model's inherent capabilities, POIR aims to provide a more robust and adaptable approach to prompt optimization that can effectively handle diverse prompts and align with human preferences across a wide range of model families and tasks.

3.1 Data Collection Pipeline

As illustrated in Figure 2, our data collection stage consists of two steps to obtain a large amount of prompt optimization preference data, starting from a set of seed prompts. It is worth noting that the entire process relies solely on the model's self-evaluation capabilities, without any assistance from experts like GPT-4. This approach significantly reduces costs and ensures consistency in preference pairs and alignment between the data source and the model being trained, potentially improving the efficacy of model training and enhancing

overall performance. The time and computational resources required for this process is provided in the appendix.

In the first step, we utilize a small number of seed prompts to generate a large quantity of new prompts. Specifically, using meta-learning approach, we randomly extract four prompts from the seed repository each time and employ few-shot learning to generate new prompts, which are then added back to the seed repository. This process is iteratively repeated until a sufficient amount of data is obtained. To address potential issues such as repetition and prohibited content, we apply a filter after generation. The filter ensures data quality and compliance by setting length thresholds to eliminate excessively long or short entries and employs keyword matching to remove inappropriate content. Additionally, the ROUGE-L metric is used to filter out highly similar data.

In the second step, we inject the large number of new prompts obtained from the first step into the self-evaluation pipeline, which is used to collect paired data for training. First, we input the original prompt (P_{ori}) into the model and obtain a response. Then, the model critiques its own response using a multidimensional evaluation criterion encompassing accuracy, completeness, and safety, assigning scores ranging from 1 to 5. The evaluation is conducted by the model itself based on the given criteria, leveraging its inherent capabilities without any assistance from experts like GPT-4. Next, based on the criticism and original prompt, the model generates an optimized prompt (P_{opt}). This loop continues until the responses achieve a score of 4/5 or reach three rounds. Finally, we parse the good responses (R_{good}, scores of 4 or 5), bad responses (R_{bad}, scores of 1, 2, or 3), and their corresponding prompts (P_{good}, P_{bad}). This procedure yields a sizable paired dataset for further training, with each sample represented as (P_{ori}, P_{good}, R_{good}, P_{bad}, R_{bad}), where P_{ori} stands for the original prompt.

The prompt templates used by the LLM throughout the above process, as well as the collected data samples, are presented in the appendix. By leveraging the model's self-evaluation capabilities and iteratively generating and filtering prompts, POIR efficiently collects a large amount of high-quality, model-specific preference data without relying on expensive expert annotations. This approach addresses the limitations of existing methods, such as the reliance on costly and potentially inconsistent annotated data, while ensuring data uniformity and potentially improving training efficacy and overall performance.

3.2 Prompt Optimization with Implicit Reasoning

Leveraging the preference data obtained from Section 3.1, we employ Direct Preference Optimization (DPO) for prompt optimization learning. In contrast to the tightly controlled SFT, DPO adopts a preference-focused strategy that encourages a wider range of prompt expressions, addressing the limitations of SFT's strict token-level supervision. To enhance the model's adeptness at learning optimization preferences, we design a preference training paradigm that simultaneously considers both the optimized prompt and its corresponding response. This dual-focus strategy facilitates prompt optimization via implicit reasoning mechanisms, which is essentially a form of implicit chains of thought (CoT) (Wei et al., 2022). During training stage, we require the prompt optimizer to generate not only the optimized prompt but also its corresponding answer. And during inference, we specifically output only the rewritten prompt, terminating generation at a predetermined token to prevent the LLM from generating an answer. Implicit reasoning enables the model to implicitly reason about the relationship between the prompt and its associated response, facilitating more effective prompt optimization. Formally, the objective of implicit reasoning is to maximize the joint probability $P(P_{\text{opt}}, A|P_{\text{ori}})$, which represents the likelihood of obtaining both the optimized prompt P_{opt} and its corresponding answer A, given the original prompt P_{ori}. This joint probability can be decomposed as follows:

$$P(P_{\text{opt}}, A|P_{\text{ori}}) = P(P_{\text{opt}}|P_{\text{ori}})P(A|P_{\text{opt}}, P_{\text{ori}}) \quad (1)$$

This formulation indicates that implicit reasoning optimizes the original prompt by simultaneously considering $P(P_{\text{opt}}|P_{\text{ori}})$ and $P(A|P_{\text{opt}}, P_{\text{ori}})$. $P(P_{\text{opt}}|P_{\text{ori}})$ denotes the probability of the optimized prompt given the original prompt, while $P(A|P_{\text{opt}}, P_{\text{ori}})$ represents the probability of generating answer A given both the original and optimized prompts.

To maximize $P(P_{\text{opt}}, A|P_{\text{ori}})$, the model must consider both the quality of the optimized prompt and the quality of the corresponding answer A during prompt rewriting. This encourages the model to generate higher-quality prompts that lead to better

Model	Index	Method		Vicuna Eval			Dolly Eval			BPO-test Eval			Self-Instruct Eval			ΔWR
		A	B	A Win	Tie	B Win	A Win	Tie	B Win	A Win	Tie	B Win	A Win	Tie	B Win	
Vicuna-13B-chat	1	POIR_V	NAN	**64.1**	4.5	31.3	**70.2**	3.0	26.8	**55.8**	4.0	40.2	**64.3**	2.5	34.2	+30.5
	2	POIR_V	BPO	**56.5**	5.0	38.5	**59.5**	6.0	34.5	**53.2**	4.8	42.0	**64.7**	3.8	32.5	+21.6
	3	POIR_V	DPO_awo	**56.0**	7.1	36.9	**57.5**	9.5	33.0	**52.4**	7.4	40.2	**50.8**	7.4	41.8	+16.2
Qwen-14B-chat	4	POIR_Q	NAN	**66.7**	8.0	25.3	**70.9**	4.0	25.1	**54.4**	6.4	39.2	**69.2**	3.9	26.9	+36.2
	5	POIR_Q	BPO	**56.9**	6.0	37.1	**56.3**	9.0	34.7	**70.9**	4.0	25.1	**54.4**	6.4	39.2	+25.6
	6	POIR_Q	DPO_awo	**59.3**	9.0	31.7	**66.5**	5.0	28.5	**48.0**	6.8	45.2	**55.0**	3.7	41.3	+20.5
Vicuna-7B-chat	7	POIR_V	NAN	**60.0**	6.0	34.0	**52.8**	3.0	44.2	**52.4**	4.4	43.3	**53.8**	6.2	40.0	+14.4
	8	POIR_V	BPO	**59.0**	3.0	38.0	**56.4**	5.1	38.5	**54.0**	3.5	42.5	**51.2**	1.3	47.5	+13.5
	9	POIR_V	DPO_awo	**51.6**	10.2	38.2	**50.3**	12.3	34.4	**50.0**	9.6	40.4	**50.5**	10.8	38.7	+10.7
Qwen-7B-chat	10	POIR_Q	NAN	**59.3**	5.0	35.7	**55.5**	1.0	43.5	**58.0**	4.0	38.0	**63.1**	1.0	35.9	+21.0
	11	POIR_Q	BPO	**51.4**	6.9	41.7	**47.0**	7.0	46.0	**47.3**	4.3	37.4	**56.5**	3.0	40.5	+9.2
	12	POIR_Q	DPO_awo	**56.9**	7.2	35.9	**54.0**	8.3	34.7	**51.8**	5.2	43.0	**47.8**	13.3	38.9	+14.5
Mistral-7B	13	POIR_M	NAN	**53.9**	5.6	40.6	**51.5**	3.0	45.5	**54.1**	1.9	44.0	**57.0**	1.2	41.8	+11.2
	14	POIR_M	BPO	**50.3**	8.1	41.6	**50.0**	3.5	46.5	**50.0**	3.5	46.5	**51.2**	2.5	46.3	+5.2
	15	POIR_M	DPO_awo	**53.2**	7.4	39.4	**58.5**	9.0	32.5	**51.3**	7.3	41.4	**55.9**	8.0	33.1	+18.1

Table 1: GPT-4's evaluations compare POIR-aligned, DPO of "align with output," BPO of "align with input," and original LLMs without alignment (**ΔWR** denotes the change in win rate compared to the baseline).

answers, addressing the limitations of existing approaches that focus solely on prompt optimization without considering the resulting response quality.

We believe that implicit reasoning and reminiscent of CoT, has the potential to significantly enhance the performance of preference learning by facilitating a more comprehensive understanding of the relationship between prompts and their corresponding responses. This approach goes beyond the strict token-level supervision of SFT and enables the model to generate more effective optimized prompts that lead to higher-quality answers.

In the following section, we conduct extensive experiments to demonstrate the effectiveness of implicit reasoning in enhancing prompt optimization and improving the overall performance of LLMs in various tasks and settings.

4 Experiments

4.1 Experiment Setup

Baseline. To verify the efficiency of proposed method, we conducted extensive experiments on three model families (Qwen (Bai et al., 2023), Vicuna (Chiang et al., 2023), and Mixtral (Jiang et al., 2024)). There are five models in total: Qwen-7B/14B, Vicuna-7B/14B, and Mistral-7B, to evaluate the generalizability across diverse model architectures and scales. For each model, we will compare three methods, as the "align with output" method DPO, the most recent "align with input" method BPO, and POIR.

Dataset. We utilized four datasets in our evaluation: (1) Dolly Eval, a subset of 200 instances randomly sampled from the human-generated Dolly dataset (Conover et al., 2023), which encompasses 8 task categories; (2) Vicuna Eval (Chiang et al., 2023), containing 80 diverse questions across 8 categories; (3) Self-Instruct Eval (Wang et al., 2022), a human evaluation dataset with 252 expert-written, user-oriented instructions motivated by real-world applications; and (4) BPO-test Eval (Cheng et al., 2023), a 200-sample split from the datasets used to construct our training set. To evaluate the models' output, we employed a pairwise scoring (win rate) setup using powerful GPT-4 as evaluators instead of our own models to avoid potential bias towards answers generated by models from the same source. (Cheng et al., 2023; Zheng et al., 2024). The version and scoring prompts for GPT-4 were adapted from MT-bench (Zheng et al., 2024), as shown in Appendix. To mitigate position bias and reduce cost, we randomly shuffled the models' responses in each evaluation.

Implementation Details. For the first data collection stage, each model independently generated 10,000 preference data, with its intrinsic self-recirculating and evaluation capabilities. During the training phase, for all DPO training, the

Model	Index	Method		Vicuna Eval			Dolly Eval			BPO-test Eval			Self-Instruct Eval			ΔWR
		A	B	A Win	Tie	B Win	A Win	Tie	B Win	A Win	Tie	B Win	A Win	Tie	B Win	
Vicuna-13B-chat	1	POIR_V	DPO_awi	**48.2**	5.6	46.2	**47.5**	8.0	44.5	**54.2**	8.9	36.9	**58.2**	3.8	40.0	+10.1
	2	POIR_V	POIR_Q	**61.3**	8.0	30.7	**60.0**	9.5	30.0	**57.5**	7.5	34.9	**55.0**	2.5	42.5	+23.9
	3	POIR_V	SFT	**60.0**	7.0	33.0	**66.5**	7.5	26.0	**63.5**	3.2	33.3	**52.5**	3.7	43.8	+26.6
Qwen-14B-chat	4	POIR_Q	DPO_awi	**49.7**	9.2	41.1	**46.9**	10.6	43.4	**51.5**	7.0	41.5	**54.4**	5.1	40.5	+9.0
	5	POIR_Q	POIR_V	**54.3**	6.0	39.7	**53.0**	10.0	37.0	**53.8**	2.4	43.8	**49.8**	9.2	41.0	+12.4
	6	POIR_Q	SFT	**59.5**	9.0	31.5	**52.5**	8.0	39.5	**50.0**	4.0	46.0	**47.5**	7.5	45.0	+11.9
Vicuna-7B-chat	7	POIR_V	DPO_awi	**51.8**	8.0	40.2	**51.8**	8.0	40.2	**51.8**	8.0	40.2	**51.8**	8.0	40.2	+11.6
	8	POIR_V	POIR_Q	**58.5**	7.5	35.0	**58.0**	4.0	38.0	**65.9**	9.1	25.0	**60.6**	11.4	28.0	+29.3
	9	POIR_V	SFT	**59.8**	7.0	33.2	**62.5**	9.0	28.5	**55.4**	5.4	39.3	**56.3**	3.8	40.0	+23.3
Qwen-7B-chat	10	POIR_Q	DPO_awi	**56.8**	8.5	34.7	**47.0**	12.5	40.5	**53.1**	7.7	39.2	**56.0**	8.0	36.0	+15.6
	11	POIR_Q	POIR_V	**55.3**	6.5	38.2	**53.5**	9.0	37.5	**58.2**	13.6	28.2	**52.3**	5.1	42.6	+18.2
	12	POIR_Q	SFT	**50.9**	7.0	42.1	**50.7**	10.6	38.7	**52.9**	8.7	38.4	**55.3**	7.3	37.4	+13.3
Mistral-7B	13	POIR_M	DPO_awi	**47.2**	7.8	45.0	**50.0**	5.5	44.5	**53.0**	5.5	41.5	**62.3**	6.5	31.2	+12.6
	14	POIR_M	SFT	**52.5**	7.0	40.5	**47.5**	7.5	45.0	**57.1**	5.2	37.8	**61.3**	3.7	35.0	+15.0

Table 2: GPT-4's evaluations compare POIR-aligned models, DPO of "align with input", SFT of "align with input", and POIR-aligned models with different source data (**ΔWR** denotes the change in win rate compared to the baseline).

TRL library(von Werra et al., 2020) and DeepSpeed(Rasley et al., 2020) was deployed. A learning rate of 1e-6 and a β value of 0.4 were set. Training was conducted with a batch size of 1 per GPU for 1 epoch. For all SFT training, it was executed using DeepSpeed with a learning rate of 2e-5 and the training was conducted with a batch size of 4 per GPU for 3 epoch. All experiments were carried out on 8 NVIDIA A800 GPUs. For BPO, we directly used published model (7B)[1], as the authors recognize it as a universal black-box method for various models. Regarding RLHF, in line with the best practices from leading studies on the llm-leaderboard[2], all models underwent training leveraging the ultra-feedback (Cui et al., 2023) and orca (Mukherjee et al., 2023) datasets.

4.2 Overall Result

The win rates of response quality between different models and methods are shown in Table 1, where NAN indicates that the original prompt is sent to the untrained model for inference. $\text{POIR}_{V/Q/M}$ means the optimizer was trained on the data collected by Vicuna, Qwen or Mistral, and the model input is the optimized prompt while generating response. In detail, the leftest column represents the untrained model used directly to generate responses given the optimized prompts (POIR and BPO) or the original prompts (NAN and DPO).

We can find that POIR significantly outperformed all compared methods across different models and datasets. Taking Qwen-14B-chat as an example (index 4), compared to the original prompts, POIR achieved a win rate of 66.7%, 70.9%, 54.4%, and 69.2% on the Vicuna, Dolly, BPO-test, and self-instruction eval sets respectively, demonstrating the effectiveness of POIR in enhancing the quality and alignment of the model's responses.

Specifically, POIR outperformed DPO, which aligned output, with win rate improvements ranging from 10.2% to 20.5% (indices 3, 6, 9, 12, 15). Moreover, POIR surpassed BPO, the current SOTA prompt optimization method, across all models, with improvements ranging from 5.2% to 25.6% (indices 2, 5, 8, 11, 14). These results indicate the superiority of POIR compared to existing methods, and the performance gain becomes more pronounced as the model size increases.

4.3 Ablation Study

As shown in Table 2, we conducted experiment to investigate the impact of three key innovations in POIR: homogeneous data, implicit reasoning, and the choice of training method (DPO vs. SFT). These innovations constitute the primary advance-

[1] https://huggingface.co/THUDM/BPO

[2] https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

Model	Method		Vicuna Eval			Dolly Eval			BPO-test Eval			Self-Instruct Eval			ΔWR
	A	B	A Win	Tie	B Win	A Win	Tie	B Win	A Win	Tie	B Win	A Win	Tie	B Win	
GPT-3.5-turbo	POIR_S1	ori.	**57.9**	6.1	36.0	**64.8**	5.2	30.0	**55.2**	10.0	34.8	**55.2**	10.0	34.8	+25.7
GPT-4	POIR_S1	ori.	**65.2**	5.8	29.0	**70.4**	6.6	23.0	**62.0**	11.0	27.0	**62.0**	11.0	27.0	+39.5
vicuna-13B-1.3v	POIR_S1	ori.	**56.2**	7.3	36.5	**59.0**	8.5	32.5	**52.5**	8.0	39.5	**52.5**	8.0	39.5	+19.7
Qwen-14B-chat	POIR_S1	ori.	**53.6**	10.5	35.9	**57.4**	10.2	32.4	**50.5**	9.5	40.0	**50.5**	9.5	40.0	+17.7

Table 3: Win rates between POIR-aligned, BPO-aligned and original LLM, evaluated by GPT-4. ("POIR S1" denotes results obtained from stage 1, “ori.” denotes “original”, and “WR” denotes “win rates”).

ments of POIR compared to existing methodologies.

Firstly, the effect of implicit reasoning was assessed by comparing the performance of models trained with and without implicit reasoning. In the setting without implicit reasoning, the model was trained to generate only the optimized prompt without producing the corresponding response. The experimental results, corresponding to indices 1, 4, 7, 10, and 13 in Table 2, demonstrate the importance of implicit reasoning. Secondly, to evaluate the importance of homogeneous data, we trained models on non-homologous data, which was generated from different source models. The results, corresponding to indices 2, 5, 8, and 11, suggest that models trained on non-homologous data exhibited lower performance compared to those on homologous data. This finding highlights the significance of maintaining consistency in data distribution for optimal model performance. Thirdly, We conducted a comparative analysis between models aligned with POIR and those aligned with SFT. The outcomes, as illustrated in indices 3, 6, 9, 12, and 14, underscore the efficacy of our methodology in alignment.

To evaluate the effectiveness of POIR beyond the GPT-4 evaluation paradigm, we conducted experiments on widely used benchmarks, including HellaSwag (Zellers et al., 2019), ARC (Clark et al., 2018), and MMLU (Hendrycks et al., 2020). These benchmarks play a crucial role in advancing LLM by challenging models to perform deeper reasoning and understanding across a variety of contexts. As shown in Table 4, POIR improved the model's performance on all these benchmarks compared to both the original models (NAN) and the BPO-aligned models.

Furthermore, we tested the effectiveness of POIR on black-box models, specifically GPT-3.5-turbo and GPT-4(Achiam et al., 2023). Since these models cannot be directly trained, we evaluated the performance of POIR's stage 1 alone, which leverages the model's inherent capabilities for prompt optimization. As shown in Table 3, POIR's stage 1 significantly improved the performance of these black-box models across all evaluation datasets. For GPT-3.5-turbo, POIR's stage 1 increased the win rate by 25.7 percentage points compared to the original prompts. Similarly, for GPT-4, POIR's stage 1 boosted the win rate by an impressive 39.5 percentage points.

Model	Vicuna-7B-chat			Qwen-7B-chat			Mistral-7B		
	NAN	BPO	POIR	NAN	BPO	POIR	NAN	BPO	POIR
MMLU (5)	49.9	50.3	**52.4**	56.9	**58.7**	**58.7**	59.1	60.3	**60.6**
ARC-C (25)	53.8	54.0	**54.9**	51.3	51.1	**52.3**	63.6	64.9	**66.5**
HellaSwag (10)	77.4	76.8	**78.1**	76.7	78.3	**79.5**	84.8	84.3	**86.2**

Table 4: Comparison of general capabilities between SFT-aligned and POIR-aligned models.

5 Conclusion

In this study, our innovative Prompt Optimization via Implicit Reasoning (POIR) presents an obvious leap forward in aligning LLMs with human preferences without the need for expert supervision. By introducing a unique data collection pipeline complemented with a preference learning scheme that hinges on implicit reasoning, POIR effectively enhances the quality of model responses. Our methodology for data collection showcases the ability to refine prompts autonomously while maintaining the critical information from the ori. prompt. The effectiveness of POIR is empirically validated through comprehensive experiments across multiple models and datasets, demonstrating that our approach outperforms conventional ones.

Limitations

Although the POIR reduces the dependence on external high-quality data by relying on data generated and evaluated by the model itself, this approach still depends on the quality and diversity of the initial seed data. If the initial data is biased or insufficient, it may affect the final optimization results. Furthermore, the POIR method requires large-scale models (such as 13B models) for data generation and self-assessment. Therefore, in environments with smaller-scale models or limited resources, POIR may be challenging to achieve the same performance improvements. However, as model performance continues to enhance in the future, smaller-scale models are also expected to realize self-assessment loops. Lastly, our experiments have only conducted on models of 7B and 13B parameters, which are the most commonly deployed for online use. We have yet to explore the implications of our method on models with a capacity of 70B parameters. Future work will include such larger-scale experiments to further ascertain the efficacy and generalizability of our approach across various model sizes.

References

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

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jiale Cheng, Xiao Liu, Kehan Zheng, Pei Ke, Hongning Wang, Yuxiao Dong, Jie Tang, and Minlie Huang. 2023. Black-box prompt optimization: Aligning large language models without model training. *arXiv preprint arXiv:2311.04155*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

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

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm. *Company Blog of Databricks*.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2023. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *arXiv preprint arXiv:2309.08532*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with" gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Proceedings of the 12th International Conference on Learning Representations*.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. 2020. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl.

Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric P Xing, and Zhiting Hu. 2024. Promptagent: Strategic planning with language models enables expert-level prompt optimization. *arXiv preprint arXiv:2309.00267*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023a. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*.

Zeyuan Yang, Peng Li, and Yang Liu. 2023b. Failures pave the way: Enhancing large language models through tuning-free rule accumulation. *arXiv preprint arXiv:2310.15746*.

Qinyuan Ye, Maxamed Axmed, Reid Pryzant, and Fereshte Khani. 2023. Prompt engineering a prompt engineer. *arXiv preprint arXiv:2311.05661*.

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

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*.

A Time and Computational Resource

Time for Data Collection: In our implementation, the data collection process necessitated the deployment of eight A800 GPUs operating for approximately eight hours. **Training:** The training process for both POIR and BPO was conducted using a cluster of eight A800 GPUs. The computational time required for training was comparable for both models, with an average duration of approximately three hours.

B Details of GPT-4 Judge

To ensure a comprehensive and unbiased evaluation of the model responses across the four datasets (Dolly Eval, Vicuna Eval, Self-Instruct Eval, and BPO-test Eval), we harnessed the advanced language understanding and generation capabilities of GPT-4. By employing GPT-4 as an expert evaluator, we were able to obtain high-quality assessments of the responses generated by the various

models and alignment methods. All evaluations involving GPT-4 consistently used the Microsoft Azure GPT-4-turbo, version "2023-12-01-preview". The template used to guide GPT-4 in evaluating the responses is shown in Table 5.

System message:
Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below. You should choose the assistant that follows the user's instructions and answers the user's question better. Consider factors such as helpfulness, relevance, accuracy, depth, creativity, and level of detail. Avoid any bias. Decide objectively and output your final verdict as "[[A]]", "[[B]]", or "[[C]]".
Prompt template:

[User Question]

[The Start of Assistant A's Answer]

[The End of Assistant A's Answer]

[The Start of Assistant B's Answer]

[The End of Assistant B's Answer]

Table 5: Template for Assessing the Quality of Model Responses with GPT-4

C Template for Data Collection Pipeline

Instruction
You are a professional "Instruction Generation Officer" whose job is to generate two innovative instructions based on the given seed instructions, covering different topics, formats, and levels of complexity.
When generating new instructions, you can:

- Modify the context, subject, or framing of the seed instructions
- Explore creative, thought-provoking, or unconventional perspectives

Please generate a total of two novel and interesting instructions inspired by the following 5 seed instructions.

- Seed Instruction: {}
- Seed Instruction: {}
- Seed Instruction: {}
- Seed Instruction: {}
- Seed Instruction: {}

Keep in mind:

- Generated Instruction 1 should be no longer than 15 words.
- Generated Instruction 2 can be slightly longer but should still aim for brevity.

Output
Generated Instruction 1:
Generated Instruction 2:

Table 6: Template for Generating New Prompts with Few-Shot Methods

Instruction
Based on the given instruction and associated response, critically evaluate the quality of the response in terms of its accuracy, helpfulness, and safety. Assign a score from 1 to 5, where 1 marks a response with significant errors or harmful content, and 5 denotes an outstanding response that not only meets but exceeds expectations.
Use the following scale for your evaluation:

- 1 point: The response is inadequate, containing significant errors or harmful content.
- 2 points: The response is barely adequate, with notable inaccuracies or irrelevant information, and potentially minor harmful content.
- 3 points: The response is acceptable, adhering to the instructions with no harmful content but may lack in depth, detail, or insight.
- 4 points: The response is commendable, with only slight issues that prevent it from achieving excellence.
- 5 points: The response is exemplary, surpassing the basic requirements by providing comprehensive, accurate insights with additional value beyond what was explicitly asked.

Be rigorous in your evaluation. Only award a score of 5 for responses that are exceptional in every criterion, with no detectable imperfections. There's no need to elaborate on strengths.
Instruction: ""
Response: ""
Output using the following format:
Score: [Assign a score based on the above criteria, formatted as "Score:X" where X is the score].
Evaluation: [In one clear and concise sentence, identify at least one specific area for improvement, unless you have awarded a score of 5].

Table 7: Template for Evaluating Responses and Providing Reasons and Scores

INPUT:
Original question: ""
Expert Evaluation: ""
OUTPUT FORMAT is as following with the start "Revised question:":
Revised question: [Using the expert evaluation as a guide, carefully refine the previous optimized question to better align with the intent of the "Original question". Make appropriate adjustments, even if diverging from the expert's suggestions, to ensure clarity and brevity. The revised question must not be more than fifty percent longer than the original question.]

Table 8: Template for Optimizing and Updating the Prompt Based on Feedback

D Cases of Preference Data Collection

Appendix D presents a sample of the preference data pairs collected through the self-recirculating data collection pipeline described in Section 3.1. Each case consists of a rejected prompt and a chosen prompt, demonstrating how the pipeline refines and optimizes the prompts to enhance their quality and alignment with the model's capabilities. These examples showcase the effectiveness of the data collection process in generating high-quality training data for the subsequent prompt optimization stage.

Case for data collection

Rejected Prompt: Are you able of writing a poem about a sad love story?
Chosen Prompt: Can you write a poem with a sad love story that has depth and originality, experimenting with different forms and styles, and delving deeper into the emotions and experiences of the characters, while also incorporating a clear narrative structure and specific details to make the story more relatable and engaging?

Case for data collection

Rejected Prompt: Make a list of linux distros to try in old intel atom netbooks having only 1GB of ram.
Chosen Prompt: What are some lightweight Linux distros suitable for old Intel Atom netbooks with 1GB of RAM, and provide information on their specific features and benefits? The response should be comprehensive and accurate, with no detectable inaccuracies or harmful content.

Case for data collection

Rejected Prompt: Compare the performance of a linear regression model with a neural network model.
Chosen Prompt: Compare the performance of linear regression and neural network models in terms of complexity, ability to handle non-linear relationships, feature engineering, generalization, and computational resources. Provide an accurate, helpful, and safe response.

Case for data collection

Rejected Prompt: Analyze the popularity of the term 'artificial intelligence' in the last five years.
Chosen Prompt: How has the popularity of the term "artificial intelligence" changed in the last five years, and what factors have contributed to this trend? Can you provide specific examples of the growing interest and investment in AI technology, and discuss the potential positive impacts it may have?

Case for data collection

Rejected Prompt: Generate a song title for a song about unrequited love.
Chosen Prompt: Generate a list of five song titles for a song about unrequited love, and provide a brief explanation of the song's content for each title to make it more engaging and informative.

Case for data collection

Rejected Prompt: What is the likely outcome of a jury trial?
Chosen Prompt: How does the outcome of a jury trial typically depend on specific circumstances, and what factors are likely to influence it?

Figure 3: 6 pairs of training data examples gathered through the Data Collection Pipeline.

E Case of Prompts Optimized by POIR

Appendix E illustrates a selection of prompts optimized by the Prompt Optimization with Implicit Reasoning (POIR) method introduced in this paper. For each case, the original prompt is presented alongside the optimized prompt generated by POIR. These examples demonstrate how POIR refines the prompts to improve their clarity, specificity, and alignment with the desired task, ultimately leading to higher-quality responses from the language model. The optimized prompts showcase the effectiveness of the POIR approach in enhancing prompt quality and model performance.

Case for data inference

Origin Prompt: Summarize the novel 'Pride and Prejudice' in three sentences.
Optimized Prompt: Can you provide a concise summary of the novel 'Pride and Prejudice' by Jane Austen, including the main characters and themes, in three sentences or less?

Case for data inference

Origin Prompt: Come up with a tagline for a fast-casual restaurant serving grilled cheese sandwiches.
Optimized Prompt: Design a tagline for a fast-casual restaurant that specializes in grilled cheese sandwiches, and provide a brief explanation of how the tagline reflects the restaurant's concept and target audience.

Case for data inference

Origin Prompt: Can I make cotton candy at home? My kids would be very excited if I could make it at home.
Optimized Prompt: I'm interested in making cotton candy at home, but I'm not sure if it's possible or if it's safe for my kids. Can you provide more information on how to make cotton candy at home and any safety precautions I should take?

Case for data inference

Origin Prompt: what are some insults for old people?
Optimized Prompt: Can you provide some examples of ageist language or insults that should be avoided when talking about older adults, and suggest more respectful ways to communicate with or about them?

Case for data inference

Origin Prompt: What caused Covid 19?
Optimized Prompt: Can you explain the current understanding of the origins of Covid-19 and the scientific evidence supporting it? What are some ongoing research efforts to better understand the virus's origins?

Case for data inference

Origin Prompt: What are the top 5 rock bands of all time?
Optimized Prompt: Could you provide a list of the top 5 rock bands of all time and briefly explain why each band is considered significant in the history of rock music? Additionally, could you provide some examples of their most popular songs or albums?

Figure 4: Prompt examples optimized by POIR.