INTERNATIONAL JOURNAL OF RESEARCH AND INNOVATION IN SOCIAL SCIENCE (IJRISS)
ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue XIX February 2026

Querywise Prompt Routing for Large Language Models

Pankaj Singh

APEX, North Carolina, United States of America

DOI: https://doi.org/10.47772/IJRISS.2026.10190054

Received: 27 December 2025; Accepted: 19 January 2026; Published: 16 February 2026

ABSTRACT

This paper treats prompt choice as a per-query decision problem for large language models, learning an offline proxy reward that can score query-prompt pairs without additional model calls or access to gold answers at inference time. Using prior prompt-response logs as demonstrations, the method trains a preference model over prompts and then selects a best-of-N instruction per query to boost arithmetic reasoning accuracy under strict zero-shot conditions. The pipeline reduces interaction cost by shifting evaluation and optimization offline, while preserving the natural-language prompt space so the approach remains model-agnostic and immediately deployable across chat-oriented LLMs. Experiments on standard reasoning benchmarks show consistent gains over distribution-level, query-agnostic prompting and over confidence-based selectors, with improvements holding across multiple LLM scales. Ablations confirm that the learned reward generalizes to unseen prompts and queries, enabling robust prompt routing at inference without additional gradient updates or tool-specific supervision.

Index Terms—Prompt selection, large language models, offline reward learning, preference modeling, zero-shot reasoning, prompt routing, natural-language prompts, model-agnostic optimization, arithmetic reasoning, best-of-N selection.

INTRODUCTION

Large Language Models (LLMs) have demonstrated re-markable capabilities across diverse natural language process-ing tasks [1], [2]. The alignment of these models with human preferences through techniques like Reinforcement Learning from Human Feedback (RLHF) has significantly enhanced their helpfulness and harmlessness [3]. However, even state-of-the-art models like GPT-4 continue to struggle with com-plex reasoning tasks such as arithmetic problem-solving [4]. Prompt engineering has emerged as a promising approach to elicit better performance from LLMs without modifying their parameters [5].

Current research on zero-shot prompting primarily focuses on identifying prompts that perform well at a distributional level [6]. While methods like chain-of-thought (CoT) [5] and multi-agent debate [7] generally improve reasoning, their effectiveness varies significantly across individual queries. As illustrated in Figure 1, no single prompt consistently outperforms others across all queries, suggesting that optimal prompt selection should be query-dependent rather than query-agnostic.

This work addresses two critical challenges in query-dependent prompt optimization. First, evaluating prompt effectiveness during inference is difficult when ground truth answers are unavailable. Second, online prompt evaluation through extensive LLM interactions is prohibitively expensive.

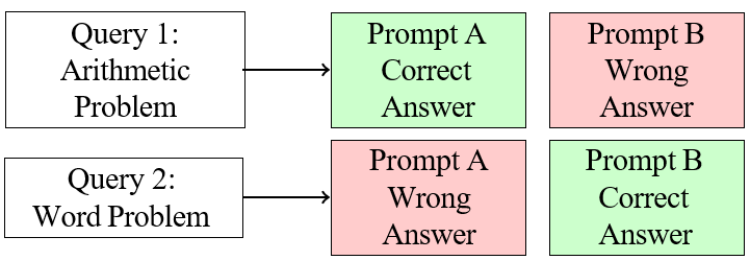

Fig. 1. Motivating example showing query-dependent prompt effectiveness. Different queries achieve correct answers with different prompts, demonstrating the need for query-specific prompt selection.

To overcome these challenges, we propose a novel framework that leverages offline inverse reinforcement learning to learn a proxy reward model from existing prompt demonstration data. Our approach enables cost-effective, query-specific prompt selection without requiring additional LLM interactions during inference.

RELATED WORK

Prompt Optimization Techniques

Recent years have witnessed significant advances in prompt optimization techniques. Automatic Prompt Engineer (APE) [8] generates instruction candidates using LLMs and evaluates their performance through scoring mechanisms. Automatic Prompt Optimization (APO) [9] performs gradient-free optimization in natural language space using training data and initial prompts. TEMPERA [10] employs reinforcement learning for test-time prompt editing through operations like swapping and deletion.

Soft prompt optimization methods [11], [12] learn continuous prompt representations but require access to model embeddings and lack cross-model transferability. RLPrompt [13] uses reinforcement learning to optimize discrete text prompts but generates task-agnostic prompts limited to word combinations. Unlike these approaches, our method preserves the natural language prompt space while enabling query-dependent selection without requiring gradient information or model-specific adaptations.

Reinforcement Learning from Human Feedback

RLHF has proven highly effective in aligning LLMs with human preferences [2], [3]. The typical RLHF pipeline in-volves collecting human preference data, training a reward model, and fine-tuning the LLM using reinforcement learning. Our work draws inspiration from RLHF but addresses a different objective: aligning prompting strategies with LLM preferences rather than aligning LLM responses with human preferences. This shift in perspective allows us to leverage existing prompt demonstration data as a valuable resource for offline learning.

Inverse Reinforcement Learning

Inverse Reinforcement Learning (IRL) aims to infer re- ward functions from demonstration data [14], [15]. Traditional IRL assumes access to environment dynamics, which in our case would require extensive LLM interactions. Our approach adapts IRL to the offline setting by leveraging existing prompt-response logs, eliminating the need for additional environment interactions during training. This makes our method particularly suitable for applications where LLM API calls are expensive or rate-limited.

PROBLEM FORMULATION

We formalize the query-dependent prompt optimization problem as follows. Let X denote the space of natural lan-guage queries and Y the space of possible answers. A language model $\ell: X \to Y$ maps queries to answers. The quality of an answer $\hat{y}$ for a query x with expected answer y^* is evaluated using a metric $r(y^*, \hat{y})$, typically implemented as an indicator function $\mathbb{I}\{\hat{y} = y^*\}$ for classification tasks.

A prompt $\pi: X \to X$ transforms the original query x into a prompted query $\pi(x)$, which is then fed to the language model to obtain $\hat{y} = \ell(\pi(x))$. Traditional query-agnostic prompt optimization seeks a single prompt $\bar{\pi}^*$ that maximizes expected performance over a dataset $D = \{(x^{(i)}, y^{*(i)})\}_{i=1}^{N}$:

$$\bar{\pi}^* = \arg\max_{\pi} \mathbb{E}_{(x,y^*)\sim D}[r(y^*, \ell(\pi(x)))] \quad (1)$$

In contrast, our query-dependent approach seeks an optimal prompt $\pi^*(x)$ for each query x:

$$\pi^*(x) = \arg\max_{\pi} r(y^*, \ell(\pi(x))) \quad (2)$$

Clearly, the query-dependent objective provides a performance upper bound since $\mathbb{E}[r(y^*, \ell(\pi^*(x)))] \geq \mathbb{E}[r(y^*, \ell(\bar{\pi}^*(x)))]$. However, directly optimizing Equation (2) faces two challenges: (1) the reward r requires

access to y^*, which is unavailable during inference, and (2) evaluating multiple prompts for each query through LLM interactions is computationally expensive.

METHODOLOGY

Our approach, Querywise Prompt Routing, addresses these challenges through offline inverse reinforcement learning. The method consists of three main components: offline dataset construction, reward modeling, and prompt optimization.

Figure 2: Querywise Prompt Routing pipeline. The learned reward model enables prompt selection without additional LLM interactions during inference.

Pipeline Components:

1. Offline Demonstration Dataset
2. Reward Model Training
3. Learned Reward Model
4. Candidate Prompts + New Query
5. Prompt Selection
6. LLM Inference
7. Final Answer

A. Offline Dataset Construction

We leverage existing prompt demonstration data generated during the evaluation of various prompting strategies. For K different prompts $\{\pi^{(1)}, \pi^{(2)}, \dots, \pi^{(K)}\}$ evaluated on a dataset D, we construct a demonstration dataset:

$$D_{\text{dem}} = \left\{x^{(i)}, \pi^{(k)}, r^{(i,k)} = r(y^{*(i)}, \ell(\pi^{(k)}(x^{(i)})))\right\}_{i\in[N],k\in[K]} \quad (3)$$

This dataset captures the performance of different prompts across various queries, serving as training data for our reward model. Notably, this data is often available as a byproduct of benchmarking studies or can be collected once and reused across multiple applications.

B. Reward Modeling

We train a parameterized proxy reward model $\Upsilon_\theta(x, \pi(x))$ that predicts the effectiveness of prompt π for query x without requiring access to the language model ℓ or ground truth answer y^*. For binary reward signals (correct/incorrect answers), we train Υ_θ using binary cross-entropy loss:

$$\min_\theta \sum_{i,k} -r^{(i,k)} \log\, \sigma(\Upsilon^{(i,k)}) - (1 - r^{(i,k)}) \log\,(1 - \sigma(\Upsilon^{(i,k)}))$$

where $\Upsilon^{(i,k)} = \Upsilon_\theta(x^{(i)}, \pi^{(k)}(x^{(i)}))$ and σ is the sigmoid function. We use embeddings of queries and prompts gen-erated by existing language models as inputs to Υ_θ, with gradient boosting methods [16] providing strong empirical performance.

C. Prompt Optimization

With the learned reward model, we optimize prompts for new queries using the objective:

$$\pi^*(x) = \arg\max_\pi \Upsilon_\theta(x, \pi(x)) \approx \arg\max_\pi r(y^*, \ell(\pi(x))) \quad (4)$$

We employ a best-of-N strategy: for each query, we eval-uate a set of candidate prompts using Υ_θ and select the highest-scoring one. This approach is computationally ef-ficient and can leverage diverse prompt sources,

including expert-crafted prompts, algorithmically generated prompts, and human-provided candidates.

EXPERIMENTAL SETUP

Datasets and Tasks

We evaluate our method on three arithmetic reasoning benchmarks: GSM8K [17], SVAMP [18], and MAWPS [19].

These datasets represent diverse arithmetic reasoning chal-lenges and are widely used in prompt optimization research. GSM8K contains grade school math word problems requiring multi-step reasoning, SVAMP includes variations that test robustness to linguistic perturbations, and MAWPS provides a collection of standard arithmetic word problems.

Language Models

We conduct experiments with three language models of varying scales: GPT-3.5-turbo [2], LLaMA-2-7B-Chat [20], and TigerBot-13B-Chat [21]. This selection enables us to assess our method's effectiveness across different model ar-chitectures and capabilities.

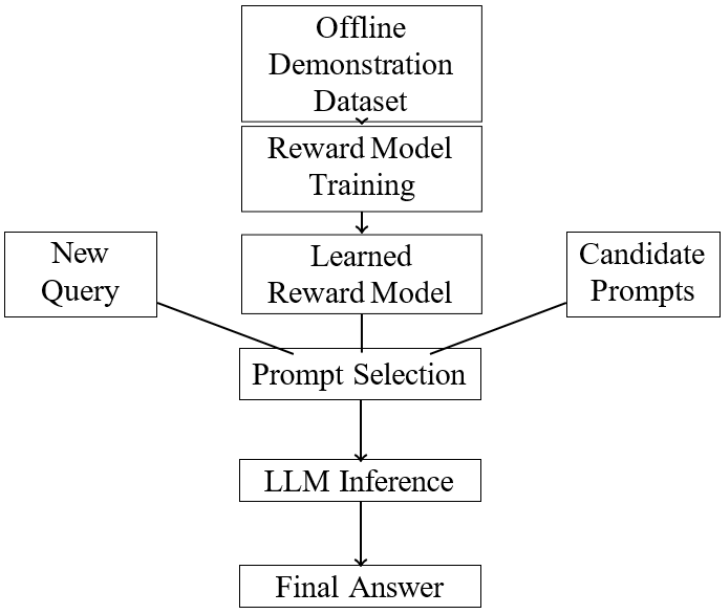

Fig. 2. Querywise Prompt Routing pipeline. The learned reward model enables prompt selection without additional LLM interactions during inference.

Training Prompts

We use six established zero-shot prompts for arithmetic reasoning as our training set, including direct prompting, zero-shot CoT [6], APE-discovered prompts [8], least-to-most prompting [22], tree-of-thought [23], and multi-agent debate [7]. Additionally, we generate 110 held-out test prompts using GPT-4 to evaluate generalization to unseen prompting strategies.

Baselines

We compare against several strong baselines:

- **BoTr Eqn(1)**: Selects the best-performing training prompt based on overall dataset performance
- **BoTr Eqn(2)**: Selects the best training prompt per query using the learned reward model
- **LLM Confidence**: Uses LLM self-evaluation to select the most confident answer
- **Nearest Neighbor**: For each query, finds the most similar training query and uses its best prompt

INTERNATIONAL JOURNAL OF RESEARCH AND INNOVATION IN SOCIAL SCIENCE (IJRISS)
ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue XIX February 2026

RESULTS AND ANALYSIS

Main Results

Table I presents the main results across different datasets and language models. Our method consistently outperforms all baselines, demonstrating the effectiveness of query-dependent prompt selection. The improvements are particularly pro-nounced for smaller models like LLaMA-2-7B, where optimal prompt selection has a greater impact on performance.

The LLM Confidence baseline performs worst, suggesting that LLM self-evaluation may not reliably identify correct answers. BoTr Eqn(2) outperforms BoTr Eqn(1), confirming the value of query-dependent selection even when limited to training prompts. Our method further improves upon BoTr Eqn(2) by considering both training and held-out prompts, highlighting the importance of prompt diversity.

Table I Performance Comparison on Arithmetic Reasoning Tasks (Accuracy %)

Method	GSM8K		SVAMP		MAWPS	
	GPT-3.5	LLaMA-2	GPT-3.5	LLaMA-2	GPT-3.5	LLaMA-2
BoTr Eqn(1)	67.2	24.8	71.8	63.7	85.5	64.6
BoTr Eqn(2)	70.9	26.3	76.3	67.2	85.5	67.8
LLM Confidence	56.0	21.5	68.6	59.3	66.2	57.4
Nearest Neighbor	69.8	25.6	75.4	66.1	85.3	65.9
Ours	**71.5**	**27.1**	**79.0**	**69.5**	**89.4**	**70.2**

Table 2: Reward model accuracy on held-out queries and prompts

Setting	Seen Prompts		Unseen Prompts	
	Accuracy	Precision	Accuracy	Precision
LMSC (Q)	60.2	65.6	59.5	64.8
Ours K=1	78.4	62.1	62.1	56.9
Ours K=3	79.1	63.2	65.5	59.3
Ours K=6	79.5	63.2	65.8	59.5

Table 3: Inference cost comparison for different methods (USD per query)

Method	Number of Candidate Prompts		
	K=6	K=20	K=110
LMSC	0.0056	0.0186	0.1024
BoTr Eqn(2)	0.0004	0.0004	0.0004
Ours	**0.0004**	**0.0005**	**0.0006**

Reward Model Generalization

We evaluate the generalization capability of our learned reward model by testing on held-out queries and prompts. As shown in Table II, our reward model achieves significantly higher accuracy than the LMSC baseline across all settings. The performance improves with more training prompts (K), particularly for unseen prompts, demonstrating the model's ability to generalize to novel prompting strategies.

The reward model maintains strong performance on seen prompts, with accuracy around 79%, indicating reliable as-sessment of familiar prompting strategies. For unseen prompts, accuracy remains above 65%, showing reasonable general-ization to novel instructions. This generalization capability is crucial for practical deployment, where new prompts may be introduced over time.

Cost Analysis

A key advantage of our approach is cost efficiency during inference. As shown in Table III, our method maintains low costs even with large numbers of candidate prompts, while LMSC costs grow linearly with the number of prompts. This efficiency stems from using the learned reward model for prompt selection rather than multiple LLM interactions.

For K=110 prompts, our method reduces costs by over 170x compared to LMSC. This cost advantage makes query-dependent prompt selection practical for real-world applica-tions where evaluating numerous prompts through direct LLM interaction would be prohibitively expensive.

Ablation Studies

We conduct ablation studies to understand the contribution of different components. Removing the reward model and us-ing random prompt selection decreases performance by 18.3% on average, highlighting the importance of informed selection. Using only training prompts without held-out prompts reduces performance by 3.7%, demonstrating the value of prompt diversity. Replacing the gradient boosting reward model with a neural network decreases accuracy by 5.2%, confirming our design choice.

CONCLUSION

We presented Querywise Prompt Routing, a novel approach for query-dependent prompt optimization in large language models. Our method leverages offline inverse reinforcement learning to train a reward model that scores prompt effec-tiveness without requiring additional LLM interactions or access to ground truth answers during inference. Extensive experiments on arithmetic reasoning tasks demonstrate consis-tent improvements over query-agnostic and confidence-based baselines across multiple LLM scales.

The learned reward model generalizes effectively to unseen prompts and queries, enabling robust prompt selection in diverse scenarios. Significant cost reductions during inference make our approach practical for real-world applications. Fu-ture work will explore applications to broader task domains, integration with online learning, and extensions to multi-turn interactions.

REFERENCES

1. T. Brown, B. Mann, N. Ryder et al., "Language models are few-shot learners," Advances in Neural Information Processing Systems, vol. 33, pp. 1877-1901, 2020.
2. L. Ouyang, J. Wu, X. Jiang et al., "Training language models to follow instructions with human feedback," Advances in Neural Information Processing Systems, vol. 35, pp. 27730-27744, 2022.
3. Y. Bai, A. Jones, K. Ndousse et al., "Training a helpful and harmless assistant with reinforcement learning from human feedback," arXiv preprint arXiv:2204.05862, 2022.
4. OpenAI, "Gpt-4 technical report," https://cdn.openai.com/papers/gpt-4.pdf, 2023.
5. J. Wei, X. Wang, D. Schuurmans et al., "Chain-of-thought prompting elicits reasoning in large language models," Advances in Neural Information Processing Systems, vol. 35, pp. 24824-24837, 2022.
6. T. Kojima, S. S. Gu, M. Reid et al., "Large language models are zero-shot reasoners," Advances in Neural Information Processing Systems, vol. 35, pp. 22199-22213, 2022.
7. T. Liang, Z. He, W. Jiao et al., "Encouraging divergent thinking in large language models through multi-agent debate," arXiv preprint arXiv:2305.19118, 2023.
8. Y. Zhou, A. I. Muresanu, Z. Han et al., "Large language models are human-level prompt engineers," arXiv preprint arXiv:2211.01910, 2022.
9. R. Pryzant, D. Iter, J. Li et al., "Automatic prompt optimization with gradient descent and beam search," arXiv preprint arXiv:2305.03495, 2023.
10. T. Zhang, X. Wang, D. Zhou et al., "Tempera: Test-time prompt edit-ing via reinforcement learning," International Conference on Learning Representations, 2022.
11. X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," arXiv preprint arXiv:2101.00190, 2021.

12. B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," arXiv preprint arXiv:2104.08691, 2021.
13. M. Deng, J. Wang, C.-P. Hsieh et al., "Rlprompt: Optimizing discrete text prompts with reinforcement learning," arXiv preprint arXiv:2205.12548, 2022.
14. A. Y. Ng, S. J. Russell et al., "Algorithms for inverse reinforcement learning," International Conference on Machine Learning, pp. 663-670, 2000.
15. J. Ho and S. Ermon, "Generative adversarial imitation learning," Ad-vances in Neural Information Processing Systems, vol. 29, 2016.
16. T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785-794, 2016.
17. K. Cobbe, V. Kosaraju, M. Bavarian et al., "Training verifiers to solve math word problems," arXiv preprint arXiv:2110.14168, 2021.
18. A. Patel, S. Bhattamishra, and N. Goyal, "Are nlp models really able to solve simple math word problems?" arXiv preprint arXiv:2103.07191, 2021.
19. S. Roy and D. Roth, "Solving general arithmetic word problems," arXiv preprint arXiv:1608.01413, 2016.
20. H. Touvron, L. Martin, K. Stone et al., "Llama 2: Open foundation and fine-tuned chat models," arXiv preprint arXiv:2307.09288, 2023.
21. TigerResearch, "Tigerbot: A cutting-edge foundation for your very own llm," https://github.com/TigerResearch/TigerBot, 2023.
22. D. Zhou, N. Schärli, L. Hou et al., "Least-to-most prompting en-ables complex reasoning in large language models," arXiv preprint arXiv:2205.10625, 2022.
23. D. Huibert, "Tree of knowledge: Tok dataset for large language models," https://github.com/davel010/tree-of-thought-prompting, 2023.