MARS: Multi-Agent Adaptive Reasoning with Socratic Guidance for Automated Prompt Optimization

Jian Zhang[*1,2], **Zhangqi Wang**[*1,3], **Haiping Zhu**[1,2†], **Kangda Cheng**[4], **Kai He**[5], **Bo Li**[1,3], **Qika Lin**[5†], **Jun Liu**[1,3], **Erik Cambria**[6]

[1]School of Computer Science and Technology, Xi'an Jiaotong University, China
[2]MOE KLINNS Lab, Xi'an Jiaotong University, China
[3]Shaanxi Province Key Laboratory of Big Data Knowledge Engineering, Xi'an Jiaotong University, China
[4]School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin, China
[5]Saw Swee Hock School of Public Health, National University of Singapore, Singapore
[6]College of Computing and Data Science, Nanyang Technological University, Singapore
zhangjian062422@stu.xjtu.edu.cn, zhuhaiping@xjtu.edu.cn

Abstract

Large language models typically operate in a question-answering paradigm, where the quality of the input prompt critically affects the response. Automated Prompt Optimization (APO) aims to overcome the cognitive biases of manually crafted prompts and explore a broader prompt design space. However, existing APO methods often suffer from rigid template structures and inefficient exploration in the prompt space. To this end, we propose a **M**ulti-Agent **A**daptive **R**easoning with **S**ocratic guidance framework (**MARS**). It consists of five complementary agents and formulates the optimization process as a Partially Observable Markov Decision Process, enabling adaptive prompt refinement through explicit state modeling and interactive feedback. Specifically, a *Planner* agent generates flexible optimization trajectories, a *Teacher-Critic-Student* triad engages in Socratic-style dialogue to iteratively optimize the prompt based on pseudo-gradient signals in the text space, and a *Target* agent executes the prompt in downstream tasks to provide performance feedback. MARS integrates reasoning, feedback, and state transition into a unified hidden-state evolution process, improving both the effectiveness and interpretability of optimization. Extensive experiments across multiple datasets show that MARS outperforms existing APO methods in optimization, efficiency, and interpretability.

Code — https://github.com/exoskeletonzj/MARS

Introduction

Large language models (LLMs) such as GPT-4 (Achiam et al. 2023) and Deepseek-R1 (Guo et al. 2025) provide robust support for thousands of natural language processing tasks (Yuan et al. 2025). By providing a natural language prompt that includes instructions and a task description, LLMs can quickly adapt and respond (Lin et al. 2025a; Shen et al. 2025; Yan et al. 2025). Consequently, the quality of the prompt is of critical importance, leading to wide interest in Automated Prompt Optimization (APO) (Pryzant et al. 2023). As shown in Figure 1, we provide LLMs with three different inputs for the word sorting task: a zero-shot prompt, a Chain of Thought (CoT) prompt, and our optimized prompt. The responses are produced in a markedly distinct way. Specifically, the zero-shot prompt incorrectly identifies the *alterate* as the more common word *alternate*. However, the task requires faithfully preserving the given sequence of words rather than correcting them. With the CoT prompt, the sorting remains incorrect because the LLM does not fully grasp the sorting task and the word sequence. In contrast, our optimized prompt produces the correct answer. This is because our prompt includes specific requirements, such as maintaining the original letter casing and specifying the sorting method.

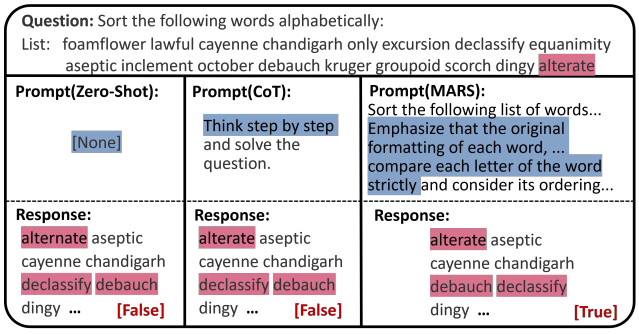

Figure 1: Three different prompts along with their corresponding responses for the word sorting task.

Thus, it is evident that APO can lead to improved performance in downstream tasks. As shown in Figure 2, recent studies (Zhou et al. 2022; Xu, Banburski-Fahey, and Jojic 2023; Wang et al. 2023; Liu et al. 2025) have explored prompt optimization by generating multiple candidates combined with diverse search strategies, while others (Yang et al. 2024a; Ye et al. 2023) focus on designing sophisticated *meta-prompts* to guide optimization. Despite these advances, two key issues remain: the limited flexibil-

[*]These authors contributed equally.
[†]Corresponding Author

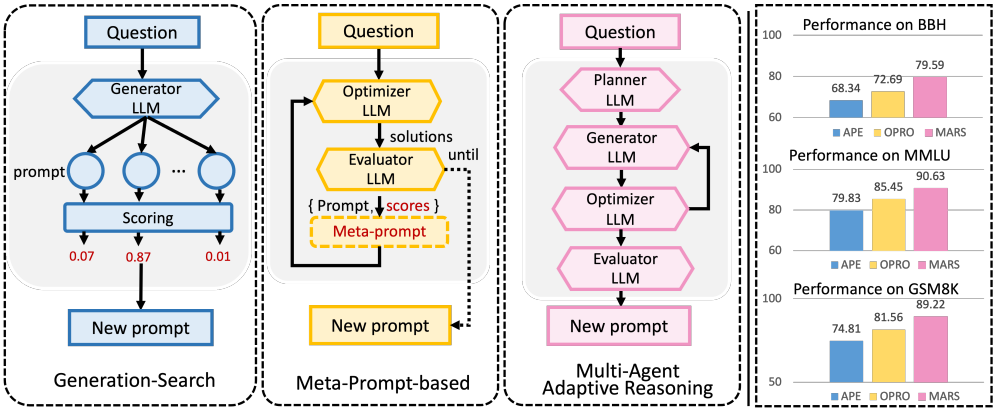

Figure 2: Comparison of APO strategies. Generation-search and meta-prompt. Multi-Agent Adaptive Reasoning enables dynamic, collaborative reasoning. Right: With GPT-4o, MARS outperforms all baselines on three benchmarks.

ity of fixed prompt templates, and the inefficiency of prompt space exploration.

The first issue is the **limited flexibility of fixed templates**. Prior works (Yang et al. 2024a; Ye et al. 2023) often rely on *meta-prompts*, which are predefined optimization templates that cannot be dynamically adapted to different tasks. Unlike domains such as event extraction (Zhang et al. 2024b) or text-to-symbol generation (Xu et al. 2024), where fixed templates suffice due to the task's structural regularity, APO requires more adaptability. Rigid templates may introduce biases or fail to capture task-specific nuances, resulting in suboptimal performance when applied to diverse or complex scenarios.

The second issue is the **inefficiency of prompt space exploration**. Several approaches (Zhou et al. 2022; Xu, Banburski-Fahey, and Jojic 2023; Wang et al. 2023) adopt a *generation-search* strategy, where a set of candidate prompts is first generated and then refined using local search techniques. However, this approach typically performs only local exploration around the initial candidates, without sufficiently covering the broader prompt space. As a result, the optimization may converge prematurely or overlook better-performing prompts, limiting overall effectiveness.

To this end, we propose a **M**ulti-Agent **A**daptive **R**easoning with **S**ocratic guidance framework (**MARS**) for APO. MARS consists of five complementary agents and formulates the optimization process as a Partially Observable Markov Decision Process (POMDP), enabling adaptive prompt refinement through explicit state modeling and interactive feedback. Functionally, to address the first challenge, MARS introduces a *Planner* agent that generates task-specific optimization trajectories, allowing prompts to be flexibly adapted to diverse task requirements. To tackle the second challenge, MARS employs a Socratic-style *Teacher-Critic-Student* dialogue mechanism, which iteratively guides prompt refinement. This module enables effective exploration of the prompt space by simulating a gradient-inspired optimization process, while also promoting interpretability.

The overall process is modeled as a POMDP, where the hidden state represents the latent reasoning state of the *Student* agent. Through multi-agent interactions and performance feedback from a *Target* agent, MARS approximates a pseudo-gradient trajectory in the discrete prompt space, progressively refining the prompt toward an optimal solution. Our contributions are three-fold:

- This work is the first to introduce a multi-agent architecture with POMDP modeling for APO. It proposes MARS, which enables hidden-state reasoning and adaptive planning through agent collaboration.
- A *Teacher-Critic-Student* Socratic dialogue mechanism is designed to enable interpretable, iterative prompt refinement via a gradient-inspired optimization trajectory.
- We demonstrate the effectiveness and generalizability of MARS through extensive experiments on both general and domain-specific benchmarks, and validate the interpretability of its optimization process.

Methodology

MARS comprises two main modules: (i) a high-level *Planner* that generates task-specific optimization trajectories, and (ii) a *Teacher-Critic-Student* triad that performs Socratic-style iterative refinement. The overall architecture is shown in Figure 3, with the complete workflow detailed in Algorithm 1. This section introduces: (1) the APO task and its POMDP formulation, (2) the *Planner* design, (3) the gradient-inspired Socratic dialogue mechanism, and (4) the evaluation-feedback loop via the *Target* agent.

Task Formulation and POMDP Modeling

Given a task-specific *Target* model $\mathcal{M}_{\text{tar}}$, the goal of APO is to iteratively refine an initial prompt p_0 to an optimal version p^* that maximizes performance on a downstream dataset $D = \{(x, y)\}$. A training subset $D_{\text{train}} \subset D$ guides the optimization, while D_{test} is used for evaluation. The objective can be formalized as:

$$p^* = \arg\max_{p} \sum_{(x,y)\in D_{\text{test}}} f\left(\pi_{\text{tar}}(x; p), y\right), \qquad (1)$$

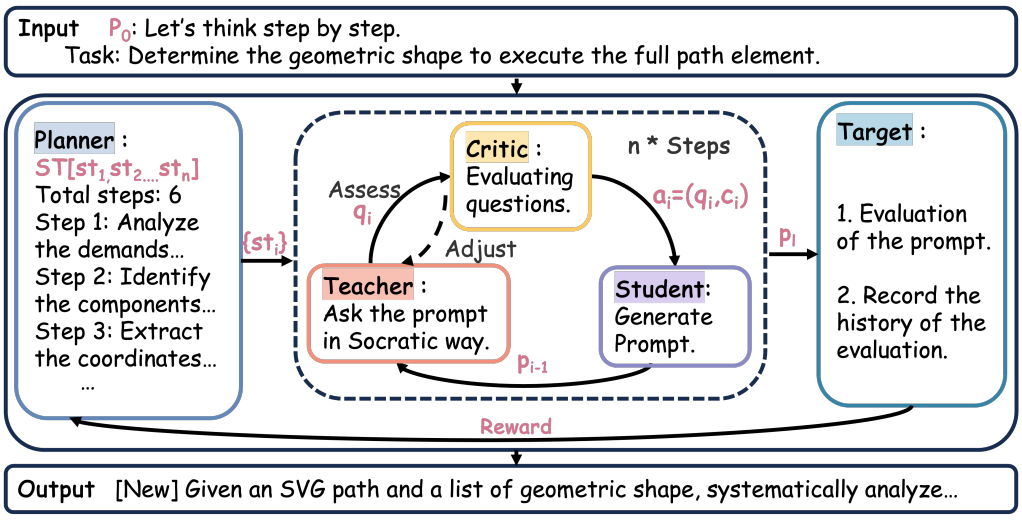

Figure 3: The overall architecture of the MARS model. It consists of five LLM agents. The *Planner* agent that autonomously generates task-specific optimization trajectories, and a *Teacher-Critic-Student* Socratic dialogue mechanism that iteratively refines prompts, with the evaluation and iterative refinement process guided by feedback from the *Target* agent.

where $\pi_{\text{tar}}(x; p)$ denotes the model output conditioned on x and prompt p, and f is a task-specific metric (e.g., accuracy, BLEU).

To capture the sequential, partially observable nature of the optimization, we model APO as a Partially Observable Markov Decision Process (POMDP) defined by:

$$\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{O} \rangle,$$

where: - $\mathcal{S}$: latent state space, representing the internal reasoning state of the *Student* agent; - $\mathcal{A}$: action space, comprising instructional signals (e.g., questions, critiques) from *Teacher* and *Critic*; - $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$: transition dynamics, updating student states; - $\mathcal{O} : \mathcal{S} \to \mathcal{P}$: observation function, mapping hidden states to prompts; - $\mathcal{R}(s, a) = f(\pi_{\text{tar}}(x; \mathcal{O}(s)), y)$: reward function, based on performance of the generated prompt.

This formulation allows MARS to perform gradient-inspired prompt refinement in a partially observable, discrete text space. Via iterative multi-agent reasoning and feedback, the system progressively transitions from p_0 to p^*.

Optimization Trajectory Planning

As illustrated in Figure 3, MARS begins with a *Planner* agent that initiates the prompt optimization process.

Planner. Given task goal g, input $x \in D_{\text{train}}$, and initial prompt p_0, the *Planner* decomposes the optimization into a sequence of sub-goals:

$$\mathbf{ST} = [st_1, st_2, \ldots, st_n] = \pi_{\text{plan}}(g, x, p_0), \tag{2}$$

where π_{plan} is the planning policy that adaptively generates an optimization trajectory.

To formalize π_{plan}, we introduce a latent planning variable $z \in \mathcal{Z}$, and model trajectory generation as:

$$\pi_{\text{plan}}(g, x, p_0) = \arg\max_{\mathbf{ST}} \mathbb{E}_{z \sim q(z|g,x)} \left[\log P(\mathbf{ST} \mid z, p_0)\right], \tag{3}$$

where $q(z|g, x)$ captures task semantics, and $P(\mathbf{ST} \mid z, p_0)$ models the trajectory conditioned on latent intent and initial prompt. This hierarchical formulation induces structured plans over latent space $\mathcal{S}$, guiding local agent decisions under global coherence and improving adaptability over static templates.

Socratic Prompt Refinement as Joint Policy Optimization

Prompt refinement in discrete language space presents unique challenges due to its non-differentiability, high variance, and semantic ambiguity. To address these issues, MARS employs a structured Socratic dialogue mechanism involving three collaborative agents—*Teacher* (π_{t}), *Critic* (π_{c}), and *Student* (π_{s})—each fulfilling a complementary role in exploring and improving prompts through guided interaction. This framework transforms prompt optimization into an interpretable, policy-driven reasoning process grounded in pedagogical principles.

At each refinement step i, given a sub-goal $st_i \in \mathbf{ST}$, the *Teacher* proposes a Socratic-style question q_i to stimulate reasoning, based on the prior prompt p_{i-1}. The *Critic* then assesses its clarity, relevance, and coherence, producing feedback c_i to revise or validate the proposed direction. Finally, the *Student* responds by updating its internal state and generating a new prompt p_i. This process is formalized as:

$$\begin{aligned} q_i &= \pi_{\text{t}}(st_i, p_{i-1}), \\ c_i &= \pi_{\text{c}}(q_i), \\ p_i &= \pi_{\text{s}}((q_i, c_i), p_{i-1}), \\ s_i &\sim \mathcal{T}(s_{i-1}, (q_i, c_i)), \quad o_i = p_i. \end{aligned} \tag{4}$$

Each agent performs a partial update to the joint optimization process: *Teacher* drives semantic direction, *Critic* provides quality control, and *Student* synthesizes the final output.

Context-Aware Interaction. To improve reasoning consistency and avoid step-wise myopia, each agent conditions not only on the current sub-goal and prompt, but also on the dialogue history $\mathcal{H}_{<i} = \{(q_j, c_j, p_j)\}_{j=1}^{i-1}$. The full context-aware behavior is given by:

$$\begin{aligned} q_i &= \pi_{\text{t}}(st_i, p_{i-1}, \mathcal{H}_{<i}), \\ c_i &= \pi_{\text{c}}(q_i, \mathcal{H}_{<i}), \\ p_i &= \pi_{\text{s}}((q_i, c_i), p_{i-1}, \mathcal{H}_{<i}). \end{aligned} \tag{5}$$

By attending to prior reasoning steps, the system forms coherent, memory-informed trajectories across iterations.

Joint Optimization Objective. We define the multi-agent policy as $\Pi = \{\pi_{\text{t}}, \pi_{\text{c}}, \pi_{\text{s}}\}$, and optimize it jointly to maximize task performance while ensuring interpretability and alignment with sub-goals:

$$\max_{\Pi} \mathbb{E}_{(x,y)\sim D} \left[\mathcal{R}(\Pi) - \lambda \sum_{i=1}^{n} \mathcal{L}_{\text{align}}((q_i, c_i), st_i) \right], \tag{6}$$

where $\mathcal{R}(\Pi)$ denotes the cumulative reward from the *Target* agent, and $\mathcal{L}_{\text{align}}$ penalizes semantic drift from intended optimization goals.

This tri-agent structure enables interpretable, step-wise refinement of prompts through structured reasoning and localized feedback, offering both flexibility and transparency in discrete prompt optimization.

Proposition 1 (Socratic Policy Improvement Bound). Let $\Pi = \{\pi_{\text{t}}, \pi_{\text{c}}, \pi_{\text{s}}\}$ denote the joint policy, and suppose the Socratic signal $a_i = (q_i, c_i)$ induces expected advantage $\bar{A}_i > 0$ over the prior state s_{i-1}. Then, under bounded variance σ^2, the cumulative improvement over n steps satisfies:

$$\mathbb{E}[\mathcal{R}(p_n)] - \mathcal{R}(p_0) \geq \sum_{i=1}^{n} \left(\bar{A}_i - \frac{\sigma^2}{2\lambda} \right), \tag{7}$$

where λ is the local Lipschitz constant of the reward surface.

This provides a lower bound on improvement, formally linking guidance signal quality to reward trajectory.

Evaluation and Iteration

Upon completing the Socratic refinement trajectory, the final prompt $p_\ell = p_n$—produced through successive dialogue transitions from latent state s_0 to s_ℓ—is evaluated by the *Target* agent π_{tar} on the held-out test set D_{test}. The evaluation provides an external signal to measure the effectiveness of the entire optimization trajectory:

$$\mathcal{R}^{(t)} = \sum_{(x,y)\in D_{\text{test}}} f\left(\pi_{\text{tar}}(x; p_\ell^{(t)}), y\right), \tag{8}$$

where $f(\cdot)$ is a task-specific scoring function (e.g., accuracy, BLEU, F1), and $p_\ell^{(t)}$ denotes the final prompt obtained at iteration t. This scalar reward serves as the global performance metric, closing the loop between prompt generation and task-level effectiveness.

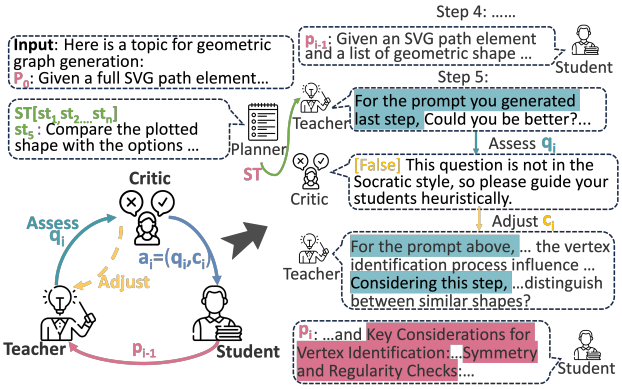

Figure 4: A specific illustration of a *Teacher-Critic-Student* Socratic guidance dialogue pattern. The case shows the fifth step optimization iteration.

Adaptive Termination. To ensure efficient convergence and prevent over-refinement, we adopt an adaptive early stopping criterion based on marginal reward improvement. The gain between two consecutive iterations is defined as:

$$\Delta\mathcal{R}^{(t)} = \mathcal{R}^{(t)} - \mathcal{R}^{(t-1)}. \tag{9}$$

The refinement continues only if:

$$\Delta\mathcal{R}^{(t)} > \delta, \quad t < I, \tag{10}$$

where δ is a minimum improvement threshold, and I is the maximum number of allowed iterations. This iterative control mechanism enforces a form of performance-aware policy halting under the POMDP framework. It ensures that MARS focuses on high-impact updates while avoiding excessive computation on marginally beneficial revisions. As a result, the system adaptively determines the optimal stopping point based on observable task performance.

Proposition 2 (Monotonic Reward Stability). Assume $\mathcal{R}(p)$ is λ-Lipschitz and each step satisfies $\|p_i - p_{i-1}\| \leq \varepsilon$. Then the reward trajectory $\{\mathcal{R}(p_i)\}$ satisfies:

$$|\mathcal{R}(p_{i+1}) - \mathcal{R}(p_i)| \leq \lambda\varepsilon.$$

In particular, if $\mathcal{R}(p_i) < \mathcal{R}(p_{i+1})$ for some i, then improvement is bounded and monotonic. *This result guarantees bounded gain/loss and motivates early stopping under stable improvement.*

Experiments

In this section, we present extensive experiments conducted on 12 general task datasets and 5 domain-specific datasets. We begin by introducing the datasets and hyperparameters, followed by the main experimental results. A detailed analysis of MARS' efficiency is also provided.

Experimental Settings

Datasets. We select a total of 17 datasets covering both general-purpose and domain-specific tasks. Specifically, we use 6 tasks from the Big-Bench Hard (BBH) suite (Suzgun

Models	B.E	D.QA	F.F.	G.S.	R.N.	S.U.	C.B.	C.M.	E.E.	W.H.	H.A.	M.T.	Avg.
Origin	74.70	51.41	52.20	43.37	59.84	60.24	82.52	69.77	63.89	73.73	66.22	81.55	64.95
CoT(ZS)	80.32	54.22	59.44	47.39	67.07	67.87	83.91	73.25	74.31	76.27	68.47	84.98	69.79
CoT(FS)	81.93	57.43	66.26	49.40	70.68	72.29	86.71	76.74	79.17	78.81	72.07	90.99	73.54
APE	83.53	61.85	61.04	51.41	77.51	74.70	88.11	75.58	69.44	82.20	75.68	87.98	74.09
ProTeGi	83.93	63.86	62.65	52.21	80.32	76.71	90.91	78.49	73.61	84.75	77.48	90.56	76.29
OPRO	86.34	66.67	63.45	53.81	83.13	82.73	93.70	83.14	77.01	86.44	79.73	92.70	79.07
PE2	87.95	65.46	63.86	54.62	84.34	75.90	93.01	81.40	76.39	88.14	81.08	93.56	78.81
Ours	**93.17**	**71.89**	**74.70**	**59.44**	**90.36**	**87.95**	**97.90**	**86.05**	**84.03**	**93.22**	**85.59**	**97.00**	**85.11**

Table 1: In the performance comparison across 12 general tasks, we carefully select 6 representative subtasks from both BBH and MMLU, two commonly used evaluation benchmarks, to comprehensively assess MARS's performance in diverse general-task settings. The evaluation results of these subtasks indicate that MARS surpasses all existing baseline methods.

Algorithm 1: MARS Optimization Procedure

Input: Dataset $\mathcal{D}$, initial prompt p_0, threshold δ, max iterations I
Output: Optimized prompt p^*
Planner: Generate sub-goal trajectory $\mathbf{ST} = \{st_1, \ldots, st_n\}$
Initialize $p_0^{(1)} \leftarrow p_0$, $\mathcal{R}^{(0)} \leftarrow 0$
for iteration $t = 1$ to I **do**
  for step $i = 1$ to n **do** *// Generate question*
    Teacher generates question $q_i \leftarrow \pi_{\text{t}}(st_i, p_{i-1}^{(t)})$
    repeat
      Critic evaluates q_i & returns feedback $c_i \leftarrow \pi_{\text{c}}(q_i)$
      Teacher revises q_i if c_i is unsatisfactory
    until Socratic quality is satisfied
    Set $a_i \leftarrow (q_i, c_i)$
    Student updates $p_i^{(t)} \leftarrow \pi_{\text{s}}(a_i, p_{i-1}^{(t)})$
  end for
  Let $p_\ell^{(t)} \leftarrow p_n^{(t)}$ *// Final prompt*
  Target evaluates reward
  $\mathcal{R}^{(t)} = \sum_{(x,y)\in D_{\text{test}}} f(\pi_{\text{tar}}(x; p_\ell^{(t)}), y)$
  if $\mathcal{R}^{(t)} - \mathcal{R}^{(t-1)} < \delta$ **then**
    break *//Early stopping*
  end if
end for
return $p^* \leftarrow p_\ell^{(t)}$

et al. 2022) and 6 tasks from MMLU (Wang et al. 2024b) to represent general reasoning and knowledge-intensive benchmarks. For domain-specific evaluation, we include 3 Chinese subject-area tasks from C-Eval (Huang et al. 2024), 1 legal reasoning task from LSAT-AR (Zhong et al. 2023), and 1 arithmetic reasoning task from GSM8K (Zhang et al. 2024a).

Hyperparameters and Evaluation Protocol. We adopt `deepseek-v2.5-1210` (Guo et al. 2025) as the primary backbone LLM for all APO tasks. The generation temperature is set to 0.6 to balance creativity and coherence. We configure the maximum number of optimization iterations as $I = 10$, with an early stopping threshold of $\delta = 0.01$ based on accuracy improvement. To enhance efficiency, each *assess-adjust* cycle is limited to a single revision per step.

Models	Chinese			Math	Law	Avg.
	A.S.	U.R.P.	CL.M.	GSM.	L.A.	
Origin	56.25	48.89	57.14	67.07	23.14	50.50
CoT(ZS)	59.38	53.33	61.90	70.26	30.57	55.09
CoT(FS)	65.63	57.78	66.67	77.54	35.81	60.69
APE	65.63	62.22	71.43	74.81	29.69	60.76
ProTeGi	68.75	66.67	76.19	77.47	31.88	64.19
OPRO	71.88	73.33	80.95	81.56	31.44	67.83
PE2	75.00	77.78	76.19	83.46	34.50	69.39
MARS	**81.25**	**84.44**	**85.71**	**89.22**	**38.42**	**75.81**

Table 2: Performance comparison on three types of domain-specific tasks: Chinese, law, and mathematics. The Chinese domain consists of three datasets, while the law and mathematics domains each have one dataset.

Final evaluation is performed using accuracy, computed by comparing the model prediction y_{pred} with the ground truth label y.

Main Results

MARS enhances the average performance across diverse task types. The experimental results in Table 1 and Table 2 present a comprehensive comparison between the prompts optimized by MARS and the baselines for the 12 tasks. As shown in Table 1, on general tasks, MARS outperforms the previous SOTA by 6.04%, and exceeds the original prompt and CoT(ZS) by 20.16% and 15.32%, respectively. This indicates that MARS-optimized prompts enable LLMs to better understand task requirements, providing stronger instructions for tasks across different scenarios. MARS surpasses existing APO methods, highlighting the limitations of both the *generate-search* approach and the *meta prompts* approach. These methods do not fully grasp the deeper essence of the APO process, which constrains their effectiveness in optimization. In contrast, MARS thoughtfully considers the prompt optimization pathways for different tasks and incorporates heuristic optimization strategies, making the prompt refinement process more efficient and precise.

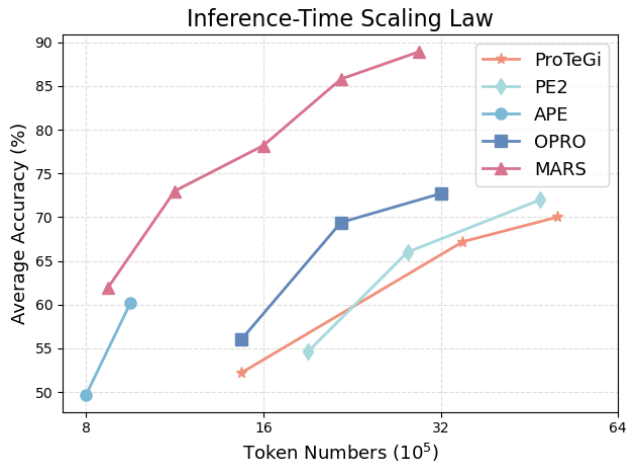

Figure 5: Inference-time scaling law. The horizontal axis denotes the inference-time computational cost, while the vertical axis represents the average performances on all tasks.

MARS achieves strong and consistent performance gains across domain-specific tasks, highlighting its effectiveness in knowledge-intensive reasoning. Table 2 presents the experimental results of MARS on domain-specific tasks, covering areas like Chinese, law, and mathematics, all of which require specialized knowledge and reasoning. In these tasks, MARS outperforms the previous SOTA methods to 6.42%, further demonstrating its ability to better guide LLMs in domain-specific knowledge discovery and application. This not only lowers the barrier to utilizing LLMs but also enhances their generalization capability. Moreover, compared to the original prompt and CoT(ZS), MARS achieves improvements of 25.31% and 20.72%, respectively, underscoring its effectiveness and practicality in these specialized domains.

Efficiency Analysis

MARS Consistently Achieves the Highest Computational Efficiency. The balance between resource consumption and performance improvement is a crucial analysis metric (Yang et al. 2024b). As shown in Figure 5, MARS consistently outperforms all baseline methods in terms of computational efficiency, as demonstrated by its superior inference-time scaling behavior across multiple APO tasks.

Notably, under the same number of output tokens, MARS achieves the highest performance across all evaluated tasks. Conversely, to reach comparable performance levels, baseline methods require more output tokens—indicating higher computational cost.

These results highlight MARS's strong ability to balance performance and resource usage through its structured optimization strategy. By performing high-level task planning followed by step-wise Socratic refinement, MARS enables more efficient resource allocation, reduces unnecessary computation, and ensures both effectiveness and robustness throughout the APO process.

Variation	B.E.	D.QA	F.F.	G.S.	R.N.	S.U.	Avg.
MARS	93.17	71.89	74.70	59.44	90.36	87.95	79.59
w/o_{Plan}	86.35	65.86	68.67	54.21	82.33	79.52	72.82
Δ	(-6.82)	(-6.03)	(-6.03)	(-5.23)	(-8.03)	(-8.43)	(-6.77)
w/o_{Soc}	84.74	63.86	62.25	49.80	74.30	74.70	68.28
Δ	(-8.43)	(-8.03)	(-12.45)	(-9.64)	(-16.06)	(-13.25)	(-11.31)
w/o_{Cri}	89.16	68.27	72.28	56.22	86.34	83.94	76.04
Δ	(-4.01)	(-3.62)	(-2.42)	(-3.22)	(-4.02)	(-4.01)	(-3.55)

Table 3: Performance under different ablation settings are analyzed. We performed ablation experiments on the planner module w/o_{Plan}, the *Teacher-Critic-Student* module w/o_{Soc}, and the *Critic* Agent w/o_{Cri} to evaluate the impact of removing these components. *w/o* indicates the experiment was run without the specified module.

Supplementary Analysis

To further validate the effectiveness of MARS, we conduct three additional analyses in this section: an ablation study to assess the contribution of each component, a convergence analysis to examine the optimization stability over iterations, and an investigation of the sensitivity to Other *Target* LLMs.

Ablation Study

The Socratic dialogue mechanism plays the most critical role in MARS, as shown by the largest performance drop upon its removal. Table 3 presents the impact of removing three key components: the *Planner* agent, the *Teacher-Critic-Student* Socratic module, and the *Critic* agent. Removing the Socratic module leads to the most substantial degradation, as the system loses its iterative refinement capability and sends unprocessed sub-goals directly to the *Target* agent, resulting in poor optimization quality. Eliminating the *Planner* also causes a notable drop, since the Socratic dialogue lacks structured guidance without its sub-goal trajectory. Finally, while the *Critic* contributes less overall, its feedback loop with the *Teacher* improves prompt quality; removing it leads to a 3.55% performance loss, as shown in Table 3.

Converagence Analysis

MARS achieves faster convergence in most tasks, improving both efficiency and optimization quality. Figure 6 presents the convergence analysis across four BBH tasks. To better monitor the APO process, we visualize the iterative optimization trajectory within a 10-iteration observation window.

The results show that MARS exhibits an upward reward trend in the early stages. For instance, in Task 'Ruin Names', it converges to the optimal solution by iteration 5. In contrast, in the OPRO task, convergence is not reached even after 10 iterations, resulting in higher resource consumption. This comparison highlights MARS's ability to reach optimal prompts in fewer steps, reducing computational cost and enhancing efficiency.

Base	Deepseek		GPT			Avg.
	-V2.5	-R1	-3.5	-4	-4o	
Origin	56.96	61.48	44.79	49.70	55.84	53.75
CoT(ZS)	62.72	73.82	63.45	66.94	70.38	67.46
MARS	**79.59**	**83.05**	**69.30**	**73.21**	**80.86**	**77.20**

Table 4: Performance comparison on BBH tasks under different *Target* model settings.

Other *Target* LLMs

MARS demonstrates strong cross-model generalization, maintaining high performance across diverse LLM backbones. We further evaluate MARS on additional *Target* LLMs—Deepseek-R1, GPT-3.5, GPT-4, and GPT-4o—using the optimized prompts from previous experiments to assess robustness across model families. As shown in Table 4, prompts optimized on the Deepseek-V2.5 base model generalize well, preserving strong performance even on larger or structurally different LLMs. MARS consistently achieves notable gains across models, validating its model-agnostic design and broad applicability.

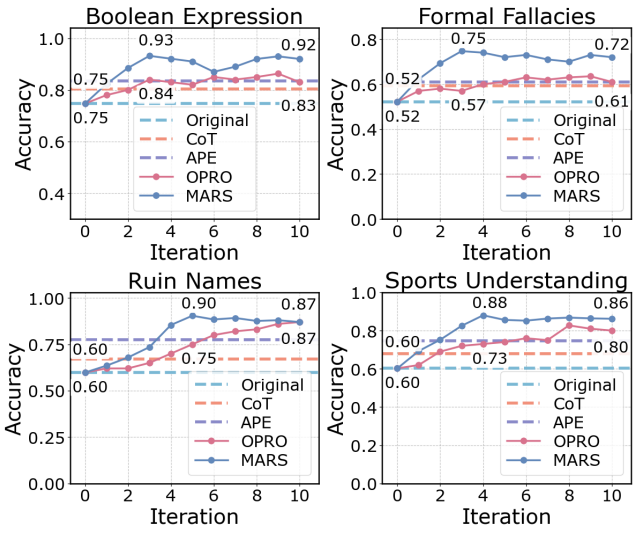

Figure 6: The convergence curves across different tasks show the learning progress as the number of iterations increases. We compare the iterative convergence process of MARS with four different baseline methods across four tasks to assess MARS's advantage in convergence speed.

Related Works

The related work is structured into two main aspects: first, an introduction to prompt optimization; and second, an exploration of multi-agent techniques.

Prompt Optimization. Early work primarily focused on two aspects: discrete optimization of hard prompts (Shin et al. 2020; Wen et al. 2024; Chen et al. 2023; Zhang et al. 2022) and continuous vector optimization of soft prompts (Lester, Al-Rfou, and Constant 2021; Li and Liang 2021; Liu et al. 2024). However, these methods are highly task-dependent and exhibit locality. With the advent of LLMs, traditional methods have become outdated. APE (Zhou et al. 2022) pioneered the use of generative methods to optimize instructions. Since APE, there have been two major approaches. The first approach (Zhou et al. 2022; Xu, Banburski-Fahey, and Jojic 2023; Pryzant et al. 2023; Wang et al. 2023) is the *generate-search* model, where multiple candidate sequences are generated, and methods like Monte Carlo search are used to optimize the prompt. The second approach (Yang et al. 2024a; Ye et al. 2023) is the *meta prompts* method, where sophisticated *meta prompts* are designed to optimize the prompt. In contrast to these two approaches, MARS employs a planned optimization path, iteratively generating high-quality prompts. This approach alleviates the inefficient search in prompt spaces issues in the first approach and addresses the challenges of limited flexibility of fixed templates in the second approach.

Multi-Agent. Based on LLMs, a combination of AI agents capable of performing specific functions forms a multi-agent system (Richards 2023; Yang, Yue, and He 2023; Wu et al. 2023; Zhang et al. 2025). Given a statement of a specific task, AI agents can attempt to break complex problem statements into subtasks and use tools, including data retrieval from the internet, to solve them step-by-step through automatic iterations. Some studies (Poldrack, Lu, and Beguš 2023; Wang et al. 2024a; Xi et al. 2025; Ni and Gao 2021; Lin et al. 2025b) use multi-agent systems to address issues such as problem identification, code development and debugging, plotting results and analysis, and providing interactive feedback with the human user. Other studies demonstrate the potential of organizing an AI multi-agent collaborative team to automatically solve mechanical problems, showcasing an enhanced ability to understand, formulate, and validate engineering problem solutions through self-correction and mutual correction (Ni and Buehler 2024; Zhang et al. 2026). Inspired by their work, we leverage multi-agent technology to autonomously plan the APO optimization path and design a *Teacher-Critic-Student* collaborative approach for iterative optimization.

Conclusion

We propose **MARS**, a novel multi-agent framework for adaptive APO that integrates Socratic guidance within a POMDP formulation. It includes: (1) a *Planner* that generates task-specific optimization trajectories, and (2) a *Teacher-Critic-Student* dialogue enabling interpretable prompt refinement. This simulates pseudo-gradient paths in discrete prompt space, narrowing the search scope. Modeled as a POMDP: the *Student*'s latent state is the hidden state, *Teacher-Critic* interactions define actions, and prompt outputs serve as observations. A *Target* agent guides iteration via performance rewards. Experiments show MARS consistently outperforms baselines while maintaining transparent optimization trajectories.

Acknowledgments

This research is supported by the Ministry of Education, Singapore under its MOE Academic Research Fund Tier 2 (MOE-T2EP20123-0005). The work is also supported by the National Natural Science Foundation of China (No. 62137002, 62277042, 62293553, 62450005, 62437002, 62477036, 62477037, 62176209, 62192781, 62306229), the "LENOVO-XJTU" Intelligent Industry Joint Laboratory Project, the Shaanxi Provincial Social Science Foundation Project (No. 2024P041), the Natural Science Basic Research Program of Shaanxi (No. 2023-JC-YB-593), and the Youth Innovation Team of Shaanxi Universities "Multi-modal Data Mining and Fusion".

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Chen, L.; Chen, J.; Goldstein, T.; Huang, H.; and Zhou, T. 2023. Instructzero: Efficient instruction optimization for black-box large language models. *arXiv preprint arXiv:2306.03082*.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Huang, Y.; Bai, Y.; Zhu, Z.; Zhang, J.; Zhang, J.; Su, T.; Liu, J.; Lv, C.; Zhang, Y.; Fu, Y.; et al. 2024. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *Advances in Neural Information Processing Systems*, 36.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Li, X. L.; and Liang, P. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Lin, Q.; Zhao, T.; He, K.; Peng, Z.; Xu, F.; Huang, L.; Ma, J.; and Feng, M. 2025a. Self-supervised Quantized Representation for Seamlessly Integrating Knowledge Graphs with Large Language Models. *arXiv preprint arXiv:2501.18119*.

Lin, Q.; Zhu, Y.; Pu, B.; Huang, L.; Luo, H.; Ma, J.; Peng, Z.; Zhao, T.; Xu, F.; Zhang, J.; et al. 2025b. A Foundation Model for Chest X-ray Interpretation with Grounded Reasoning via Online Reinforcement Learning. *arXiv preprint arXiv:2509.03906*.

Liu, W.; Luo, H.; Lin, X.; Liu, H.; Shen, T.; Wang, J.; Mao, R.; and Cambria, E. 2025. Prompt-R1: Collaborative Automatic Prompting Framework via End-to-end Reinforcement Learning. arXiv:2511.01016.

Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; and Tang, J. 2024. GPT understands, too. *AI Open*, 5: 208–215.

Ni, B.; and Buehler, M. J. 2024. MechAgents: Large language model multi-agent collaborations can solve mechanics problems, generate new data, and integrate knowledge. *Extreme Mechanics Letters*, 67: 102131.

Ni, B.; and Gao, H. 2021. A deep learning approach to the inverse problem of modulus identification in elasticity. *Mrs Bulletin*, 46: 19–25.

Poldrack, R. A.; Lu, T.; and Beguš, G. 2023. AI-assisted coding: Experiments with GPT-4. *arXiv preprint arXiv:2304.13187*.

Pryzant, R.; Iter, D.; Li, J.; Lee, Y. T.; Zhu, C.; and Zeng, M. 2023. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, (EMNLP)*, 7957–7968.

Richards, T. B. 2023. Auto-GPT: An experimental open-source attempt to make GPT-4 fully autonomous.

Shen, T.; Mao, R.; Wang, J.; Zhang, X.; and Cambria, E. 2025. Flow-guided Direct Preference Optimization for Knowledge Graph Reasoning with Trees. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, 1165–1175. New York, NY, USA: Association for Computing Machinery. ISBN 9798400715921.

Shin, T.; Razeghi, Y.; Logan IV, R. L.; Wallace, E.; and Singh, S. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Suzgun, M.; Scales, N.; Schärli, N.; Gehrmann, S.; Tay, Y.; Chung, H. W.; Chowdhery, A.; Le, Q. V.; Chi, E. H.; Zhou, D.; et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Wang, L.; Ma, C.; Feng, X.; Zhang, Z.; Yang, H.; Zhang, J.; Chen, Z.; Tang, J.; Chen, X.; Lin, Y.; et al. 2024a. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6): 186345.

Wang, X.; Li, C.; Wang, Z.; Bai, F.; Luo, H.; Zhang, J.; Jojic, N.; Xing, E. P.; and Hu, Z. 2023. Promptagent: Strategic planning with language models enables expert-level prompt optimization. *arXiv preprint arXiv:2310.16427*.

Wang, Y.; Ma, X.; Zhang, G.; Ni, Y.; Chandra, A.; Guo, S.; Ren, W.; Arulraj, A.; He, X.; Jiang, Z.; et al. 2024b. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.

Wen, Y.; Jain, N.; Kirchenbauer, J.; Goldblum, M.; Geiping, J.; and Goldstein, T. 2024. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *Advances in Neural Information Processing Systems*, 36.

Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Zhang, S.; Zhu, E.; Li, B.; Jiang, L.; Zhang, X.; and Wang, C. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*.

Xi, Z.; Chen, W.; Guo, X.; He, W.; Ding, Y.; Hong, B.; Zhang, M.; Wang, J.; Jin, S.; Zhou, E.; et al. 2025. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2): 121101.

Xu, F.; Wu, Z.; Sun, Q.; Ren, S.; Yuan, F.; Yuan, S.; Lin, Q.; Qiao, Y.; and Liu, J. 2024. Symbol-LLM: Towards foundational symbol-centric interface for large language models. In *Proceedings of the ACL*, 13091–13116.

Xu, W.; Banburski-Fahey, A.; and Jojic, N. 2023. Reprompting: Automated chain-of-thought prompt inference through gibbs sampling. *arXiv preprint arXiv:2305.09993*.

Yan, H.; Xu, F.; Xu, R.; Li, Y.; Zhang, J.; Luo, H.; Wu, X.; Tuan, L. A.; Zhao, H.; Lin, Q.; et al. 2025. Mur: Momentum uncertainty guided reasoning for large language models. *arXiv preprint arXiv:2507.14958*.

Yang, C.; Wang, X.; Lu, Y.; Liu, H.; Le, Q. V.; Zhou, D.; and Chen, X. 2024a. Large Language Models as Optimizers. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yang, H.; Yue, S.; and He, Y. 2023. Auto-gpt for online decision making: Benchmarks and additional opinions. *arXiv preprint arXiv:2306.02224*.

Yang, J.; Jin, H.; Tang, R.; Han, X.; Feng, Q.; Jiang, H.; Zhong, S.; Yin, B.; and Hu, X. 2024b. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6): 1–32.

Ye, Q.; Axmed, M.; Pryzant, R.; and Khani, F. 2023. Prompt engineering a prompt engineer. *arXiv preprint arXiv:2311.05661*.

Yuan, L.; Cai, Y.; Shen, X.; Li, Q.; Huang, Q.; Deng, Z.; and Wang, T. 2025. Collaborative Multi-LoRA Experts with Achievement-based Multi-Tasks Loss for Unified Multimodal Information Extraction. In Kwok, J., ed., *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, 6940–6948. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Zhang, H.; Da, J.; Lee, D.; Robinson, V.; Wu, C.; Song, W.; Zhao, T.; Raja, P.; Slack, D.; Lyu, Q.; et al. 2024a. A careful examination of large language model performance on grade school arithmetic. *arXiv preprint arXiv:2405.00332*.

Zhang, J.; Wang, Z.; Wang, Z.; Zhang, X.; Xu, F.; Lin, Q.; Mao, R.; Cambria, E.; and Liu, J. 2026. MAPS: A Multi-Agent Framework Based on Big Seven Personality and Socratic Guidance for Multimodal Scientific Problem Solving. In *Proceedings of AAAI*.

Zhang, J.; Wei, B.; Qi, S.; Liu, J.; Lin, Q.; et al. 2025. GKG-LLM: A Unified Framework for Generalized Knowledge Graph Construction. *arXiv preprint arXiv:2503.11227*.

Zhang, J.; Yang, C.; Zhu, H.; Lin, Q.; Xu, F.; and Liu, J. 2024b. A Semantic Mention Graph Augmented Model for Document-Level Event Argument Extraction. *arXiv preprint arXiv:2403.09721*.

Zhang, T.; Wang, X.; Zhou, D.; Schuurmans, D.; and Gonzalez, J. E. 2022. Tempera: Test-time prompting via reinforcement learning. *arXiv preprint arXiv:2211.11890*.

Zhong, W.; Cui, R.; Guo, Y.; Liang, Y.; Lu, S.; Wang, Y.; Saied, A.; Chen, W.; and Duan, N. 2023. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*.

Zhou, Y.; Muresanu, A. I.; Han, Z.; Paster, K.; Pitis, S.; Chan, H.; and Ba, J. 2022. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*.