MAPGD: Multi-Agent Prompt Gradient Descent for Collaborative Prompt Optimization

Yichen Han[1]*, Bojun Liu[2]*, Zhengpeng Zhou[3]*, Guanyu Liu[4], Zeng Zhang[1],
Yang Yang[5], Wenli Wang[5], Isaac N Shi[5], Yunyan[5], Lewei He[1]†, Tianyu Shi[6]†
[1]South China Normal University [2]University of Sydney
[3]Shanghai Jiaotong University [4]University of Macau
[5]Silicon Sapiens LLC [6]University of Toronto
helewei@m.scnu.edu.cn, shi@mail.utoronto.ca

Abstract

Prompt engineering is crucial for leveraging large language models (LLMs), but existing methods often rely on a single optimization trajectory, limiting adaptability and efficiency while suffering from narrow perspectives, gradient conflicts, and high computational cost. We propose **MAPGD (Multi-Agent Prompt Gradient Descent)**, a framework integrating multi-agent collaboration with gradient-based optimization. MAPGD features specialized agents for task clarity, example selection, format design, and stylistic refinement; semantic gradient coordination to resolve conflicts; bandit-based candidate selection for efficient exploration-exploitation; and theoretical convergence guarantees. Experiments on classification, generation, and reasoning tasks show MAPGD outperforms single-agent and random baselines in accuracy and efficiency. Ablations confirm the benefits of gradient fusion, agent specialization, and conflict resolution, providing a unified, gradient-inspired multi-agent approach to robust and interpretable prompt optimization.

1 Introduction

Recent advances in large language models (LLMs) have demonstrated that while increased scale enhances generalization, it also amplifies sensitivity to prompts. Minor variations in wording, structure, or phrasing can induce substantial shifts in model outputs, revealing the inherent brittleness of existing prompt-based interactions. This observation underscores the critical need for robust and efficient prompt optimization strategies. Traditional approaches, including manual engineering and random search, often suffer from inefficiency, inconsistency, and limited scalability. Even more sophisticated single-agent, gradient-inspired optimization methods are constrained by their reliance on a single trajectory, which inherently restricts their adaptability and introduces conflicts among competing improvement signals.

To address these limitations, we introduce **MAPGD (Multi-Agent Prompt Gradient Descent)**, a framework that reconceptualizes prompt optimization as a collaborative, multi-agent process. MAPGD draws inspiration from human team dynamics, where complementary expertise is distributed across specialized agents, each dedicated to refining a distinct aspect of the prompt, including instructional clarity, example selection, format structuring, and stylistic adaptation. Operating in parallel, these agents generate specialized gradients that collectively capture multi-faceted improvement signals. To reconcile heterogeneous updates, MAPGD employs a semantic gradient coordinator that projects textual feedback into a shared semantic embedding space, enabling systematic conflict

*Equal contribution.
†Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: SEA.

detection, semantic clustering, and fusion of competing signals. This principled integration preserves coherence while effectively guiding collaborative prompt descent.This process is illustrated in Figure 1.

Candidate prompts are further refined through a combination of beam search with Monte Carlo sampling and a bandit-based selection mechanism, which dynamically balances exploration and exploitation under constrained evaluation budgets. From a theoretical perspective, MAPGD offers provable convergence guarantees, demonstrating almost sure convergence to a local optimum at a rate of $\mathcal{O}(1/\sqrt{T})$. Although multi-agent collaboration introduces additional computational overhead, semantic fusion and bandit-based selection significantly reduce redundant evaluations compared to exhaustive search or single-agent baselines. Extensive empirical studies across classification, generation, and reasoning tasks validate MAPGD's effectiveness, showing superior performance, improved efficiency-per-token, and reduced evaluation cost. Beyond practical gains, MAPGD advances the conceptual foundation of prompt optimization by bridging gradient-based learning principles with multi-agent cooperation, establishing a scalable and robust paradigm for aligning LLMs with complex human intentions.

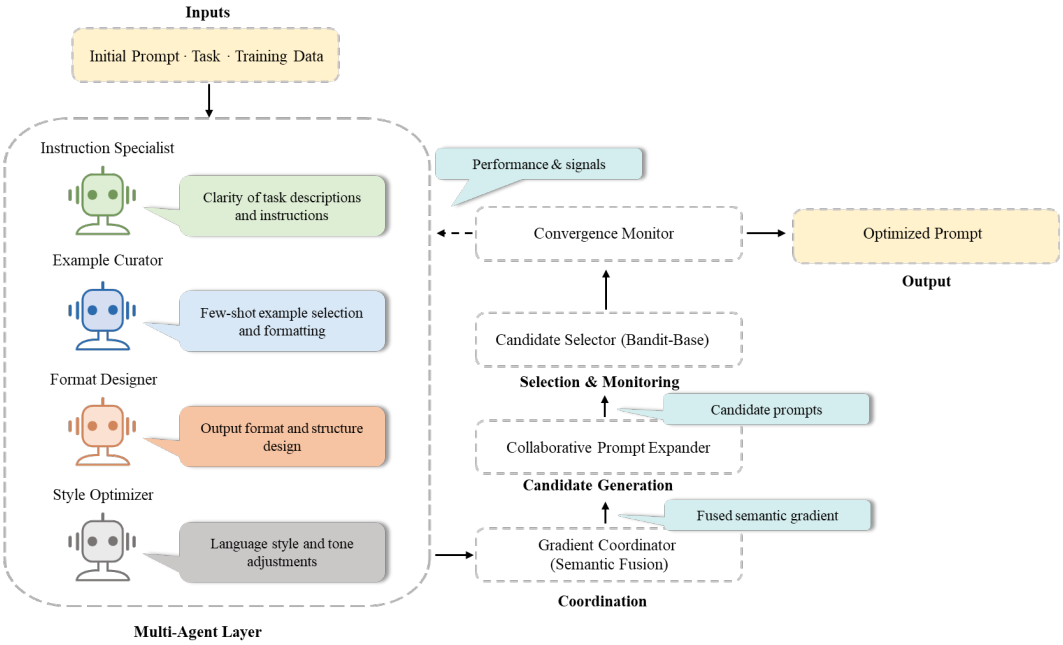


Figure 1: Overview of the MAPGD framework, illustrating the multi-agent collaboration from initial prompt to optimized prompt via specialized agents, performance signals, candidate prompts, and fused semantic gradients.

2 Related Work

Prompt optimization lies at the intersection of several active research directions.

Prompt Learning and Optimization. The rise of large language models has spurred a surge of interest in prompt design. Manual prompt engineering, while effective in some cases, lacks scalability. Automated approaches, including gradient-free methods such as reinforcement learning (RLHF Deng et al. (2022)), evolutionary search (Fernando et al., 2023), and Bayesian optimization, attempt to navigate the discrete prompt space more systematicallySahoo et al. (2025). Meanwhile, continuous prompt tuning methods (e.g., prefix tuning (Li, Liang, 2021), soft prompts (Lester et al., 2021)) optimize embeddings rather than natural language, which limits interpretability. MAPGD instead focuses on interpretable natural language optimization, leveraging structured feedback to improve both human readability and model alignment (Prasad et al., 2022).

Multi-Agent Collaboration. Multi-agent systems have long been studied in reinforcement learning, distributed AI, and game theory. In the context of NLP, recent works have explored multi-agent debate (Liang et al., 2023; Du et al., 2023), cooperative reasoning, and collaborative content generation (Hong et al., 2024). These systems demonstrate that agent specialization—assigning roles with distinct expertise—can outperform monolithic optimization. MAPGD inherits this principle, assigning agents to distinct prompt dimensions and coordinating their outputs via gradient fusion (Wu et al., 2023).

Gradient-Inspired Prompt Descent. ProTeGi (Pryzant et al., 2023) and related frameworks approximate prompt gradients through LLM self-feedback, iteratively refining prompts based on performance errors. While effective, single-agent gradient descent struggles with signal diversity and conflict resolution (Xiang et al., 2025). MAPGD advances this line by introducing semantic gradient embedding, enabling conflict detection through cosine similarity and fusion through LLM-based synthesis (Pryzant et al., 2023).

In summary, MAPGD contributes a unified framework that connects prompt learning, multi-agent collaboration, and gradient descent principles, offering a scalable and interpretable pathway for robust prompt optimization (Li, Liang, 2021; Lester et al., 2021; Deng et al., 2022; Liang et al., 2023; Pryzant et al., 2023).

3 Methodology

3.1 Framework Overview

MAPGD conceptualizes prompt optimization as a hybrid discrete-continuous gradient descent process in the space of natural language prompts. Unlike continuous embeddings used in soft prompt tuning, MAPGD explicitly operates on interpretable textual prompts while leveraging gradient-inspired signals for refinement.

In MAPGD, the definition of the prompt search space departs fundamentally from traditional approaches that rely on static enumeration or random sampling. Instead, MAPGD constructs the search space dynamically by leveraging multi-agent gradient generation and adaptive filtering. At each iteration, specialized agents analyze misclassified or suboptimal examples and propose improvement directions in the form of textual gradients. These gradients are semantically vectorized, clustered, and fused to mitigate conflicts and ensure coherent optimization trajectories. The fused gradients are then expanded into a pool of successor prompts, from which a bandit-based strategy selects the most promising candidates under computational constraints. Iterative repetition of this cycle results in a search space that evolves as a sequence of adaptively curated prompts, effectively balancing exploration of diverse modifications with exploitation of high-performing solutions. This dynamic formulation not only reduces the computational burden compared to exhaustive search but also guarantees that the search trajectory remains aligned with task-specific optimization goals(see Figure 2).

Formally, we define the optimization objective as:

$$F(p) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\ell(M(x;p),y)\right] \tag{1}$$

where $\ell(\cdot,\cdot)$ is a task-specific loss function (e.g., cross-entropy for classification, negative ROUGE for summarization). The optimization seeks:

$$p^* = \arg\min_p F(p). \tag{2}$$

Unlike stochastic gradient descent (SGD), where gradients are continuous and computed analytically, MAPGD constructs pseudo-gradients from agent feedback:

$$\nabla F(p^{(t)}) \approx g^{(t)}, \tag{3}$$

but in textual form. These pseudo-gradients act as semantic analogues of numerical gradients, guiding structured textual refinements.

3.2 Specialized Prompt Agents

Each agent is specialized in one dimension of optimization, mimicking orthogonal gradient directions in parameter space. For example:A_1: clarity of task instruction (g_1);A_2: example selection (g_2);A_3: format enforcement (g_3); A_4: stylistic refinement (g_4).

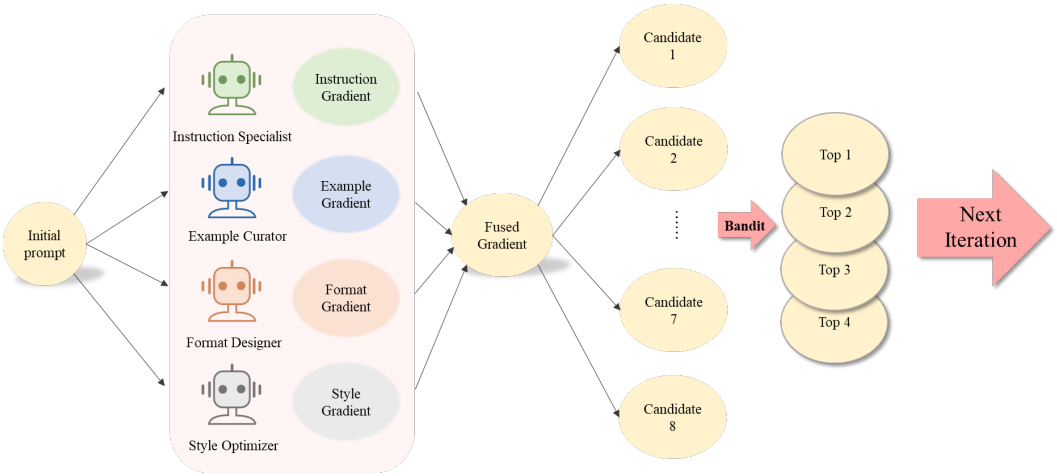


Figure 2: Illustration of the MAPGD prompt search space. Starting from an initial prompt, multi-agent gradients are fused to generate candidate prompts, which are then filtered by bandit-based selection before entering the next iteration.

Thus, the multi-agent gradient set at iteration t is: $G^{(t)} = \{g_1^{(t)}, g_2^{(t)}, \ldots, g_K^{(t)}\}$.

This decomposition enables parallel exploration of multiple optimization directions, mitigating the local minima problem that plagues single-agent methods.

3.3 Semantic Gradient Coordination

MAPGD introduces semantic gradient embeddings to reconcile conflicting signals. Each gradient is embedded via:

$$v_k^{(t)} = \phi(g_k^{(t)}), \quad v_k^{(t)} \in \mathbb{R}^d, \tag{4}$$

where ϕ is a pre-trained encoder (e.g., Sentence-BERT) (Xu et al., 2023).

Conflicts are identified when:

$$\text{sim}(v_i, v_j) = \frac{v_i \cdot v_j}{\|v_i\|\|v_j\|} < -\theta \tag{5}$$

In such cases, gradients are fused using a weighted scheme:

$$g_{\text{fused}}^{(t)} = \Psi\left(\sum_{k=1}^{K} w_k g_k^{(t)}\right), \quad w_k = \frac{\exp(\lambda \cdot s_k)}{\sum_j \exp(\lambda \cdot s_j)}, \tag{6}$$

where s_k is the validation score improvement of $g_k^{(t)}$, and λ controls sharpness.The process is illustrated in Figure 3b.

3.4 Bandit-Based Candidate Selection

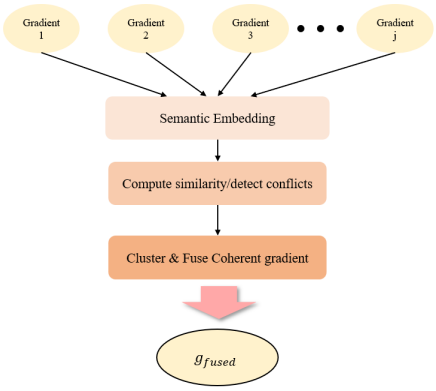
To ensure computational efficiency, MAPGD employs a multi-armed bandit (MAB) approach (Bouneffouf, 2016). The algorithm is illustrated in Figure 3b.Given candidate prompts $\{p_j^{(t+1)}\}$, the expected reward is as follows:

$$r_j = \mathbb{E}_{(x,y)\sim B}\left[\mathbb{I}(M(x; p_j) = y)\right], \tag{7}$$

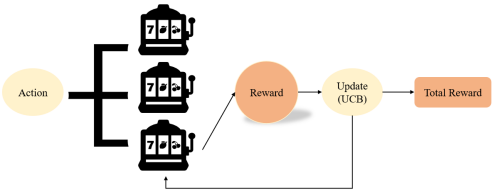
where $B \subset \mathcal{D}$ is a minibatch.

We apply UCB1 selection:

$$j^* = \arg\max_j \left(\hat{r}_j + \sqrt{\frac{2 \ln t}{n_j}}\right). \tag{8}$$



(a) Semantic gradient coordination process: gradients from multiple agents are embedded, similarity is computed to detect conflicts, and coherent gradients are clustered and fused into g_{fused}.



(b) Illustration of the multi-armed bandit process: An agent selects an action (pulls one of several slot machines), receives a reward, and updates its value estimates using the Upper Confidence Bound (UCB) algorithm. The iterative process accumulates into maximizing the total reward over time.

Figure 3: Overview of two methods: (a) Semantic gradient coordination process; (b) Multi-armed bandit selection process.

Algorithm 1 MAPGD Iterative Optimization Loop

Require: Initial prompt p_0, train data D_{train}, dev data D_{dev}, agents $\{A_i\}_{i=1}^{N}$, iterations R, beam width k

Initialize $p_\star^{(0)} \leftarrow p_0$, $B_0 \leftarrow \{p_0\}$; initialize each agent prompt $A_i.p \leftarrow p_0$
for $t = 1$ to R **do**
  $M_t \leftarrow \textsc{SampleMiniBatch}(D_{\text{train}}, b)$
  $\mathcal{G}_t \leftarrow \textsc{GenerateAgentGradients}(\{A_i\}, M_t)$ ▷ Alg. 2
  $\tilde{\mathcal{G}}_t \leftarrow \textsc{CoordinateAndFuse}(\mathcal{G}_t)$ ▷ Alg. 3
  $\mathcal{C}_t \leftarrow \textsc{ExpandPrompts}(p_\star^{(t-1)}, \tilde{\mathcal{G}}_t)$ ▷ Alg. 4
  $B_t, p_\star^{(t)} \leftarrow \textsc{BanditSelect}(\mathcal{C}_t, D_{\text{dev}}, k)$
  $\textsc{SynchronizeAgents}(\{A_i\}, p_\star^{(t)})$
  if $\textsc{Converged}(p_\star^{(t)})$ **then break**
  end if
end for
return Best prompt over $\{p_\star^{(1)}, \ldots, p_\star^{(t)}\}$

This balances exploration (testing diverse prompt candidates) and exploitation (refining promising prompts).

Algorithm 1 summarizes the end-to-end control logic.

3.5 Complexity and Parallelism

We summarize per–iteration costs; LLM generations dominate, while vector operations are minor. Notation: N agents, m reasons/agent $\Rightarrow$ $G = Nm$ atomic gradients; embedding dim d; clusters $K \leq G$; fused gradients $|\tilde{\mathcal{G}}|$; variants/gradient s; MC paraphrases/variant n_{mc}; candidates $|\mathcal{C}|$; beam width k; bandit rounds T_b with K_{eval} arms and dev mini-batch size b.

Stage costs (time).

1. Agent gradient generation: $O(N)$ LLM calls; parallelization reduces wall time to $\approx \max t_{\text{LLM}}$.
2. Embedding + conflict checks: $O(Gd) + O(G^2 d)$ (tiny as G is small, e.g. 16).
3. Clustering: $O(GKId)$ (negligible for small G, $I < 20$).

Algorithm 2 Multi-Agent Textual Gradient Generation

Require: Agents $\{A_i\}$, mini-batch M_t, task T, predictor Π, per-agent error cap e, feedback count m

for all agent A_i in parallel **do**
  $(\hat{y}, y)$ pairs $\leftarrow T.\textsc{InferAndLabel}(A_i.p, M_t, \Pi)$
  $E_i \leftarrow \textsc{SelectErrors}(\hat{y}, y, e)$
  if $|E_i| = 0$ **then** $E_i \leftarrow \textsc{DiverseSamples}(M_t, e)$
  end if
  $raw_i \leftarrow \text{LLM}\textsc{GradientPrompt}(A_i.role, A_i.p, E_i, m)$
  $g_i \leftarrow \textsc{ParseGradientBlocks}(raw_i)$ ▷ Split by delimiters
end for
return $\mathcal{G}_t = \{(A_i.role, g_i)\}_{i=1}^{N}$

Algorithm 3 Gradient Coordination and Fusion

Require: Gradient dict $\mathcal{G}_t$, similarity threshold θ, max clusters $K_{\max}$

$\mathcal{R} \leftarrow$ flatten all atomic reasons
$V \leftarrow \phi(\mathcal{R})$ ▷ Batch embeddings
$\mathcal{C}_{\text{conf}} \leftarrow \{(r_a, r_b) : \text{sim}(v_a, v_b) < -\theta\}$
$K \leftarrow \min(K_{\max}, |\mathcal{R}|)$
$\{S_k\}_{k=1}^{K} \leftarrow \text{KM}\textsc{eans}(V, K)$
$F \leftarrow \emptyset$
for $k = 1$ to K **do**
  if $|S_k| = 1$ **then** append unique reason to F
  else
    $meta \leftarrow \textsc{BuildFusionPrompt}(S_k, \mathcal{C}_{\text{conf}})$
    $f_k^{raw} \leftarrow \text{LLM}\textsc{Fuse}(meta)$
    $f_k \leftarrow \textsc{ParseFusion}(f_k^{raw})$
    Append f_k to F
  end if
end for
return $\tilde{\mathcal{G}}_t \leftarrow F$

4. Fusion (multi–item clusters only): up to $O(K_{\text{merge}})$ LLM calls (parallelizable).
5. Expansion + MC: $O(|\tilde{\mathcal{G}}|)$ gradient applications + $O(|\tilde{\mathcal{G}}| sn_{\text{mc}})$ paraphrases (LLM bound).
6. Diversity filtering: embeddings $O(|\mathcal{C}|d)$; naive pairwise $O(|\mathcal{C}|^2)$ (acceptable for tens).
7. Bandit evaluation: $O(K_{\text{eval}} b T_b)$ model probes vs. exhaustive $O(|\mathcal{C}|\,|D_{\text{dev}}|)$.

Space. $O(|\mathcal{C}|L_{\text{avg}})$ text + $O((G + |\mathcal{C}|)d)$ embeddings (few MB). Optional caches scale with unique prompts.

4 Experiments

We evaluate MAPGD on three representative datasets: **LIAR** for fact-checking, **Jailbreak** for adversarial robustness, and **Ethos** for hate speech detection.

This diverse setting enables us to examine MAPGD's effectiveness under different prompt optimization challenges. Our experiments are designed to answer the following questions:

(i) How does MAPGD perform compared with existing baselines?

(ii) What is the contribution of multi-agent collaboration and semantic gradient fusion?

(iii) How robust is MAPGD across different datasets and optimization budgets?

Unless otherwise noted, we adopt **four specialized agents** and run **ten optimization iterations** by default.

Algorithm 4 Gradient-Guided Prompt Expansion

Require: Base prompt p, fused gradients $\tilde{\mathcal{G}}_t$, successor cap S, MC samples n_{mc}, diversity margin δ
$C \leftarrow \emptyset$
for all $\tilde{g} \in \tilde{\mathcal{G}}_t$ **do**
    $variants \leftarrow \text{LLMAPPLYGRADIENT}(p, \tilde{g})$
    $C \leftarrow C \cup variants$
end for
for all $c \in C$ **do**
    $mc\_set \leftarrow \text{PARAPHRASEMC}(c, n_{\text{mc}})$
    $C \leftarrow C \cup mc\_set$
end for
$C \leftarrow \text{SEMANTICFILTER}(C, \delta)$
$C \leftarrow \text{TRUNCATE}(C, S)$
return C

4.1 Overall Performance

Table 1 compares MAPGD with ProTeGi (Pryzant et al., 2023) and Monte-Carlo (Zhou et al., 2022) optimization across three datasets: LIAR, Jailbreak, and Ethos. MAPGD consistently achieves the best F1 scores on all datasets. Specifically, MAPGD improves from 0.62 (MC) and 0.64 (ProTeGi) to 0.71 on LIAR, from 0.76 (MC) and 0.81 (ProTeGi) to 0.88 on Jailbreak, and from 0.94 (MC) and 0.95 (ProTeGi) to 0.98 on Ethos. These results highlight the effectiveness of multi-agent collaboration and semantic gradient fusion in enhancing optimization across diverse tasks.

Table 1: Performance comparison of MAPGD, ProTeGi, and Monte-Carlo across three datasets.

Method	LIAR (F1)	Jailbreak (F1)	Ethos (F1)
Monte-Carlo (MC)	0.62	0.76	0.94
ProTeGi (baseline)	0.64	0.81	0.95
MAPGD (Ours)	**0.71**	**0.88**	**0.98**

4.2 Ablation on Bandit Strategies

To examine the effect of different exploration–exploitation strategies, we replace the default UCB with Thompson Sampling and Greedy. As shown in Table 2, UCB achieves the best performance (0.6844), while Thompson Sampling drops to 0.63 and Greedy further degrades to 0.56. This indicates that UCB provides a more principled trade-off between exploration and exploitation under constrained evaluation budgets.

Table 2: Performance comparison under different bandit strategies (test=150, train=50).

Bandit Strategy	Best F1 Score
UCB	**0.6844**
Thompson	0.6300
Greedy	0.5600

4.3 Ablation on Search Strategies

We further compare search algorithms under the UCB framework. Table 3 shows that beam search achieves a significantly higher F1 (0.6844) than Monte Carlo sampling (0.50). This suggests that structured search is more effective than purely stochastic exploration in identifying high-quality prompts.

Table 3: Comparison of search algorithms under UCB (test=150, train=50).

Search Strategy	Best F1 Score
Beam Search	**0.6844**
Monte Carlo	0.5000

4.4 Experimental Insights

Across all experiments, three consistent findings emerge:

1. **Multi-agent specialization is essential.** MAPGD consistently surpasses single-agent optimization, confirming that orthogonal expertise enables more comprehensive prompt refinement.
2. **Semantic gradient fusion enhances coherence.** The integration of heterogeneous signals avoids conflicting updates and yields stable improvements, especially in reasoning-heavy tasks.
3. **Bandit-based selection ensures efficiency.** UCB with beam search reduces redundant evaluations while maintaining superior accuracy, highlighting the importance of budget-aware optimization.

Collectively, these results validate MAPGD as a robust and efficient framework for prompt optimization.

5 Discussion

MAPGD contributes to the broader discourse on interpretable prompt optimization by showing that gradient-inspired reasoning can be effectively combined with multi-agent collaboration. This design paradigm has several implications.

Modularity. MAPGD decomposes prompt optimization into orthogonal dimensions, enabling transparent analysis of which aspects of a prompt contribute most to performance. Such modularity is often absent in end-to-end continuous prompt tuning.

Coordination. By embedding gradient signals into a semantic vector space, MAPGD explicitly reasons about conflicts and complementarities among agents. This resonates with recent advances in multi-gradient optimization in deep learning, where task interference is addressed via gradient surgery. The analogy suggests that natural language prompt optimization can inherit principles from multi-task learning.

Budget-awareness. MAPGD also provides a resource-conscious optimization framework. Unlike settings where model access is unlimited, practical LLM applications face strict token and API usage constraints. MAPGD's bandit selection mechanism ensures that optimization progress is achieved under realistic computational budgets.

Theoretical grounding. Beyond empirical results, we provide a convergence analysis in Appendix A, establishing that MAPGD achieves a sublinear rate of $O(1/\sqrt{T})$ under standard stochastic approximation assumptions. This bridges discrete prompt optimization with classical guarantees of stochastic gradient descent.

Despite these strengths, MAPGD still faces challenges. The reliance on embeddings for conflict resolution may be fragile under domain shift or adversarial prompt distributions. Moreover, the pseudo-gradient approximation depends on the quality of agent feedback, which remains an open challenge in aligning LLM self-evaluations with ground-truth task metrics.

We also present a case study in Appendix B, where MAPGD is applied to optimize a system prompt for a large language model assistant, demonstrating its utility in real-world applications, particularly in domains that require data authenticity, verification, and contextual accuracy.

Future work may explore:

1. Cross-task generalization: training reusable gradient agents that transfer knowledge across domains.
2. Human-in-the-loop optimization: incorporating expert oversight to guide semantic gradient fusion.
3. Hybrid discrete-continuous search: combining MAPGD with differentiable prompt tuning for joint interpretability and efficiency.

6 Conclusion

In this work, we presented MAPGD: Multi-Agent Prompt Gradient Descent, a new paradigm for optimizing prompts in large language models. By decomposing prompt optimization into orthogonal semantic dimensions, MAPGD enables diverse and parallel exploration. Through semantic gradient embeddings and fusion, the framework resolves conflicts among agents, while bandit-based selection ensures budget-aware efficiency.

Beyond empirical validation, we also provide *theoretical guarantees* for MAPGD. Under mild stochastic approximation assumptions, we prove that MAPGD achieves a sublinear convergence rate of $O(1/\sqrt{T})$ in both convex and non-convex settings. This establishes MAPGD on firm mathematical footing, showing that despite operating in a discrete prompt space, its semantic gradient mechanism preserves the efficiency of classical stochastic gradient methods. The convergence analysis highlights how semantic alignment and variance control—enforced respectively by gradient fusion and bandit-based sampling—are key to ensuring stability.

MAPGD thus contributes to the growing field of interpretable prompt learning, offering both **practical advances** in multi-agent prompt optimization and **theoretical insights** into its convergence behavior. While challenges remain—such as dependency on embedding models and sensitivity to LLM feedback—MAPGD establishes a foundation for future research. Possible directions include cross-task generalization of gradient agents, integration with human preference alignment, and extensions to multimodal prompts.

We believe MAPGD represents a step toward more robust, interpretable, and efficient prompt optimization, contributing practical solutions for real-world deployment and theoretical guarantees for the study of language model alignment.

References

Bottou Léon, Curtis Frank E., Nocedal Jorge. Optimization Methods for Large-Scale Machine Learning. 2018.

Bouneffouf Djallel. Finite-time analysis of the multi-armed bandit problem with known trend // 2016 IEEE Congress on Evolutionary Computation (CEC). 2016. 2543–2549.

Deng Mingkai, Wang Jianyu, Hsieh Cheng-Ping, Wang Yihan, Guo Han, Shu Tianmin, Song Meng, Xing Eric P, Hu Zhiting. Rlprompt: Optimizing discrete text prompts with reinforcement learning // arXiv preprint arXiv:2205.12548. 2022.

Du Yilun, Li Shuang, Torralba Antonio, Tenenbaum Joshua B, Mordatch Igor. Improving factuality and reasoning in language models through multiagent debate // Forty-first International Conference on Machine Learning. 2023.

Fernando Chrisantha, Banarse Dylan, Blundell Charles, Rocktäschel Tim, Osindero Simon. PromptBreeder: Self-Referential Self-Improvement Via Prompt Evolution // arXiv preprint arXiv:2309.16797. 2023.

Hong Sirui, Zhuge Mingchen, Chen Jiaqi, Zheng Xiawu, Cheng Yuheng, Zhang Ceyao, Wang Jinlin, Wang Zili, Yau Steven Ka Shing, Lin Zijuan, Zhou Liyang, Ran Chenyu, Xiao Lingfeng, Wu Chenglin, Schmidhuber Jürgen. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. 2024.

Lester Brian, Al-Rfou Rami, Constant Noah. The power of scale for parameter-efficient prompt tuning // arXiv preprint arXiv:2104.08691. 2021.

Li Xiang Lisa, Liang Percy. Prefix-Tuning: Optimizing Continuous Prompts for Generation // Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2021. 4582–4597.

Liang Weizhe, Wu Yizhong, Lee Mina, Chi Ed, Zhou Denny, Song Yiming, Li Xiang Lisa. Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate // arXiv preprint arXiv:2305.19118. 2023.

Prasad Archiki, Hase Peter, Zhou Xiang, Bansal Mohit. Grips: Gradient-free, edit-based instruction search for prompting large language models // arXiv preprint arXiv:2203.07281. 2022.

Pryzant Reid, Jain Shauli, al. et. Automatic Prompt Optimization with Gradient Descent and Beam Search // arXiv preprint arXiv:2304.08442. 2023.

Sahoo Pranab, Singh Ayush Kumar, Saha Sriparna, Jain Vinija, Mondal Samrat, Chadha Aman. A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications. 2025.

Wu Yizhou, Wang Tianhao, al. et. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. 2023. Microsoft Research Project.

Xiang Jinyu, Zhang Jiayi, Yu Zhaoyang, Liang Xinbing, Teng Fengwei, Tu Jinhao, Ren Fashen, Tang Xiangru, Hong Sirui, Wu Chenglin, Luo Yuyu. Self-Supervised Prompt Optimization. 2025.

Xu Jiahao, Shao Wei, Chen Lihui, Liu Lemao. SimCSE++: Improving Contrastive Learning for Sentence Embeddings from Two Perspectives // Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. 2023. 12028–12040.

Zhou Yongchao, Muresanu Andrei Ioan, Han Ziwen, Paster Keiran, Pitis Silviu, Chan Harris, Ba Jimmy. Large language models are human-level prompt engineers // The eleventh international conference on learning representations. 2022.

A Theoretical Analysis

In this section, we provide convergence guarantees for MAPGD under mild assumptions, following the stochastic approximation framework. Our goal is to bridge the gap between the continuous optimization theory of stochastic gradient descent (SGD) and the discrete prompt optimization carried out in MAPGD. We show that, despite operating in a structured and discrete search space, MAPGD achieves the same sublinear convergence rate of $O(1/\sqrt{T})$ in both convex and non-convex settings.

A.1 Assumptions

We begin with a set of assumptions standard in stochastic optimization but reinterpreted in the context of multi-agent prompt optimization.

- **(A1) Alignment (Unbiasedness).** For some $\mu > 0$, the stochastic semantic gradient $g^{(t)}$ maintains alignment with the true gradient:

$$\mathbb{E}\Big[\langle g^{(t)}, \nabla F(p^{(t)})\rangle \,\Big|\, p^{(t)}\Big] \;\geq\; \mu \|\nabla F(p^{(t)})\|^2.$$

  This reflects the role of semantic fusion: multi-agent aggregation reduces the chance of adversarial or noisy updates, ensuring progress along descent directions.

- **(A2) Bounded Second Moment.** For constants $\rho, \sigma^2 \geq 0$,

$$\mathbb{E}\Big[\|g^{(t)}\|^2 \,\Big|\, p^{(t)}\Big] \;\leq\; \rho \|\nabla F(p^{(t)})\|^2 + \sigma^2.$$

  This captures the variance-control effect of the bandit-based selection mechanism, which prevents uncontrolled explosion of gradient magnitude.

- **(A3) Smoothness or Lipschitzness.** For convex tasks, F is G-Lipschitz with domain diameter D. For non-convex tasks, F is L-smooth: $\|\nabla F(u) - \nabla F(v)\| \leq L\|u - v\|$.

A.2 Supporting Lemmas

We restate two standard lemmas, adapted to the MAPGD setting.

Lemma 1 (Convex Projection Inequality). For convex F with feasible set $\mathcal{P}$, the projected subgradient update

$$p^{(t+1)} = \Pi_{\mathcal{P}}\Big(p^{(t)} - \eta g^{(t)}\Big)$$

satisfies

$$\|p^{(t+1)} - p^*\|^2 \;\le\; \|p^{(t)} - p^*\|^2 - 2\eta\langle g^{(t)}, p^{(t)} - p^*\rangle + \eta^2\|g^{(t)}\|^2.$$

Lemma 2 (Non-Convex Descent Lemma). If F is L-smooth, then for update $p^{(t+1)} = p^{(t)} - \eta g^{(t)}$, we have

$$F(p^{(t+1)}) \;\le\; F(p^{(t)}) - \eta\langle \nabla F(p^{(t)}), g^{(t)}\rangle + \frac{L}{2}\eta^2\|g^{(t)}\|^2.$$

A.3 Main Results

Convex Convergence. Suppose F is convex, G-Lipschitz, and $\mathcal{P}$ has diameter D. Let $\bar{p}_T = \frac{1}{T}\sum_{t=1}^{T} p^{(t)}$. Under (A1)–(A2) and step size $\eta = \frac{D}{G\sqrt{T}}$, we obtain:

$$\mathbb{E}[F(\bar{p}_T)] - F(p^*) \;=\; O\!\left(\frac{1}{\sqrt{T}}\right).$$

Proof sketch. By Lemma 1 and convexity:

$$F(p^{(t)}) - F(p^*) \le \langle g^{(t)}, p^{(t)} - p^*\rangle.$$

Summing over $t = 1, \ldots, T$ and applying (A1)–(A2), we bound the regret:

$$\sum_{t=1}^{T} \mathbb{E}[F(p^{(t)}) - F(p^*)] \;\le\; \frac{D^2}{2\eta} + \frac{\eta G^2 T}{2}.$$

Using Jensen's inequality for $\bar{p}_T$ and optimizing η, we conclude the $O(1/\sqrt{T})$ rate.

Non-Convex Convergence. Suppose F is L-smooth and (A1)–(A2) hold. With constant step size $\eta = \Theta(1/\sqrt{T})$, we have

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\Big[\|\nabla F(p^{(t)})\|^2\Big] = O\!\left(\frac{1}{\sqrt{T}}\right).$$

Proof sketch. Applying Lemma 2 and taking conditional expectation:

$$\mathbb{E}[F(p^{(t+1)})] \le \mathbb{E}[F(p^{(t)})] - \eta\mu\mathbb{E}[\|\nabla F(p^{(t)})\|^2] + \tfrac{L}{2}\eta^2(\rho\mathbb{E}[\|\nabla F(p^{(t)})\|^2] + \sigma^2).$$

Summing over $t = 1 \ldots T$ gives

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\Big[\|\nabla F(p^{(t)})\|^2\Big] \;\le\; \frac{2(F(p^{(1)}) - F_{\inf})}{\mu T\eta} + \frac{L\sigma^2}{\mu}\eta.$$

Balancing terms with $\eta = \Theta(1/\sqrt{T})$ yields the claimed rate.

A.4 Discussion: Connecting Theory and Mechanism

These results demonstrate that MAPGD achieves the same $O(1/\sqrt{T})$ convergence rate as classical stochastic gradient descent Bottou et al. (2018), despite operating in a discrete prompt space.

- **Alignment (A1)** is enforced by semantic gradient fusion, which aggregates diverse agent proposals into a direction aligned with the true descent.

- **Variance control (A2)** is realized via bandit-based sampling, which downweights unreliable or redundant updates.
- **Smoothness (A3)** is reasonable because empirical loss functions used in NLP tasks are typically Lipschitz or smooth under standard embeddings.

In summary, our analysis justifies MAPGD as a principled extension of stochastic approximation methods to structured prompt spaces, combining theoretical soundness with empirical effectiveness.

B Case Study: System Prompt Optimization

To further illustrate the applicability of MAPGD, we present a case study where our method is applied to optimize the system prompt of a large language model assistant. The original prompt is designed to support multi-source analysis and financial data interpretation for decision-making tasks. Using MAPGD, we refine the prompt to enhance robustness, accuracy, and interpretability by embedding explicit verification protocols, structured analysis guidelines, and risk prioritization frameworks. This example highlights how MAPGD can be deployed in practical LLM applications beyond benchmark datasets, particularly in domains where **data authenticity, reliability, and interpretability are critical**.

Original System Prompt.

> You are an AI assistant designed to process, analyze, and synthesize information from multiple sources in order to answer user questions, generate insights, and prepare detailed reports. You have specialized capabilities in **financial data interpretation**, **knowledge retrieval**, and **multi-source analysis**. You support both operational and strategic decision-making for Golden Section's portfolio companies.

Optimized System Prompt via MAPGD.

> You are an AI assistant specialized in processing, analyzing, and synthesizing information from multiple sources to answer user questions, generate insights, and prepare detailed reports. Your core capabilities include **financial data interpretation**, **knowledge retrieval**, and **multi-source analysis**, with a focus on supporting operational and strategic decision-making for Golden Section's portfolio companies.
>
> In performing your duties, you must ensure that all input data undergoes rigorous verification for authenticity, accuracy, and completeness before any analysis is conducted. This includes implementing protocols to validate financial figures, legal terms, and other critical information for alignment with established norms and credible sources. Your process must involve:
>
> - Cross-referencing information from diverse, credible sources to detect and mitigate false, exaggerated, or incomplete data.
> - Assessing the reliability of each source, prioritizing primary sources where available.
> - Identifying and resolving inconsistencies, ambiguities, or potential misinformation through systematic checks.
> - Ensuring all risk assessments and conclusions are based solely on validated and accurate inputs to maintain the integrity of your outputs.
>
> **Data Authenticity and Completeness Verification**
>
> - Scrutinize contextual cues (e.g., "Context: Section: Payback Period:") to ensure alignment with expected data types and structures.
> - Check for numerical or factual inconsistencies, such as typos (e.g., "11975" instead of "1975"), exaggerations (e.g., "$22.0M" without supporting context), or missing critical information.
> - Validate that all referenced data points are present, logically consistent, and contextually appropriate.

- Flag and document any anomalies for further investigation before proceeding with classification or analysis.

Context Interpretation and Parsing Guidelines

- Carefully interpret and utilize contextual cues, especially in nested or ambiguous contexts (e.g., "Context: Section: Name & Headquarters:").
- Accurately parse section headers and contextual clues to prevent misclassification or incomplete analysis in multi-section reports.
- Anchor analysis to the document's structure by adhering to hierarchical or sequential organization.
- Resolve discrepancies in contextual labeling or structure to maintain coherence.

Structured Classification and Risk Prioritization Framework

1. **Factual Reporting and Descriptive Analysis:** Present verified information such as corporate history, operational metrics, and financial data neutrally, before transitioning to evaluative content.
2. **Business Analysis:** Evaluate performance, market positioning, and strategic initiatives; assess risks by severity, likelihood, and propose contextualized mitigation.
3. **Legal Risk Analysis:** Examine compliance, regulatory, and contractual risks; assess impact and propose mitigation actions aligned with legal context.
4. **Cross-Domain Analysis:** For overlapping elements, classify by primary context and document dual-category cases with rationale.

Validation Mechanisms for Cross-References

- Distinguish between source types (e.g., governance vs. identity records).
- For each statement, explicitly identify the source type and ensure contextual alignment.