RE-GPS: Reflective Evolutionary Gradient Prompting System for Large Language Models

Nikita Kulin, Viktor Zhuravlev, Artur Khairullin, Sergey Muravyov
ITMO University
Saint-Petersburg, Russia
{242106,334857,368983}@niuitmo.ru, smuravyov@itmo.ru

***Abstract*—Despite their effectiveness in text generation, Large Language Models (LLMs) remain highly sensitive to input instructions (prompts), where suboptimal prompts frequently cause hallucinations, shallow reasoning, and logical inconsistencies. Automatic prompt optimization (autoprompting) based on evolutionary algorithms effectively frames prompt engineering as a black-box optimization problem, yet existing methods still suffer from critical limitations, including a tendency to get stuck in local optima and insufficient exploration of the prompt space.**

We propose RE-GPS (Reflective Evolutionary Gradient Prompting System), a novel autoprompting framework that integrates textual gradients into a reflective evolution pipeline. By generating precise, data-driven feedback on prompt performance over training subsets, RE-GPS substantially improves short-term reflections that guide the evolutionary search. Evaluations with GPT-4o-mini across 7 diverse datasets demonstrate that RE-GPS achieves state-of-the-art average performance relative to evolutionary baselines while showing strong robustness across tasks.

I. Introduction

Large Language Models (LLMs) have revolutionized the field of natural language processing and artificial intelligence, enabling unprecedented capabilities in tasks such as text generation, machine translation, question answering, and code synthesis [1]. Currently, LLMs play a vital role in numerous industries, from healthcare, where they assist in diagnostics and patient interactions, to finance for algorithmic trading and risk assessment, and education for personalized tutoring systems [2]. Their widespread adoption underscores their importance in improving human-machine interactions and automating complex cognitive tasks [3].

However, the performance of LLMs is highly dependent on the quality of input prompts. Subtle variations in prompt phrasing, structure, or included examples can lead to significantly different outputs, often resulting in inconsistencies or suboptimal performance [4]. This sensitivity necessitates careful prompt design to elicit desired responses effectively.

Traditionally, prompt optimization has been a manual process known as prompt engineering. This involves iterative trial-and-error experimentation by experts who have deep knowledge of LLM behaviors and domain-specific requirements. Although effective, manual prompt engineering is time-consuming, resource-intensive, and inaccessible to non-experts. It also struggles with scalability across diverse tasks and lacks standardization, hindering reproducibility [5].

To address these limitations, several automatic prompt optimization (APO) methods have been developed. These include techniques like Automatic Prompt Engineer (APE) [6], which generates and scores prompt candidates using LLMs themselves. Optimization by PROmpting (OPRO) [7], which treats prompt optimization as a search problem guided by natural language feedback [7]. Other approaches leverage evolutionary algorithms (EvoPrompt) or gradient-based optimization (Prompt Tuning) to iteratively improve prompts [8], [9].

Despite their advancements, existing APO methods have notable drawbacks. Most of them often rely on high-quality initial prompts or datasets, limiting their applicability in data-scarce scenarios. In addition, they rely predominantly on undirected search strategies, severely limiting their efficiency and search space exploration. Additionally, these methods may not generalize well across different LLMs or tasks, and can cause issues such as hallucinations if not carefully managed [10].

To address these limitations, we introduce the Reflective Evolutionary Gradient Prompting System (RE-GPS), a novel framework that unifies reflective evolution with textual gradients [11]. First, RE-GPS grounds the evolutionary search by constructing a dynamic problem description and synthesizing the vague initial prompt with empirical training examples. Second, it transforms standard crossover into a precise error-correcting operation via short-term reflections; these are generated by extracting *textual gradients* from an explicit analysis of K specific failure instances for each parent. Third, to prevent premature convergence, RE-GPS continuously aggregates these epoch-level insights into a long-term reflection memory, utilizing this accumulated historical knowledge to rigorously steer elitist mutation.

II. Related Work

This section positions our Reflective Evolutionary Gradient Prompting System (RE-GPS) within prior work on prompting for large language models (LLMs), manual prompt engineering, and automatic prompt optimization (APO). We first review manual and automated approaches, provide a concise comparative analysis, including a focused comparison of PromptBreeder and GEPA, and finally explain how RE-GPS addresses remaining gaps.

A. Manual prompt engineering

Historically, prompt engineering has been a predominantly manual, human-driven process: practitioners iteratively craft, test, and refine prompts to elicit desired behaviors [12]. Manual methods benefit from human intuition and domain expertise and can yield high-quality prompts for well-scoped problems. Manual methods range from simple input templates such as few-shot techniques [13] to advanced strategies such as Chain-of-Thought prompting [14], Self-Discover [15], Tree-of-Thoughts [16], etc. [17] However, manual engineering is time-consuming, poorly scalable across tasks and domains, prone to subjective choices, and difficult to reproduce [5]. These limitations motivate the development of automatic prompt optimization techniques that aim to reduce human effort while improving robustness and reproducibility.

B. Automatic Prompt Optimization (APO)

Automatic prompt optimization (APO) has emerged as a research area that systematically searches for or constructs prompts to maximize task performance. Below we present a practical taxonomy that groups APO methods into three broad families: (1) LLM-driven search and feedback methods, (2) evolutionary and population-based approaches, and (3) gradient-based or continuous ("soft") prompt methods. For each family, we highlight representative works, strengths, and limitations.

1) LLM-driven prompt search and natural-language feedback: A class of APO methods treats the LLM itself as an oracle or an optimizer: candidate prompts are generated and scored by LLMs, and natural-language feedback or self-evaluation guides iterative refinement. Examples include the Automatic Prompt Engineer (APE), which generates and ranks candidate prompts using LLM capabilities [6], and optimization-by-prompting techniques that view prompt search as a language-mediated optimization loop [7]. LLM-driven methods have the conceptual advantage of leveraging the model's own knowledge to propose semantically coherent prompts, but they often require many LLM calls, can amplify model biases, and may suffer from instability in the self-evaluation signal or sensitivity to sampling choices.

2) Evolutionary and population-based prompt optimization: Evolutionary algorithms and population-based searches apply principles of mutation, crossover, and selection to explore discrete prompt spaces. Work such as EvoPrompt adapts evolutionary operators to the prompt optimization problem, constructing populations of prompts and evolving them via fitness-driven selection [8]. These approaches are effective for broad, non-convex search landscapes and can escape local optima more reliably than purely local methods. Nevertheless, conventional evolutionary strategies for text encounter a number of difficulties: genetic operators are often syntactically primitive (relying on random edits and recombinations), they lack strong semantic guidance for mutation and crossover, and the approaches can be sample-inefficient, as converging to a solution necessitates evaluating a large number of offspring.

3) Gradient and soft-prompt methods: An alternative research direction formulates prompt optimization in a continuous parameter space. Prompt tuning (soft prompts) learns continuous embeddings that are added to the inputs and updates them via gradient-based training using task losses [9]. More recent methods attempt to extract a notion of *textual gradient* — i.e., using model critiques or textual feedback as an analogue of gradient information — to guide discrete text edits [11]. Gradient-style methods provide fine-grained, directed improvement and can be parameter-efficient, but they typically perform local optimization and may struggle with global exploration or with operating purely in discrete natural-language prompt spaces without a differentiable proxy.

C. PromptBreeder and GEPA

Two recent systems demonstrate complementary ideas that are especially relevant when positioning RE-GPS: *PromptBreeder* and *GEPA*.

PromptBreeder is a self-referential evolutionary framework that simultaneously evolves task-prompts and the *mutation-prompts* that generate edits [18]. The method maintains a population of task-level prompts. In each generation, an LLM proposes mutated offspring guided by mutation-prompts. Crucially, mutation-prompts themselves are subject to evolution, producing a self-improving mutation strategy that refines how offspring are proposed. PromptBreeder achieves strong empirical gains on arithmetic and commonsense benchmarks and demonstrates the power of evolving not just solutions (prompts) but the operators that mutate them.

GEPA [19] combines genetic-style evolution with reflective natural-language diagnosis and Pareto-aware selection. GEPA samples execution traces or rollouts, uses LLM-based reflection to diagnose failure modes and produce candidate fixes, and applies a Pareto-front selection procedure to maintain diversity. GEPA emphasizes sample efficiency and demonstrates that reflective, language-driven diagnostics together with Pareto selection can outperform RL-style methods on certain tasks while using far fewer rollouts.

a) Comparative observations.: While PromptBreeder emphasizes self-referential improvement of mutation operators (i.e., learning to mutate better), GEPA emphasizes reflective diagnosis and Pareto-driven selection to preserve trade-offs and diversity. Both approaches illustrate how bringing semantic, LLM-driven signals into evolutionary search improves performance compared to naive genetic operators. However, they differ in important ways that inform the RE-GPS design:

- **Operator learning vs. error-aware recombination.** PromptBreeder learns better mutation operators over time (meta-evolution), which can produce higher-quality offspring, but mutations remain driven by the evolving mutation-prompts rather than explicit, failure-specific corrective signals. RE-GPS instead constructs *textual gradients* from concrete failure instances to produce error-correcting recombination (i.e., crossover becomes a semantically guided corrective operation).

TABLE I. COMPARATIVE ANALYSIS OF REPRESENTATIVE APO APPROACHES

Method	Search paradigm	Typical weakness
APE	LLM-driven generation and ranking	High LLM cost; unstable self-evaluation signal
OPRO	Iterative natural-language feedback / LLM-as-optimizer	Many iterations; sensitive to feedback quality
EvoPrompt	Evolutionary / population-based search	Crude operators; sample-inefficient
PromptBreeder	Self-referential evolution (evolving mutation-prompts)	Operator evolution may still lack explicit error-correction signals
GEPA	Reflective genetic evolution with Pareto-aware selection	System complexity; depends on quality of reflections
Prompt Tuning	Continuous / gradient-based (soft prompts)	Prone to local optima; requires model parameter access
TextGrad	Textual-gradient / critique-guided edits	Local search; limited global exploration

- **Pareto selection and multi-objective diversity.** GEPA's Pareto-aware selection explicitly optimizes for multiple objectives and preserves diverse trade-off solutions. RE-GPS is compatible with Pareto selection, but prioritizes accumulation of epoch-level reflections into a long-term memory to guide elitist mutation and to correct persistent weaknesses across runs.
- **Memory and long-term reflection.** PromptBreeder's self-referential loop improves mutation prompts within a run, and GEPA leverages reflection per rollout and preserves diversity via Pareto selection; neither places the same emphasis on a compact, cross-epoch reflection memory that aggregates error patterns to steer future mutation systematically. RE-GPS explicitly implements such a memory to avoid repeating transient or noisy corrections and to bias exploration toward resolving recurring failures.

D. Comparative analysis

We synthesize the main trade-offs across APO families in Table I and summarize persistent limitations below.

Common, unresolved challenges include:

- **Weak error-awareness in operators.** Evolutionary crossover and mutation are often syntactic or random and do not exploit explicit analyses of failure cases to guide recombination.
- **Lack of long-term memory and reflective learning.** Most APO methods optimize within an episode or single run and do not accumulate persistent knowledge about recurring failure modes or effective corrective edits.
- **Generalization and hallucination risks.** If optimization procedures rely on flawed evaluation signals, they can reinforce hallucination-prone behaviors or fail to generalize across tasks and models [10].

E. Positioning RE-GPS relative to prior work

RE-GPS synthesizes ideas from the evolutionary and textual-gradient families while addressing the limitations above. Concretely, RE-GPS:

1) **Grounds evolutionary search with task-specific problem descriptions.** Instead of relying on purely syntactic population operators, each candidate prompt is grounded by a problem description, that is constructed by synthesizing the initial prompt with representative examples sampled from the training dataset. This grounding reduces drift and anchors search to concrete task probes.
2) **Transforms crossover into error-correcting operations.** Traditional crossover is replaced by a reflection-driven recombination: for each parent, RE-GPS extracts *textual gradients* in the form of short-term reflections derived from K specific failure instances. These reflections serve as semantically meaningful corrections that guide precise recombination and reduce the production of syntactically plausible but semantically useless offspring.
3) **Maintains long-term reflection memory for guided elitist mutation.** To avoid premature convergence, RE-GPS accumulates epoch-level reflections into a memory buffer. This historical knowledge biases subsequent elitist mutation and selection toward correcting persistent weaknesses rather than repeating transient changes.

Relative to PromptBreeder, RE-GPS retains the benefits of evolutionary population search and operator refinement while shifting the operator focus from self-referential mutation-prompt evolution to explicit failure-driven textual gradients that act as corrective recombination signals. Compared to GEPA, RE-GPS shares the emphasis on reflection and sample efficiency, but it additionally prioritizes a compact, cross-epoch reflection memory and formalizes textual gradients as a first-class artifact used at crossover time (rather than only during diagnosis-and-proposal phases).

F. Summary

Overall, prior APO methods have advanced the state of automatic prompt construction but leave important challenges open in operator semantics, memory, and sample efficiency. RE-GPS addresses these gaps by unifying reflective textual gradients with population-based search and a compact reflection memory.

III. RE-GPS

Framework overview. The Reflective Evolutionary Gradient Prompting System (RE-GPS) frames prompt engineering as a guided black-box optimization over a discrete prompt space. The pipeline is illustrated in Fig. 1.

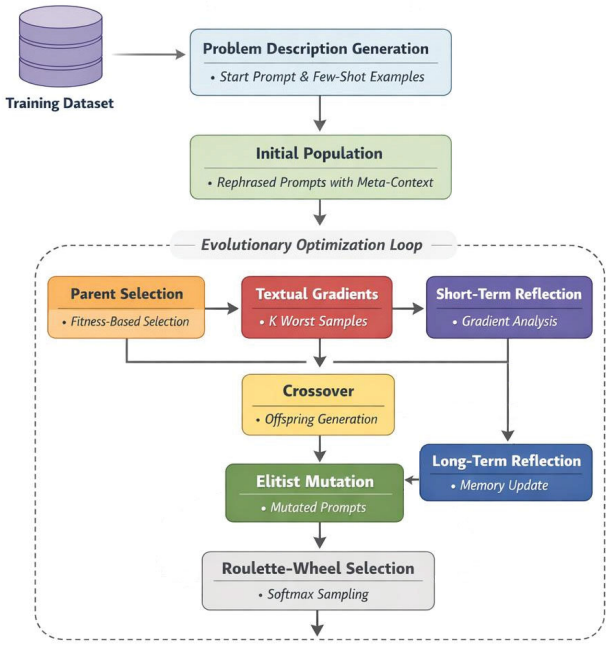

Fig. 1. The overall RE-GPS pipeline. The process initializes with an data-driven problem description. Each evolutionary epoch incorporates textual gradients to form short-term reflections for crossover, while continually updating a long-term reflection memory that guides elitist mutation.

A. Method initialization

Because user instructions are frequently vague or ambiguous, RE-GPS begins by synthesizing a comprehensive *problem description*: the original user instruction is enriched with representative examples sampled from the training set. This enriched description is injected into the meta-prompt so that subsequent LLM-based operations (generation, critique, and recombination) are grounded in task-specific dependencies and evaluation criteria. Using the enriched description, RE-GPS initializes a starting population by producing N_{pop} diverse alternative versions of the original prompt via controlled LLM generation and data-driven paraphrasing.

B. Algorithmic loop and components

RE-GPS proceeds in epochs; each epoch refines the population through a sequence of logically-separated stages that combine classical evolutionary machinery with LLM-derived semantic signals.

a) Parent selection: Parents are sampled from the current population with probabilities proportional linearly to their empirical fitness scores. This selection mechanism resembles canonical proportional selection in evolutionary computation [20], but we note two practical modifications: (i) selection is carried out on fitness scores computed against the enriched problem description (not raw prompt outputs alone), and (ii) RE-GPS optionally applies temperature-controlled smoothing to avoid the over-exploitation of transiently high-scoring individuals.

b) Textual gradients (failure-driven critique): For each parent prompt, RE-GPS identifies a small set of K failures or low-performance examples sampled from the training pool. The LLM is then prompted to act as a critic: for each failure instance, it produces a *textual gradient* — a structured, natural-language critique that (a) explains why the prompt produced an undesirable output on that example, (b) pinpoints missing or ambiguous instruction elements, and (c) suggests corrective edits or additional constraints that are explicitly tailored to the observed data distribution. This mechanism builds on the idea of using text-based model critiques as gradient analogues [11] but applies the critique specifically to localized failure cases extracted relative to each parent.

c) Short-term reflections and gradient-guided crossover: For each parent pair, RE-GPS synthesizes the parents together with their respective textual gradients to produce a short-term reflection: a concise, structured summary that captures complementary strengths and locally-relevant corrective edits. The crossover operator is implemented as a *reflection-guided recombination* that uses the short-term reflection as an instruction to the LLM for composing offspring prompts. Unlike naive string-level splice-and-mutate operators, this crossover is semantically informed: the LLM is asked to produce offspring that preserve high-value traits while applying corrective edits recommended by textual gradients. This produces semantically richer children and reduces the rate of syntactically plausible but semantically useless offspring described in prior evolutionary text work [8].

d) Long-term reflection memory: After completing crossover for the entire population, RE-GPS aggregates epoch-level short-term reflections into a long-term reflection text (a compact, human- and machine-readable memory). The update rule for this memory is incremental: the previous long-term reflection is fused with distilled signals from the newest short-term reflections, e.g., by asking the LLM to summarize recurrent failure patterns and to update canonical mitigation strategies. This memory provides cross-epoch continuity, enabling the optimizer to prioritize corrections for persistent failure modes rather than repeatedly exploring ephemeral or noisy edits. The notion of maintaining cross-run knowledge distinguishes RE-GPS from systems that only operate within single-run episodes (see Sec. II).

e) Elitist mutation: Elitist mutation uses the current elite (the best-performing prompt) together with the long-term reflection memory to generate additional N_{pop} mutated variants. Because mutation is conditioned on the distilled, long-term reflection, it is biased toward corrective, high-utility edits rather than blind perturbations. This form of directed elitist mutation preserves strong individuals while systematically probing semantically-plausible neighborhoods that are more likely to resolve recurring errors. The use of elitism and directed mutation follows standard principles in evolutionary computation [21] but adapts them for semantic text spaces by using natural-language reflections as mutation guides.

f) Final population selection: For the final selection, each candidate (parents, offspring, and mutants) is evaluated

Initial Prompt:
Given a context answer on the question.

Final Prompt:
Please read the context provided carefully and restate the problem in your own words to ensure you fully understand it. Identify and summarize all key numerical values and their relationships before proceeding. Break down the problem into clear, manageable steps and outline each step before performing any calculations. As you work through the calculations, double-check for consistency and ensure that your final numeric answer logically corresponds to the original question. After arriving at your answer, take a moment to re-evaluate your calculations and reasoning to confirm that you haven't overlooked any details. Consider the practical implications of your answer to ensure it makes sense in a real-world context. Present your final numeric answer clearly at the end of your response.

Fig. 2. Example of RE-GPS prompt optimization on GSM8k dataset

against the objective metric. Scores are transformed with a softmax-like mapping to survival probabilities (a standard technique to stabilize selection pressure; see e.g., softmax action selection in sequential decision problems [22]). RE-GPS then applies a roulette-wheel selection to sample the next generation of N_{pop} prompts. This probabilistic selection balances the exploitation of high-performing prompts with the retention of structural diversity that supports continued exploration.

An example of the RE-GPS optimization is shown in Figure 2

C. Design rationale and connections to prior work

RE-GPS intentionally combines population-based global exploration (inherited from classical evolutionary algorithms [20], [21]) with localized, gradient-like, data-driven corrections (inspired by textual-critique methods [11]) and with reflection-driven memory (in the spirit of reflective optimization approaches such as GEPA and PromptBreeder). The main motivations for each design choice are:

- **Example-driven grounding.** Enriching prompts with representative examples reduces ambiguity and enables more reliable LLM critique and recombination: generation conditioned on concrete cases produces more targeted edits than unconstrained paraphrasing [6].
- **Failure-focused textual gradients.** Extracting critiques from concrete failure instances yields high-signal guidance for edits; this contrasts with generic operator evolution (PromptBreeder) which learns how to mutate but does not necessarily produce error-specific corrections.
- **Memory-augmented elitism.** Persisting distilled reflections across epochs mitigates repeated oscillation between transient fixes and helps the search converge toward stable, correct-instruction formulations.
- **Stochastic selection with tempered pressure.** Softmax transformation and roulette-wheel selection provide a controlled selection pressure that preserves useful diversity while still favoring high-fitness prompts; such tempered stochasticity is important when the evaluation signal has variance (e.g., due to sampling-based LLM outputs).

D. Limitations and failure modes

We emphasize several limitations and potential failure modes to guide future work and the careful application of RE-GPS:

- **Quality of textual gradients.** The effectiveness of RE-GPS depends on the fidelity of LLM critiques. If the model produces misleading or biased critiques, these can drive search toward suboptimal or hallucinatory corrections. To address these issues, several strategies can be employed: filtering critiques through ensemble methods or confidence thresholds, and incorporating human-in-the-loop verification for high-stakes deployments.
- **Dependence on example sampling.** The initial problem description and failure-instance selection depend on the sampling strategy; poor sampling can bias search. We recommend stratified or adversarial sampling schemes to surface representative failure modes.

IV. Experimental Evaluation

This section describes the experimental protocol used to evaluate RE-GPS, the baselines and metrics considered, and implementation details necessary to reproduce the results.

A. Goals

Our empirical study is designed to answer the following questions:

1) **Effectiveness**: Does RE-GPS produce prompts that yield better downstream performance than state-of-the-art automated prompt optimizers across a wide range of tasks?
2) **Generality**: How well does RE-GPS generalize across heterogeneous task families (reasoning, summarization, QA, classification, code generation)?

B. Benchmarks and evaluation metrics

We evaluate on seven established datasets chosen to cover diverse NLP capabilities and evaluation metrics:

- **CommonGen** (commonsense generation) — evaluated with BERTScore.
- **XSUM** (news summarization) — BERTScore.
- **MediQA** (medical question answering) — BERTScore.
- **GSM8K** (mathematical reasoning) — Exact Match.

- **SQuAD v2** (reading comprehension / QA) — BERTScore.
- **TweetEval** (sentiment classification) — Macro F1-Score.
- **CONCODE** (code generation) — CodeBERTScore.

Where applicable, we follow standard dataset splits for train, validation and test – 100/50/300 to ensure comparability with baselines.

C. Baselines

We compare RE-GPS against representative automated prompt optimization approaches and a reference prompt baseline:

- **EvoPrompt** — evolutionary / population-based prompt optimizer.
- **PromptBreeder** — self-referential evolution that evolves mutation-prompts.
- **GEPA** — reflective genetic evolution with Pareto-aware selection.
- **ReflectivePrompt** — (denoted further as Ref. Prompt) autoprompting method based on reflective evolution. [23]

These baselines represent the major APO families discussed in Section II.

D. Model and evaluation protocol

All prompt generation and evaluation procedures used `gpt-4o-mini` to provide a consistent computational baseline and to reflect practical closed-API settings.

E. Implementation and hyperparameters

Key implementation choices used in our reported experiments are:

- **Population size** N_{pop}: 10 as a balance between exploration and cost.
- **Temperature**: 0.7.
- **Number of each method run**: 3.

Rest hyperparameters of compared baselines were set as in their prior work.

V. Results

Table II summarizes the primary results: per-dataset performance for each method and the arithmetic average across datasets. The table reports the principal metric for each dataset (BERTScore, Exact Match, Macro F1, or CodeBERTScore as appropriate).

TABLE II. Performance comparison of automated prompt generation methods using `gpt-4o-mini`. Best results are **bolded**.

Dataset	EvoPrompt	PromptBreeder	Ref.Prompt	GEPA	**RE-GPS (Ours)**
CommonGen	0.808	0.803	0.808	0.823	**0.824**
XSUM	0.714	0.711	0.721	0.716	**0.725**
MediQA	**0.736**	0.734	0.720	0.720	0.720
GSM8K	0.905	0.892	0.861	0.880	**0.912**
SQuAD v2	0.827	0.859	0.860	**0.922**	0.912
TweetEval	0.557	**0.576**	0.575	0.475	0.561
CONCODE	0.664	0.655	0.655	0.670	**0.671**
Average	0.744	0.747	0.744	0.743	**0.761**

A. Key observations

- **Overall performance.** RE-GPS achieves the highest **average** performance across the seven tasks (0.761), indicating robust cross-task gains when compared to the baselines.
- **Strong performance on reasoning and generative tasks.** RE-GPS attains the best results on **GSM8K** (mathematical reasoning), **CommonGen** (commonsense generation), **XSUM** (summarization), and **CONCODE** (code generation). These tasks share the need for structured, semantically-rich prompts, which aligns with RE-GPS's textual-gradient and reflection-driven recombination mechanisms.
- **Where RE-GPS is not the top performer.** On **MediQA** and **SQuAD v2**, EvoPrompt and GEPA respectively achieve the best scores. On **TweetEval** PromptBreeder shows a small advantage. We analyze these differences in Section VI.

VI. Discussion

In this section we interpret results, analyze likely causes for per-dataset differences, discuss ablations and robustness, and point out limitations and practical considerations.

A. Why RE-GPS excels on certain tasks

The combination of failure-focused textual gradients and long-term reflection memory appears particularly effective when:

- The task requires precise instruction of multi-step reasoning or output formatting (e.g., GSM8K, CONCODE). Textual gradients expose concrete reasoning failures and recommend targeted fixes that the crossover operator can inject into offspring.
- The optimal prompt benefits from explicit exemplars or constrained output structure (e.g., CommonGen, XSUM). RE-GPS's example-driven problem descriptions help the LLM produce and critique prompts that preserve these structural cues.

B. Understanding cases where RE-GPS lags

For datasets where RE-GPS is not the best, several plausible explanations exist:

- **Domain specificity and knowledge requirements (MediQA).** Medical QA often requires domain-specific phrasing and specialized evaluation signals; simple prompt-space edits may not fully compensate for domain-specific training data or model internals. Methods with different mutation dynamics (e.g., EvoPrompt) may have fortuitously found phrasing that matches medical tokenization or domain-specific characteristics.
- **Pareto-front and multi-objective trade-offs (SQuAD v2).** GEPA's explicit Pareto-aware selection preserves a diverse set of trade-off solutions, which can be advantageous for tasks with different objective facets (e.g., precision vs recall in SQuAD v2). RE-GPS focuses on consolidating persistent failure corrections, which may

slightly bias toward robust average behavior rather than extreme Pareto-optimal candidates.

- **Low-resource discriminative tasks (TweetEval).** Sentiment classification on short, noisy texts often benefits from prompt variants that encode lexical signals and domain-specific heuristics; PromptBreeder's operator-evolution mechanism may have discovered mutation strategies particularly suited to short informal text.

C. Limitations and future work

Key limitations to be addressed in future research:

- **Quality of critiques.** If the LLM produces low-fidelity or biased textual gradients, RE-GPS can be driven toward suboptimal corrections; ensemble critiques, calibrations, or lightweight human verification can mitigate this risk.
- **Sensitivity to sampling strategy.** The choice of failure-instance sampling affects which error modes get corrected; more adversarial or stratified sampling strategies may improve coverage of rare but important failures.
- **Cost-aware extensions.** Integrating cost-aware objectives explicitly into selection (e.g., API cost vs performance trade-offs) is an important practical extension. For example, GEPA-style Pareto optimization is a promising direction.

D. Conclusions from experiments

Overall, empirical evidence indicates RE-GPS offers a robust, general-purpose approach to automated prompt optimization that consistently improves average performance across heterogeneous tasks. Its design trades some additional per-epoch LLM expense for improved sample efficiency and semantic quality of offspring prompts; this trade-off is often acceptable in practical scenarios where prompt quality materially impacts downstream task performance.

VII. Conclusion

In this paper we introduced RE-GPS (Reflective Evolutionary Gradient Prompting System), a novel automated prompt-optimization framework that unites population-based evolutionary search with failure-driven *textual gradients* and a compact long-term reflection memory. RE-GPS reframes crossover as an error-correcting, reflection-guided recombination and augments elitist mutation with distilled cross-epoch knowledge. These design choices enable the optimizer to convert concrete failure analyses into semantically meaningful prompt edits rather than relying on syntactic or blind perturbations.

In summary, RE-GPS demonstrates that grounding evolutionary operations in concrete failure analysis and retaining distilled long-term reflections produces semantically richer and more effective prompt variants. We believe this hybrid, reflection-first strategy is a practical and extensible step toward more reliable automated prompt engineering, and we hope it catalyzes additional research into memory-augmented, critique-guided optimization for LLMs.

References

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, 2020.

[2] Clarifai, "Applications of large language models across industries," https://www.clarifai.com/blog/large-language-model-applications, 2025.

[3] HatchWorks AI, "How large language models are transforming business," https://hatchworks.com/blog/gen-ai/large-language-models-business-impact, 2026.

[4] R. Wolfe, A. Patel, and M. Kim, "Prompt sensitivity in large language models," *arXiv preprint arXiv:2305.00000*, 2023.

[5] S. Murthy and K. Rao, "Challenges in prompt engineering and reproducibility," https://example.com/prompt-engineering-challenges, 2025.

[6] Y. Zhou, Y. Chen, R. Li *et al.*, "Large language models are human-level prompt engineers," *arXiv preprint arXiv:2211.01910*, 2022.

[7] C. Yang, Y. Zhou *et al.*, "Large language models as optimizers," *arXiv preprint arXiv:2309.03409*, 2023.

[8] Q. Guo, J. Zhao *et al.*, "Evoprompt: Connecting llms with evolutionary algorithms yields powerful prompt optimizers," *arXiv preprint arXiv:2309.08532*, 2023.

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

[10] L. Chen and Z. Huang, "Generalization and hallucination issues in prompt optimization," https://example.com/prompt-optimization-hallucinations, 2025.

[11] M. Yuksekgonul *et al.*, "Textgrad: Automatic "differentiation" via text," *arXiv preprint arXiv:2406.07496*, 2024.

[12] P. Liu, W. Yuan, J. Fu, H. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, 2023.

[13] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: a survey on few-shot learning," *ACM computing surveys (csur)*, vol. 53, no. 3, pp. 1–34, 2020.

[14] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, and et al., "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[15] P. Zhou, J. Pujara, X. Ren, X. Chen, H.-T. Cheng, Q. V. Le, E. Chi, D. Zhou, S. Mishra, and H. S. Zheng, "Self-discover: large language models self-compose reasoning structures," *Advances in Neural Information Processing Systems*, vol. 37, pp. 126 032–126 058, 2024.

[16] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan, "Tree of thoughts: deliberate problem solving with large language models," *Advances in neural information processing systems*, vol. 36, pp. 11 809–11 822, 2023.

[17] S. Vatsal and H. Dubey, *A survey of prompt engineering methods in large language models for different NLP tasks*, *https://arXiv.org/abs/2407.12994*.

[18] C. Fernando, D. Banarse, H. Michalewski, S. Osindero, and T. Rocktäschel, "Promptbreeder: Self-referential self-improvement via prompt evolution," *arXiv preprint arXiv:2309.16797*, 2023.

[19] L. A. Agrawal, S. Tan, D. Soylu, N. Ziems, R. Khare, K. Opsahl-Ong, A. Singhvi, H. Shandilya, M. J. Ryan, M. Jiang, C. Potts, K. Sen, A. G. Dimakis, I. Stoica, D. Klein, M. Zaharia, and O. Khattab, "Gepa: Reflective prompt evolution can outperform reinforcement learning," *arXiv preprint arXiv:2507.19457*, 2025.

[20] J. H. Holland, *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[21] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[22] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.

[23] N. Kulin, V. Zhuravlev, A. Khairullin, A. Sitkina, and S. Muravyov, "Coolprompt: Automatic prompt optimization framework for large language models," in *2025 38th Conference of Open Innovations Association (FRUCT)*. IEEE, 2025, pp. 158–166.