SELF-EVOLVING PROMPT OPTIMIZATION FOR OPEN-ENDED TEXT WITH PAIRWISE ATOMIC FEEDBACK AND COMPARATIVE PROMPT

Anonymous authors
Paper under double-blind review

ABSTRACT

Prompt optimization can improve large language models (LLMs) on a target task without changing their parameters. While most existing methods focus on tasks with well-defined or verifiable evaluation criteria, open-ended text generation poses a harder setting. Responses are often long, lack reliable references, and vary in quality across multiple dimensions, making fine-grained differences difficult for scoring-based LLM judges to detect. Recent methods use pairwise comparisons, but often reduce them to an overall preference or generate feedback only for the losing response. We propose AtomicPO, a self-evolving prompt optimization method that uses pairwise comparisons for both evaluation and feedback generation, without requiring reference texts or sample-specific evaluation criteria. AtomicPO decomposes feedback into atomic strengths and weaknesses across quality dimensions for both responses. During refinement, it also provides a comparative prompt to help preserve useful instructions and guide targeted refinements. Across four open-ended text generation benchmarks, AtomicPO consistently outperforms strong baselines using scoring-based or pairwise LLM judges. These results show the value of atomic feedback from both responses and comparative prompts for optimizing open-ended text generation.

1 INTRODUCTION

Prompt optimization methods Yuksekgonul et al. (2024); Guo et al. (2024) improve the performance of large language models (LLMs) on specific tasks by refining their prompts without modifying model parameters. Compared with fine-tuning Ouyang et al. (2022), prompt optimization requires fewer training examples and less computation, making it attractive when resources are limited.

Previous prompt optimization methods (Agrawal et al., 2026; Yuksekgonul et al., 2024) have primarily focused on tasks with verifiable or well-defined evaluation criteria, such as coding and mathematics Balunovic et al. (2026), text classification Suzgun et al. (2023), and short-form text generation Bajaj et al. (2016). Responses to these tasks can often be evaluated using programmatic verifiers Guo et al. (2025), similarity to reference texts Lin (2004), or scoring-based LLM judges Liu et al. (2023). Open-ended text generation tasks, such as long-form essay writing and multi-turn conversations, have received less attention despite their importance in general-purpose LLM applications. CriSPO (He et al., 2025) addresses problem by comparing generated responses with reference texts across automatically identified aspects. However, its reliance on reference texts limits its applicability to open-ended tasks for which suitable references are unavailable.

These tasks present several challenges for prompt optimization. Responses are often long and complex, making them difficult to verify, and high-quality reference texts may be unavailable. Their quality is also multi-dimensional: a response may be strong in factuality but weak in relevance or style (Hosking et al., 2024). Therefore, a response can simultaneously be a winner and loser on different dimensions in a pairwise comparison. Moreover, scoring-based LLM judges may struggle to distinguish fine-grained differences in quality between responses Zheng et al. (2023).

Pairwise comparisons can help evaluate such differences, but existing prompt optimization methods make limited use of the information they provide. Some methods Guo et al. (2024); Wu et al. (2026a) use pairwise comparisons to select the better prompt based on overall response quality,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

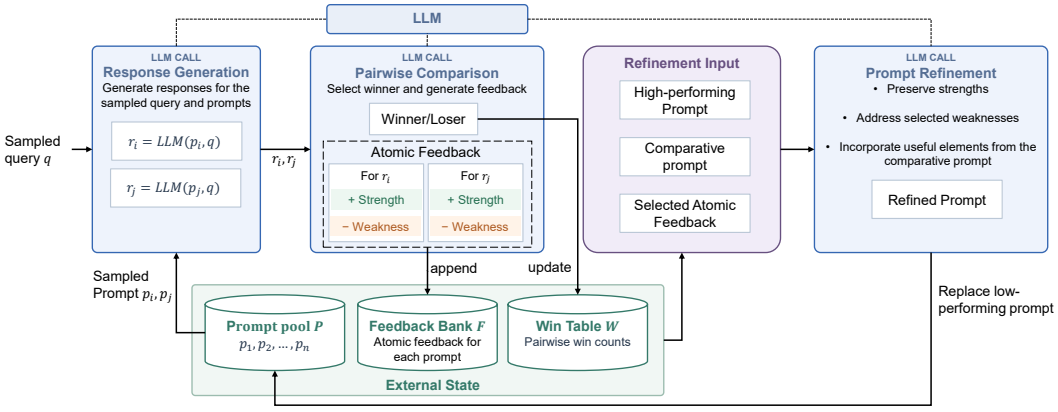

Figure 1: Workflow of AtomicPO. We introduce the notation of AtomicPO in Sec.3.1, pairwise evaluation and feedback generation in Sec.3.2, and prompt refinement in Sec. 3.3.

without capturing strengths and weaknesses across individual dimensions. Singhal et al. (2026) uses pairwise comparisons to generate feedback, but relies on scoring-based LLM judges for evaluation and generates feedback only about the losing response. Consequently, existing methods do not fully use the multi-dimensional strengths and weaknesses revealed by pairwise comparisons.

To address these challenges, we propose AtomicPO, a self-evolving prompt optimization method that uses pairwise comparisons for both evaluation and feedback generation, without requiring reference texts or sample-specific evaluation criteria. AtomicPO expresses feedback as atomic points, each describing a distinct strength or weakness of a response. It generates this feedback for both responses in a pair since the overall winner may be weaker on a particular dimension. During refinement, AtomicPO also presents the LLM with a comparative prompt. This comparison provides concrete guidance on which parts of the prompt can be fixed, reducing unnecessary wholesale rewrites.

We evaluate AtomicPO on four commonly used benchmarks for open-ended text generation Wu et al. (2026b); Yifei et al. (2026) in different domains. We compare AtomicPO with strong prompt optimization baselines based on both scoring-based and pairwise LLM judges. Across the evaluated benchmarks and LLMs, AtomicPO consistently outperforms these baselines, demonstrating the effectiveness of atomic feedback and comparative prompts for optimizing prompts for open-ended text generation.

Our contributions are four-fold:

- We propose AtomicPO, a self-evolving method that uses pairwise comparisons for both prompt evaluation and feedback generation on open-ended text generation tasks;
- We introduce atomic feedback that captures distinct strengths and weaknesses of both responses in a pair across multiple dimensions;
- We introduce comparative prompts during prompt refinement to prevent complete rewrite;
- We demonstrate consistent improvements over strong prompt optimization baselines across four open-ended text generation benchmarks.

2 Related Work

2.1 Prompt Optimization

Prompt optimization improves LLM performance on a target task by refining the instructions provided to the model. Early methods explored prompts through iterative search. OPRO Yang et al. (2024) generates new candidates based on the observed performance of previous prompts, while

PromptBreeder Fernando et al. (2023) applies evolutionary mutations to populations of prompts and meta-prompts. Other methods use feedback to guide refinement. TextGrad Yuksekgonul et al. (2024) propagates textual feedback from task losses to prompt parameters, MIPRO Opsahl-Ong et al. (2024) jointly optimizes instructions and few-shot examples through Bayesian optimization, and GEPA Agrawal et al. (2026) uses natural-language reflection and Pareto-based selection to improve candidate prompts. CriSPO compares generated texts with references across multiple aspects and turns its critiques into suggestions for prompt refinement He et al. (2025). However, its reliance on reference texts limits its applicability to open-ended tasks. Another line of work uses pairwise comparisons to obtain evaluation signals. EvoPrompt Guo et al. (2024) and related approaches Wu et al. (2026a) use pairwise comparisons to select or improve prompts, while PrefPO Singhal et al. (2026) uses an LLM judge to compare outputs from candidate prompts and provides feedback for the less-preferred prompt. These methods demonstrate the value of comparative evaluation, but typically only use comparisons to establish an overall preference or provide feedback for the losing prompt. AtomicPO instead extracts atomic strengths and weaknesses from both responses across quality dimensions, then uses this feedback to guide prompt refinement for open-ended text generation.

2.2 Evaluating Open-ended Texts with LLM judges

LLM judges are widely used to evaluate open-ended text (Wang et al., 2023; Fu et al., 2024). They commonly assign a score to an individual response (Gao et al., 2023) or compare two responses to the same query (Zheng et al., 2023). To improve scoring-based evaluation, prior work has modeled uncertainty in predicted scores (Liu et al., 2023), prompted judges to analyze the response and evaluation instructions before scoring (Chiang & Lee, 2023; Kim et al., 2024b), and introduced interactions among multiple judges(Kim et al., 2024a; Kumar et al., 2025). However, scoring-based judges may struggle to capture fine-grained differences in the quality of open-ended texts (Wang et al., 2026). Instance-specific rubrics offer a more detailed basis for evaluation (Kim et al., 2025; Arora et al., 2025), but creating or validating high-quality rubrics at scale can require substantial expert effort. In contrast, pairwise judges directly compare two responses to the same query, enabling them to more reliably identify fine-grained differences between responses (Chiang et al., 2024; Dubois et al., 2023). Since AtomicPO does not assume access to sample-specific rubrics, we adopt pairwise comparison for both evaluation and feedback generation.

3 Method

In this section, we present AtomicPO, our method for optimizing system prompts for open-ended text generation. Section 3.1 introduces the notation, Section 3.2 describes pairwise evaluation and atomic feedback generation, and Section 3.3 explains how AtomicPO updates and refines the prompt pool. Figure 1 provides an overview.

3.1 Notation

AtomicPO optimizes a system prompt for a given task and a set of user-specified evaluation criteria C. It starts with a pool of n prompts, $P = \{p_1, \ldots, p_n\}$, and a training set of queries, Q. Given a prompt p and a query $q \in Q$, the LLM generates a response $r = \mathrm{LLM}(p, x)$, where p is the system prompt and q is the query. Following the self-evolving setting, AtomicPO uses the same LLM to generate responses, evaluate them, produce feedback, and refine prompts. AtomicPO maintains a win matrix W, where $W_{i,j}$ records the number of times a response generated with p_i has been preferred over one generated with p_j. It also maintains a feedback bank F, where $F_{i,j}$ stores atomic feedback on responses generated with p_i when compared with responses generated with p_j.

3.2 Pairwise Evaluation and Feedback Generation

In this section, we describe how AtomicPO performs pairwise evaluation and feedback generation.

Before each prompt refinement stage, AtomicPO conducts m rounds of pairwise evaluation. In each round, it samples k prompt pairs (p_i, p_j) from P using Double Thompson Sampling Agrawal & Goyal (2012); Wu et al. (2026a), based on the current win matrix W. This strategy first samples a

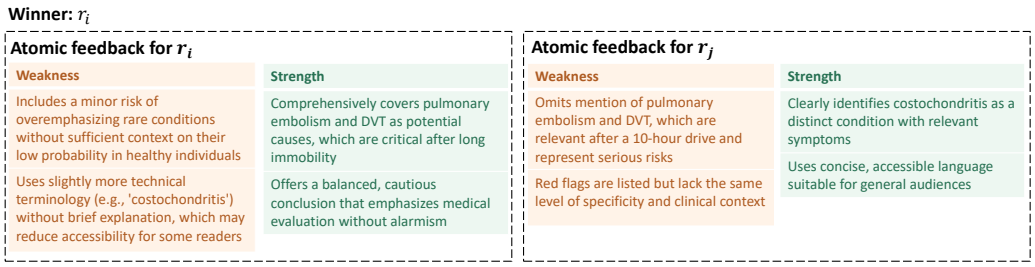

Figure 2: Example of atomic feedback generated by AtomicPO.

prompt likely to perform well, then samples a second prompt whose performance relative to the first is uncertain. We refer readers to the cited work for details.

We next describe how AtomicPO performs pairwise comparison and generates feedback for a sampled prompt pair. For each sampled pair (p_i, p_j), AtomicPO samples a query q from the training set Q. It then generates two responses, r_i and r_j, by using the corresponding prompts to the same query: $r_* = LLM(p_*, q), * \in \{i, j\}$. The query q and the two responses r_i and r_j are then provided to the LLM for evaluation and feedback generation. As in Fig. 1, the LLM is instructed to first analyze the two responses and select an overall winner $w \in \{i, j\}$ based on the criteria c. It also generates separate atomic feedback f_i and f_j for both responses, regardless of which response is selected as the winner:

$$w, f_i, f_j = LLM(c, q, r_i, r_j) \tag{1}$$

The atomic feedback f_i is designed to better capture the multi-dimensional nature of open-ended text quality. In general, a response r_i is unlikely to outperform another response r_j across all quality dimensions and may simultaneously exhibit both strengths and weaknesses relative to r_j. Specifically, f_i consists of the strengths s_i and weaknesses w_i of response r_i compared with response r_j. Each of s_i and w_i contains multiple atomic points, where each point is self-contained and describes a single strength or weakness. In rare cases, r_i may completely outperform or underperform r_j across all dimensions. In such cases, the w_i and s_j or s_i and w_j, respectively, may be empty. We show an example atomic feedback in Fig. 2 and the instruction for pairwise comparison in Tab. 11.

Once AtomicPO obtains the winner response w and the atomic feedback f_i and f_j from all k pairs in a round, it updates the win matrix and feedback bank accordingly. Specifically, if r_i is the winner $(w = i)$, the corresponding entry $W_{i,j}$ in the win matrix is increased by one. Otherwise, $W_{j,i}$ is increased by one. Meanwhile, if the feedback is non-empty, the feedback f_i is added to $F_{i,j}$, and f_j is added to $F_{j,i}$ in the feedback bank F.

3.3 Pool Update and Prompt Refinement

In this section, we describe how AtomicPO updates the prompt pool and performs prompt refinement using atomic feedback and comparative prompts.

AtomicPO first discards the o low-performing prompts from the prompt pool based on their Copeland scores (Henriet, 1985). It also removes their corresponding entries from the win matrix w and feedback bank F obtained from comparisons involving these discarded prompts. AtomicPO then selects the l high-performing prompts in the remaining pool P, again based on their Copeland scores, for refinement. AtomicPO generates o refined prompts to replace the discarded ones, allocating refinements as evenly as possible across the l selected prompts.

An effective refinement of a high-performing prompt should address its weaknesses while preserving its strengths. Motivated by this principle, AtomicPO performs focused refinements rather than completely rewriting the original prompts. Specifically, for a given prompt p_i, AtomicPO uses all strength points from all comparisons with other prompts in the prompt pool, $\{s \mid s \in \cup_{j=1}^{n} F_{i,j}\}$, to indicate which aspects of p_i already perform well and should be preserved. To identify how p_i should be modified, AtomicPO selects a comparative prompt p_{i^c} and uses the weakness points of p_i revealed

Algorithm 1 AtomicPO

Require: Prompt pool P, queries Q, criteria c, refinement interval m, pairs per round k, number of discarded prompts o, number of selected prompts l, iterations x

Initialize win matrix W and feedback bank F
for $t = 1, \ldots, x$ **do** ▷ Pairwise evaluation
  for $h = 1, \ldots, m$ **do**
    for $z = 1, \ldots, k$ **do**
      $(p_i, p_j) \leftarrow$ DoubleThompsonSampling(P, W); $q \sim Q$
      $r_i \leftarrow LLM(p_i, q)$, $r_j \leftarrow LLM(p_j, q)$
      $(w, f_i, f_j) \leftarrow LLM(c, q, r_i, r_j)$
      Update W and F using (w, f_i, f_j)
    end for
  end for
  if $t \neq x$ **then**
    Remove the o low-performing prompts from P by Copeland score ▷ Pool update and refinement
    Select the l high-performing prompts $R \subseteq P$
    for $p_i \in R$ **do**
      for $u = 1, \ldots, o/l$ **do**
        $i_c = \arg\max_{j \in \mathcal{C}_i} |\{w \mid w \in Fi, j\}|$ ▷ Select p_{i^c} with the most weakness feedback
        $S_i \leftarrow \{s \mid s \in \cup_{j=1}^{n} F_{i,j}\}$, $W_i \leftarrow \{w \mid w \in F_{i,i^c}\}$
        $p^r_{i,i^c} \leftarrow LLM(p_i, S_i, W_i, p_{i^c})$
      end for
    end for
    Add all refined prompts to P and initialize their W, F entries
  end if
end for
$p^* \leftarrow \arg\max_{p_i \in P} \text{Copeland}(p_i)$
return p^*

by its comparison with p_{i^c}, $\{w \mid w \in F_{i,i^c}\}$. AtomicPO does not use weakness points from all comparisons because aggregating them may introduce noisy or conflicting feedback, making the refinement less focused. Instead, it selects the comparative prompt that reveals the largest number of weakness points for p_i among the eligible comparative prompts: $i_c = \arg\max_{j \in \mathcal{C}_i} |\{w \mid w \in Fi, j\}|$, where $\mathcal{C}_i$ denotes the set of eligible comparative prompts. To promote diversity among refinements, AtomicPO excludes prompts from $\mathcal{C}_i$ if they have been previously used as the comparative prompts for refining p_i or if they have previously been refined using p_i as their comparative prompt. The LLM is instructed to preserve the identified strength points, address the selected weakness points, and incorporate useful elements from the comparative prompt while making only focused modifications to the original prompt. Finally, AtomicPO generates the refinement p^r_{i,i^c} of the prompt p_i using the selected comparative prompt, all strength points of p_i, and weakness points identified in its comparison with p_{i^c}:

$$p^r_{i,i_c} = LLM(p_i, \{s \mid s \in \cup_{j=1}^{n} F_{i,j}\}, \{w \mid w \in F_{i,i^c}\}, p_{i^c}) \tag{2}$$

The instruction for prompt refinement is in Tab. 13. All o refined prompts are then added to the prompt pool P, with their corresponding entries in the win matrix W and feedback bank F initialized as empty. This refinement process is repeated x times throughout the optimization process. Eventually, the prompt with the highest Copeland score calculated on the win matrix W is returned as the optimized prompt of AtomicPO. The pseudo code of AtomicPO is shown in Alg. 1

4 Experiment

4.1 Dataset

We conduct experiments on four open-ended text benchmarks spanning diverse domains: (i) the Finance & Business subset of WritingBench (Wu et al., 2026b), which evaluates essay writing under complex requirements; (ii) the Materials Science subset of ResearchQA (Yifei et al., 2026), which evaluates question answering on scientific topics; (iii) HealthBench (Arora et al., 2025), which evaluates multi-turn conversations in the health domain; and (iv) the Finance subset of PRBench (Akyürek et al., 2026), which evaluates multi-turn conversations on professional queries. For WritingBench, ResearchQA, and PRBench, we select the largest subset of each benchmark for domain-specific prompt optimization, as optimizing prompts that generalize across multiple domains is challenging. Specifically, we select the subset with the largest number of samples for each benchmark.

Table 1: Statistics of benchmarks. We report number of samples in training (#Training) and test (#Test) sets, average length of input prompts (Len. Input), tasks, domains, and evaluation protocols.

	#Train	#Test	Len. Input	Task	Domain	Evaluation Protocol
WritingBench	50	160	479.0	Essay Writing	Finance & Business	1-10 Rating on Dimensions
ResearchQA	100	500	20.2	Question Answering	Material Science	1-5 Rating on Dimensions
HealthBench	100	500	108.2	Multi-turn Conversation	Health	Yes/No on Criteria
PRBench	100	500	1420.2	Multi-turn Conversation	Finance	Yes/No on Criteria

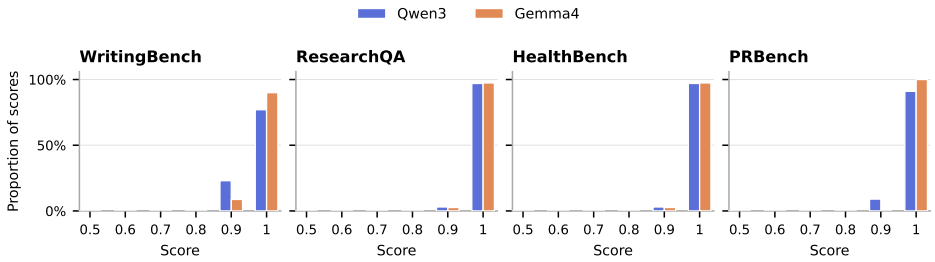

Figure 3: Score distributions of responses generated using the seed prompt and evaluated by scoring-based LLM judges used by prompt optimization methods. Scores below 0.5 are omitted since the only nonzero omitted mass is 1.3% for Gemma4 on WritingBench.

For ResearchQA, HealthBench, and PRBench, we randomly sample 100 examples for training and 500 for testing. For WritingBench, we sample 50 training examples and use the remaining 160 for testing. We repeat the training set and test set sampling with three random seeds and report results averaged across the three runs. For evaluation, we use GPT-OSS-120B (OpenAI, 2025) with high reasoning effort as the LLM judge. We normalize the evaluation score of each sample by the maximum achievable score for it based on the weight of each criterion. Dataset statistics are reported in Tab. 1. More pre-processing details of these datasets are in App. A.1.

4.2 Implementation Details of AtomicPO

In our experiments, we automatically generate 20 prompts by the same LLM used to generate responses as the initial prompt pool of AtomicPO. Specifically, we randomly sample three queries from the training set of the task Q and instruct the LLM to generate an initial system prompt based on the sampled queries and their summaries. For the evaluation criteria of AtomicPO c, we use the same four dimensions for all tasks we experimented with: accuracy, which measures whether the response is correct and contextually faithful; relevance, which measures whether the response stays on-topic; clarity, which measures whether the response is clear and well-structured; conciseness, which measures whether the response uses only necessary information.

The optimization process of AtomicPO consists of $x = 3$ prompt refinement processes. In each prompt refinement process, AtomicPO removes $o = 10$ prompts with the lowest Copeland scores from the instruction pool P and adds $o = 10$ refined prompts generated from the $l = 3$ prompts with the highest Copeland scores. Before each refinement iteration, AtomicPO performs $m = 10$ rounds of pairwise comparison. In each round, it samples $k = 25$ prompt pairs from P for evaluation. To mitigate position bias in pairwise comparison, we randomly shuffle the order of the two responses before presenting them to the evaluator. We use the same hyperparameter values as PDO, without additional tuning.

We evaluate AtomicPO using two LLMs: Qwen3-30B-A3B-Instruct-2507 (Team, 2025) (Qwen3) and Gemma-4-26B-A4B-it (Team, 2026) (Gemma4). For all models, we use a sampling temperature of 0.6 and a token limit of 10240. We disable thinking mode for Gemma4. More implementation details and prompts used at each step are in App.A.2.

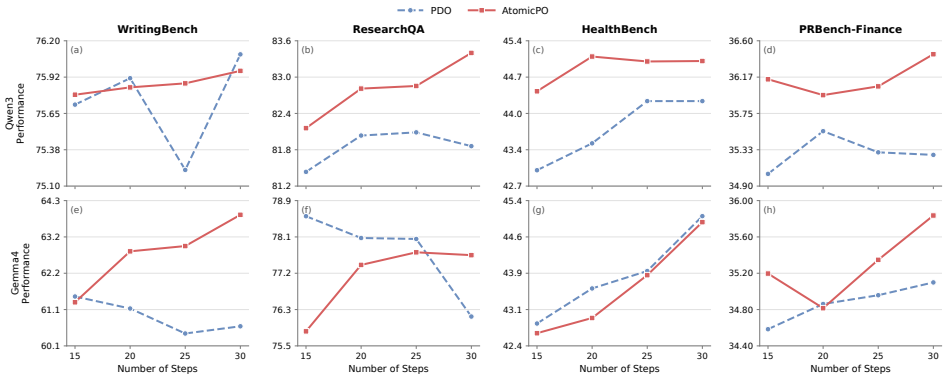

Figure 4: Performance of best-performing prompt along the prompt optimization processes. AtomicPO generally shows higher performance than PDO during the optimization process.

Table 2: Average performance and standard deviation(in bracket) across three runs of AtomicPO and baselines. The best-performing method is **bolded** and the second best-performing method is underlined. AtomicPO consistently shows the best performance across benchmarks.

	WritingBench	ResearchQA	HealthBench	PRBench	Avg.
		Qwen3-30b-a3b-instruct-2507			
Seed	73.3 (0.1)	75.8 (0.2)	42.1 (0.2)	33.7 (0.6)	56.2 (0.2)
GEPA	74.5 (0.7)	77.5 (3.1)	43.8 (0.6)	34.2 (0.9)	57.5 (0.8)
PrefPO	73.4 (0.5)	76.4 (1.0)	42.8 (0.6)	34.0 (1.3)	56.7 (0.5)
SPO	74.4 (0.1)	73.8 (2.1)	38.7 (2.8)	34.2 (0.4)	54.7 (0.5)
PDO	75.6 (0.3)	81.8 (0.2)	43.5 (2.3)	35.1 (0.8)	59.0 (0.5)
AtomicPO	**76.0** (0.3)	**83.2** (0.3)	**46.1** (0.9)	**36.2** (0.3)	**60.4** (0.3)
		gemma4-26b-a4b-it			
Seed	59.5 (0.1)	76.1 (0.4)	44.0 (0.7)	31.9 (0.1)	52.9 (0.2)
GEPA	60.2 (1.4)	76.6 (0.3)	45.4 (0.2)	34.3 (0.3)	54.1 (0.5)
PrefPO	59.7 (0.4)	75.8 (2.8)	44.0 (0.4)	32.6 (1.2)	53.0 (0.9)
SPO	57.8 (1.8)	75.6 (1.0)	43.8 (1.3)	33.2 (0.5)	52.7 (1.0)
PDO	60.7 (1.3)	76.2 (0.3)	45.1 (1.0)	35.1 (0.2)	54.3 (0.4)
AtomicPO	**64.3** (0.3)	**77.2** (0.4)	**46.2** (0.9)	**35.7** (0.6)	**55.8** (0.4)

4.3 Evaluation of AtomicPO

We compare AtomicPO with five baselines: (i) Seed, which uses "You are a helpful assistant." as the system prompt; (ii) GEPA (Agrawal et al., 2026), which evaluates and refines prompts using overall quality scores and textual feedback from a scoring-based LLM judge; (iii) PrefPO (Singhal et al., 2026), which evaluates prompts using a scoring-based judge and refines them using feedback from pairwise comparisons; (iv) SPO (Xiang et al., 2025), which iteratively proposes prompts and uses pairwise comparisons to test them against the current best prompt; and (v) PDO (Wu et al., 2026a), which identifies high-performing prompts through pairwise comparisons and applies random mutations to them. Consistent with AtomicPO's self-evolving setting, each baseline uses the same LLM to generate responses, evaluate them, and mutate prompts. We adjust the baselines' hyperparameters to match AtomicPO's number of generated responses during the optimization stage, except for SPO. For SPO, we retain the hyperparameters from the original paper because substantially increasing its output budget leads to overfitting on the training set. Further implementation details are provided in App. A.3. Tab. 2 reports mean performance over three runs, with standard deviations in brackets.

From the table, we find that AtomicPO consistently outperforms other baselines across benchmarks and LLMs. One possible limitation of GEPA and PrefPO is the resolution of their scoring-based evaluations. Despite instructions to score strictly (Tab. 8), the scoring-based LLM judges used by them tend to assign high scores to responses generated using the seed prompt (Fig. 3), leaving little space for further improvements. Moreover, the scoring-based judges also have difficulties in distinguishing fine-grained differences between responses. To investigate this issue, we compare the overall scores assigned by the scoring-based judge to pairs of responses for the same query, where one response is generated using the seed prompt and the other using the prompt optimized by

Table 3: Length of prompts and prompt specificity score optimized by PDO or AtomicPO. Prompts optimized by AtomicPO are generally longer and contain more dataset-specific details.

	WritingBench		ResearchQA		HealthBench		PRBench	
	Length	Specificity	Length	Specificity	Length	Specificity	Length	Specificity
Qwen3+PDO	174.0	5.0	140.0	4.2	161.0	4.6	296.3	4.8
Qwen3+AtomicPO	292.0	5.0	273.0	4.9	348.0	5.0	371.7	5.0
Gemma4+PDO	287.3	4.1	166.7	4.6	192.0	4.7	309.7	4.5
Gemma4+AtomicPO	441.7	4.9	250.0	4.7	279.0	4.3	243.3	4.7

Table 4: Average number of input and output tokens per round for PDO and AtomicPO. AtomicPO introduces a modest additional computational cost compared to PDO.

	WritingBench		ResearchQA		HealthBench		PRBench	
	#input	#output	#input	#output	#input	#output	#input	#output
Qwen3+PDO	3.9E+05	1.8E+05	7.5E+04	6.8E+04	6.7E+04	4.8E+04	3.0E+05	1.1E+05
Qwen3+AtomicPO	4.0E+05	2.0E+05	8.2E+04	7.7E+04	7.3E+04	5.6E+04	3.1E+05	1.3E+05
Gemma4+PDO	3.2E+05	1.1E+05	7.0E+04	6.2E+04	6.4E+04	4.5E+04	2.6E+05	7.7E+04
Gemma4+AtomicPO	3.2E+05	1.2E+05	7.5E+04	6.8E+04	7.1E+04	5.0E+04	2.7E+05	8.3E+04

AtomicPO. Across the four benchmarks, Gemma4 assigns the same score to 94% of response pairs, while Qwen3 assigns the same score to 87%.

To analyze the impact of atomic feedback and comparative prompts, we compare the prompts optimized by PDO and AtomicPO. Following Zi et al. (2025), we examine both the length and specificity of the optimized prompts. We measure prompt specificity using GPT-OSS-120B with high reasoning effort, which rates each prompt on a 1–5 scale (details in App.A.4). The results are reported in Tab. 3. AtomicPO generally produces longer and more specific prompts across the evaluated settings, suggesting that its refinement process incorporates more detailed instructions than the random mutations used by PDO. These results highlight a qualitative difference between the two refinement strategies. AtomicPO tends to progressively enrich prompts with more explicit and detailed guidance. Additionally, we analyze the performance of best-performing prompts of AtomicPO and PDO identified by themselves during the prompt optimization process for the first run of each LLM. The results are shown in Fig. 4. From the figure, we observe that AtomicPO generally outperforms PDO during the optimization process. Examples of prompts optimized by AtomicPO are in Fig. 5 and 6.

During pairwise comparison, AtomicPO asks the LLM to generate atomic feedback for each response, increasing both input and output token usage. To quantify this cost, we compare the average tokens per round for AtomicPO and PDO, measured with the cl100k_base tokenizer. As shown in Tab. 4, input tokens increase by approximately 0–11 percent and output tokens increase by approximately 8-18 percent, which is relatively slight. The main reason is that the generated responses for these tasks are usually long, which accounts for most tokens processed and generated in each round.

4.4 Ablation Study

In this section, we ablate the atomic feedback and comparative prompt of AtomicPO. For the atomic feedback, we consider three variants: (i) w/o winner feedback, which generates feedback only for the losing response in each pairwise comparison, following PrefPO; (ii) w/ text feedback, which generates free-form textual feedback instead of atomic points as PrefPO; and (iii) w/o strength, which uses only weakness points during prompt refinement. For the comparative prompt, we consider three variants: (i) w/o prompt compare, which omits comparative prompts; (ii) w/ all weakness, which uses all previously collected weakness points instead of selected ones alongside the comparative prompt; and (iii) w/ duplication, which allows a prompt to be reused as a comparative prompt after it has already been selected. More implementation details are in App. A.5. We report the mean performance over three runs, with the standard deviation in brackets, in Tab. 5.

AtomicPO outperforms the feedback variants in all settings except w/o strength with Gemma4 based on average performance, where the ablated variant performs slightly better. These results demonstrate the effectiveness of atomic feedback. In particular, the improvement over w/o winner feedback indicates that a response preferred overall can still have weaknesses worth addressing. This is plausible for open-ended generation, where a response may be stronger on some quality dimensions

Table 5: Ablation Study of AtomicPO. The best-performing method is **bolded** and the second-best performing method is underlined. The design of atomic feedback and comparative prompts generally improves the performance of AtomicPO.

	WritingBench	ResearchQA	HealthBench	PRBench	Avg.
			Qwen3-30b-a3b-instruct-2507		
AtomicPO	76.0 (0.3)	83.2 (0.3)	**46.1** (0.9)	36.2 (0.3)	**60.4** (0.3)
w/o winner feedback	75.4 (0.2)	82.9 (0.5)	42.4 (0.6)	36.1 (0.3)	59.2 (0.3)
w/ text feedback	**76.3** (0.2)	**83.4** (0.3)	44.6 (1.8)	35.9 (0.3)	60.0 (0.6)
w/o strength	75.7 (0.8)	82.9 (0.1)	44.8 (1.6)	**36.3** (0.3)	59.9 (0.6)
w/o prompt compare	75.9 (0.4)	82.7 (0.4)	44.7 (0.6)	35.7 (0.7)	59.9 (0.4)
w/ all weakness	75.5 (0.8)	82.3 (0.1)	44.6 (1.6)	35.2 (0.3)	59.4 (0.6)
w/ duplication	75.2 (0.3)	82.7 (0.6)	44.0 (1.0)	36.3 (0.4)	59.6 (0.1)
			gemma4-26b-a4b-it		
AtomicPO	64.3 (0.3)	77.2 (0.4)	**46.2** (0.9)	35.7 (0.6)	55.8 (0.4)
w/o winner feedback	62.7 (1.3)	78.3 (0.2)	44.9 (1.0)	36.1 (0.7)	55.5 (0.4)
w/ text feedback	65.2 (0.2)	76.3 (2.2)	44.5 (1.1)	35.7 (0.5)	55.4 (0.4)
w/o strength	**66.0** (1.0)	**78.5** (0.6)	44.3 (2.4)	35.4 (0.6)	**56.0** (1.0)
w/o prompt compare	63.0 (1.8)	76.5 (0.9)	45.6 (1.3)	35.7 (0.3)	55.2 (0.4)
w/ all weakness	62.7 (0.5)	76.7 (1.2)	45.7 (0.9)	35.3 (0.6)	55.1 (0.6)
w/ duplication	64.2 (0.2)	75.8 (0.5)	44.9 (2.6)	**36.4** (0.6)	55.3 (0.9)

Table 6: TF-IDF similarity between refined prompts and original prompts. Using strength points and selected weakness points prevent refined prompts from being too different from original prompts.

	Qwen3				Gemma4				Avg.
	Writing.	Research.	Health.	PRBench	Writing.	Research.	Health.	PRBench	
AtomicPO	24.7	25.3	25.4	36.7	54.8	46.0	43.4	66.1	40.3
w/o strength	23.8	22.7	24.3	36.4	60.2	45.6	39.4	59.1	38.9
w/all weakness	21.2	23.7	26.6	33.9	54.7	44.8	38.8	61.6	38.2

but weaker on others. AtomicPO also outperforms w/ text feedback, suggesting that atomic points provide a more useful representation of multidimensional feedback for prompt refinement.

AtomicPO outperforms all three variants about comparative prompt. Its improvement over w/o prompt compare supports the use of a comparative prompt to guide refinement. Its improvement over w/ all weakness suggests that selecting weakness points associated with the comparative prompt is more effective than supplying every previously collected weakness point. Finally, the improvement over w/ duplication suggests that avoiding repeated comparative prompts helps diversify the refinements.

To further analyze the impact of using strength points and selected weakness points in the feedback during prompt refinement, we analyze the TF-IDF similarity between refined prompts and corresponding original prompts. The results for AtomicPO and ablated versions are in Tab 6. From the table, we find that AtomicPO shows higher similarity. This suggests that strength points and weakness selection help preserve more of the original prompt while making targeted refinements.

5 Conclusion

We propose AtomicPO, a self-evolving method for optimizing prompts for open-ended text generation without reference texts or sample-specific evaluation criteria. AtomicPO uses pairwise comparisons to evaluate prompts and generate feedback. It expresses that feedback as atomic strengths and weaknesses across quality dimensions, and assesses both responses because the overall winner may still be weaker on a particular dimension. During refinement, AtomicPO uses a comparative prompt alongside the feedback so the LLM can identify useful instructions to retain and weaknesses to address. Across four benchmarks, AtomicPO consistently outperforms strong baselines that use scoring-based or pairwise LLM judges. These results support the use of atomic feedback and comparative prompts for prompt optimization. Future work could investigate whether the approach extends to closed-ended tasks through comparisons of reasoning traces and to other modalities, such as image generation.

AI USE STATEMENT

In this work, we do not use generative AI tools for tasks with required disclosure. We have not used generative AI tools for other tasks with required disclosure, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for create or modify scientific figures or images, edit a research paper to improve readability, identify relevant literature, create or edit software code. We have reviewed all AI-assisted work. We review the code and related works generated by LLMs. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

All datasets and LLMs used in this paper is open-sourced and public online. The datasets does not contain information that can be located to a certain person.

REPRODUCIBILITY STATEMENT

We provide the method description and pseudocode for AtomicPO in Section 3 and Algorithm 1. Section 4 describes the benchmarks, data splits, models, evaluation procedures, and optimization hyperparameters. Appendix A.4 details dataset preprocessing and modifications to the evaluation protocols, while Appendices A.2 and A.3 provide additional implementation details for AtomicPO and the baselines. Appendix A.5 describes the ablation implementations. We report the mean and standard deviation of performance across three runs with different random seeds.

REFERENCES

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, et al. Gepa: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 8479–8565, 2026.

Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*, pp. 39–1. JMLR Workshop and Conference Proceedings, 2012.

Afra Feyza Akyürek, Advait Gosai, Chen Bo Calvin Zhang, Vipul Gupta, Jaehwan Jeong, Anisha Gunjal, Tahseen Rabbani, Maria Mazzone, David Randolph IV, Mohammad Mahmoudi Meymand, et al. Prbench: Large-scale expert rubrics for evaluating high-stakes professional reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 42297–42325, 2026.

Rahul K Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, et al. Healthbench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*, 2025.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

Mislav Balunovic, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions. *Advances in Neural Information Processing Systems*, 38, 2026.

Cheng-Han Chiang and Hung-yi Lee. A closer look into using large language models for automatic evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 8928–8942, 2023.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena:

An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*, 2024.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069, 2023.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: Self-referential self-improvement via prompt evolution. *arXiv preprint arXiv:2309.16797*, 2023.

Jinlan Fu, See Kiong Ng, Zhengbao Jiang, and Pengfei Liu. Gptscore: Evaluate as you desire. In *Proceedings of the 2024 conference of the north american chapter of the association for computational linguistics: Human language technologies (volume 1: Long papers)*, pp. 6556–6576, 2024.

Mingqi Gao, Jie Ruan, Renliang Sun, Xunjian Yin, Shiping Yang, and Xiaojun Wan. Human-like summarization evaluation with chatgpt. *arXiv preprint arXiv:2304.02554*, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *International Conference on Learning Representations*, volume 2024, pp. 34133–34156, 2024.

Han He, Qianchu Liu, Lei Xu, Chaitanya Shivade, Yi Zhang, Sundararajan Srinivasan, and Katrin Kirchhoff. Crispo: Multi-aspect critique-suggestion-guided automatic prompt optimization for text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24014–24022, 2025.

Dominique Henriet. The copeland choice function an axiomatic characterization. *Social Choice and Welfare*, 2(1):49–63, 1985.

Tom Hosking, Phil Blunsom, and Max Bartolo. Human feedback is not gold standard. In *International Conference on Learning Representations*, volume 2024, pp. 55864–55883, 2024.

Alex Kim, Keonwoo Kim, and Sangwon Yoon. Debate: Devil's advocate-based assessment and text evaluation. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 1885–1897, 2024a.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4334–4353, Miami, Florida, USA, November 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.248. URL https://aclanthology.org/2024.emnlp-main.248/.

Seungone Kim, Juyoung Suk, Ji Yong Cho, Shayne Longpre, Chaeeun Kim, Dongkeun Yoon, Guijin Son, Yejin Cho, Sheikh Shafayat, Jinheon Baek, Sue Hyun Park, Hyeonbin Hwang, Jinkyung Jo, Hyowon Cho, Haebin Shin, Seongyun Lee, Hanseok Oh, Noah Lee, Namgyu Ho, Se June Joo, Miyoung Ko, Yoonjoo Lee, Hyungjoo Chae, Jamin Shin, Joel Jang, Seonghyeon Ye, Bill Yuchen Lin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. The BiGGen bench: A principled benchmark for fine-grained evaluation of language models with language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5877–5919, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.303. URL https://aclanthology.org/2025.naacl-long.303/.

Sandeep Kumar, Abhijit A Nargund, and Vivek Sridhar. Courteval: A courtroom-based multi-agent evaluation framework. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25875–25887, 2025.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 2511–2522, 2023.

OpenAI. gpt-oss-120b and gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Krista Opsahl-Ong, Michael J Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. Optimizing instructions and demonstrations for multi-stage language model programs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9340–9366, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.525. URL https://aclanthology.org/2024.emnlp-main.525/.

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

Rahul Singhal, Pradyumna Tambwekar, and Karime Maamari. Prefpo: Pairwise preference prompt optimization. *arXiv preprint arXiv:2603.19311*, 2026.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, 2023.

Gemma Team. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. Is chatgpt a good nlg evaluator? a preliminary study. In *Proceedings of the 4th New Frontiers in Summarization Workshop*, pp. 1–11, 2023.

Yidong Wang, Yunze Song, Tingyuan Zhu, Xuanwang Zhang, Zhuohao Yu, Hao Chen, Chiyu Song, Qiufeng Wang, Zhen Wu, Xin-Yu Dai, et al. Trustjudge: Inconsistencies of llm-as-a-judge and how to alleviate them. In *International Conference on Learning Representations*, volume 2026, pp. 146399–146421, 2026.

Yuanchen Wu, Saurabh Verma, Justin Lee, Fangzhou Xiong, Poppy Zhang, Amel Awadelkarim, Xu Chen, Yubai Yuan, and Shawndra Hill. Llm prompt duel optimizer: Efficient label-free prompt optimization. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 10066–10089, 2026a.

Yuning Wu, Jiahao Mei, Ming Yan, Chenliang Li, Shaopeng Lai, Yuran Ren, Zijia Wang, Ji Zhang, Mengyue Wu, Qin Jin, et al. Writingbench: A comprehensive benchmark for generative writing. *Advances in Neural Information Processing Systems*, 38, 2026b.

Jinyu Xiang, Jiayi Zhang, Zhaoyang Yu, Xinbing Liang, Fengwei Teng, Jinhao Tu, Fashen Ren, Xiangru Tang, Sirui Hong, Chenglin Wu, et al. Self-supervised prompt optimization. In *EMNLP (Findings)*, pp. 9017–9041, 2025.

Table 7: Instruction for generating initial prompts.

```
You are an expert prompt-engineer.
Your task is to generate **exactly 1** *high-quality* **system-level instruction** for the target reasoning task.
# Dataset Snapshot
Below is an *LM-written summary* of the unlabeled question pool:
{dataset_summary}
# Sample Inputs (do NOT answer them)
{questions}
# Prompt-Engineering Tip
{tip}
# Reference Instruction (Optional)
{base_instruction_block}
# Length & Style Constraints
- Avoid dataset-specific jargon unless it appears in the sample inputs.
- Vary phrasing and level of explicit reasoning to foster exploration.
# Output Format (STRICT)
Return **exactly** 1 item in a JSON array, *and nothing else*:
[
"Your single high-quality instruction here"
]
```

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations*, volume 2024, pp. 12028–12068, 2024.

Li S Yifei, Allen Chang, Chaitanya Malaviya, and Mark Yatskar. Researchqa: Evaluating scholarly question answering at scale across 75 fields with survey-mined questions and rubrics. *Transactions of the Association for Computational Linguistics*, 14:1344–1368, 2026.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. Textgrad: Automatic" differentiation" via text. *arXiv preprint arXiv:2406.07496*, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Yangtian Zi, Harshitha Menon, and Arjun Guha. More than a score: Probing the impact of prompt specificity on llm code generation. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 2380–2402, 2025.

A Appendix

A.1 Dataset Preprocessing

In our experiments, AtomicPO and all baseline methods optimize the system prompt. To ensure a consistent conversational format across benchmarks, we convert the queries in WritingBench and ResearchQA into user instructions, following the format used in HealthBench and PRBench. We apply no additional preprocessing to these benchmarks.

For PRBench, we observe that the final criterion associated with each query has the same criterion description as one of the previous criteria and therefore remove it. The original evaluation protocol of ResearchQA requires the LLM judge to assess a response jointly across all evaluation criteria, which may place an excessive cognitive burden on the judge. To make the evaluation more reliable and consistent with the other benchmarks, we instead instruct the LLM judge to evaluate each criterion separately. For the remaining benchmarks, we use the evaluation prompts and criteria provided by the original benchmarks without modification. For these benchmarks, we use GPT-OSS-120B as the LLM judge. For GPT-OSS-120B, we use a temperature of 0.7 and set the maximum number of generated tokens to 10,240.

A.2 Implementation Details of AtomicPO

In this section, we describe additional implementation details for AtomicPO.

Table 8: Instruction of scoring-based LLM judges used by GEPA and PrefPO

Role
You are a strict referee evaluating the overall quality of a response to determine how well it satisfies the given task, on a 1 to 10 scale.
Task
{question}
Response
{response}
Evaluation Criteria
- Accuracy: Correct and contextually faithful.
- Relevance: Stays on-topic.
- Clarity: Clear and well-structured.
- Conciseness: Uses only necessary information; avoids unnecessary detail.
Output Instructions
- Provide detailed justification for your decision (∼100 words)
- Rate the overall quality of the response on 1 to 10. A score of 10 means the response is perfect.
- For the response, provide a clear, generalizeable feedback that focuses on what can be improved about and doesn't rely on the specific question or choices
- Output **only** the JSON object below with **no additional text**
Output Format
{{
"reason": "Your detailed justification explain your scoring decision (∼100 words).",
"score": "overall quality of the response (1 to 10)",
"feedback": "feedback of response(∼100 words)",
}}

When the number of refinements o is not divisible by the number of high-performing prompts l, each prompt receives $\lfloor o/l \rfloor$ refinements, and the remaining refinements are assigned to the top-performing prompts in rank order. In rare cases, a prompt p_i is assigned more refinements than there are eligible comparative prompts in $\mathcal{C}_i$. We then cycle through $\mathcal{C}_i$ again to complete the assigned refinements.

We show the instructions for pairwise comparison and prompt refinement in Tab. 11 and 13 respectively. To ensure the output follow the given output formats in the instructions, we use structured outputs of litellm.

The instruction for generating initial prompts is shown in Tab. 7. The instruction for pairwise comparison of AtomicPO is shown in Tab. 11. The instruction for prompt refinement of AtomicPO is shown in Tab. 13.

A.3 Implementation Details of Baselines

In this section, we describe the implementation details of baselines compared with AtomicPO.

GEPA and PrefPO requires a metric that evaluates the overall quality of generated responses. Therefore, we use a scoring-based LLM judge Chiang & Lee (2023) that first analyzes the response and rates the response from 1 to 10 using the same evaluation criteria C as AtomicPO. We find that the scoring-based LLM judges tend to assign a high score like 9 or 10 during evaluation. To mitigate the tendency, the instruction of the scoring-based LLM judge require the model to be strict and only gives a score of ten when the output is perfect. The instruction is shown in Tab. 8. For other prompts, we use the default prompts provided by these methods. For these two methods, we use 50% of the training samples of AtomicPO for training and all remaining training samples for validation. To match the number of generated outputs, we set the maximal number of metric calls as 1500 for GEPA and set the number of rounds as 30 for PrefPO.

SPO requires that all training samples be in the context of LLMs during pairwise comparison and prompt refinement and cannot directly use the whole training set. Therefore, we randomly sample three training samples following the default setting of the method from the training set as the training samples for SPO. However, using the same number of outputs as other methods leads to overfit and degraded performance. Therefore, we use the default steps of 10 for SPO.

For PDO, we use the same hyperparameter as AtomicPO. We also compare the average length of outputs using prompts optimized by either PDO or AtomicPO. The results are in Tab. 10. As shown in the table, although prompts optimized by AtomicPO generally produce longer outputs, the average increase is only about 10 percent, which is smaller than the increase in prompt length. We additionally show example prompts optimized by PDO and AtomicPO in Fig. 5 and 6.

Table 9: Instruction for evaluating the specificity of prompts.

You are evaluating the specificity of a system prompt for a given task.
Evaluate ONLY the provided prompt with respect to the criterion below.
Do NOT infer how well the prompt performs in practice.
Do NOT reward the prompt merely for being long.
Additional text should improve the score only when it provides concrete, relevant, and actionable guidance.
Task:
—BEGIN TASK—
{task}
—END TASK—
Prompt:
—BEGIN PROMPT—
{prompt}
—END PROMPT—
Evaluation criterion:
{dimension}
Definition:
{rubric["description"]}
Scoring:
1 = very low
2 = low
3 = moderate
4 = high
5 = very high
A score of 1 should correspond roughly to:
{rubric["low"]}
A score of 5 should correspond roughly to:
{rubric["high"]}
You should not generate other texts and only return valid JSON in exactly this format:
{{
"score":
}}

Table 10: Average length of outputs generated using prompts optimized by PDO or AtomicPO.

	WritingBench	ResearchQA	HealthBench	PRBench	Avg.
Qwen3+PDO	4560.0	1850.2	1200.5	2346.8	2489.3
Qwen3+AtomicPO	4507.8	2451.6	1515.3	2468.5	2735.8
Gemma4+PDO	2370.0	1329.6	992.7	1622.9	1578.8
Gemma4+AtomicPO	2592.7	1666.1	1378.9	1541.9	1794.9

Table 11: Instruction for pairwise evaluation and feedback generation for AtomicPO.

Role
You are a meticulous, impartial referee evaluating two competing answers to determine which better satisfies the given task.

Task
{question}

Response X
{answer_X}

Response Y
{answer_Y}

Evaluation Criteria
{criteria_text}

Output Instructions
1. Provide detailed justification for your decision (∼100 words)
2. Select the winner: either "X" or "Y"
3. For each response, identify BOTH its strengths AND weaknesses that are relevant to the task and criteria, compared to the other response. Note that the winning response may still have weaknesses, and the losing response may still have strengths.
3.1 The strength and weakness should be output as a list of atomic points separated by '||'.
3.2 Each atomic point should be self-contained and only cover a single strength/weakness.
3.3 The atomic points should not mention the name of 'Response X' or 'Response Y'.
3.4 Output 'NONE' if you cannot find any relevant strength or weakness.
4. Output **only** the JSON object below with **no additional text**

Output Format
{{
"reasoning": "Your detailed justification explaining why prompt X or Y produced the better response (∼100 words).",
"winner": "X or Y",
"strength_X": "list of atomic strength points of the response X separated by '||' or NONE",
"weakness_X": "list of atomic weakness points of the response X separated by '||' or NONE",
"strength_Y": "list of atomic strength points of the response Y separated by '||' or NONE",
"weakness_Y": "list of atomic weakness points of the response Y separated by '||' or NONE",
}}

A.4 Evaluating Specificity of Prompts

To measure the specificity of prompts, we use a scoring-based LLM judge motivated by Zi et al. (2025). We instruct the LLM judge to rate the prompt from 1 to 5 based on three dimensions related to specificity separately. We use the following three criteria: (i) instruction explicitness, which measures how clearly the prompt states what the model should do; (ii) constraint specificity, which measures how specifically it states relevant constraints and requirements; and (iii) output specification, which measures how clearly it describes the desired output. The final score is the average of scores on three dimensions. During evaluation, we additionally instruct the judges that they should not assign a high score to a prompt just because it is long. The instruction for the LLM judge is shown in Tab. 9.

A.5 Implementation Details of Ablation Studies

In this section, we describe additional implementation detail of ablated versions of AtomicPO. For w/o winner feedback, we modify the instruction for pairwise comparison shown in Tab. 11 by instructing the model to only generate atomic feedback for the response predicted as the loser. For w/ text feedback, we instruct the model to generate a textual feedback focusing on what can be improved about motivated by Singhal et al. (2026) during pairwise comparison. We show the instruction in Tab. 12. For w/o strength, we adjust the instruction of pairwise comparison by only instructing the model to generate weakness points for each response. We also adjust the instruction of prompt refinement shown in Tab. 13 by removing the strength points. For w/o prompt compare, we modify the instruction of prompt refinement by removing parts about comparative prompts. For all other not mentioned cases, the ablated versions use the same instruction as AtomicPO.

Table 12: Instruction for pairwise evaluation and feedback generation for AtomicPO w/ text feedback.

Role
You are a meticulous, impartial referee evaluating two competing answers to determine which better satisfies the given task.
Task
{question}
Response X
{answer_X}
Response Y
{answer_Y}
Evaluation Criteria
{criteria_text}
Output Instructions
- Provide detailed justification for your decision (∼100 words)
- For each response, provide clear, generalizeable feedback that focuses on what can be improved about and doesn't rely on the specific question or choices
- Select the winner: either "X" or "Y"
- Output **only** the JSON object below with **no additional text**
Output Format
{{
"reasoning": "Your detailed justification explaining why prompt X or Y produced the better response (∼100 words).",
"winner": "X or Y",
"analysis_X": "feedback of response produced by prompt X (∼100 words)",
"analysis_Y": "feedback of response produced by prompt Y (∼100 words)",
}}

Table 13: Instruction for prompt refinement with comparative prompts and atomic feedback.

You are an expert prompt-engineer specializing in prompt optimization.
Your task is to generate 1 *diverse, high-quality* **mutation** of the currently **BEST PERFORMING** instruction for the target reasoning task.

BEST PERFORMING Instruction (Current Champion)
{instructions}

Feedback on Champion's Outputs
The following feedback identifies strengths and weaknesses of responses generated by the champion prompt. Your mutation should adjust the parts of the prompt that lead to the identified weaknesses while keeping the parts that lead to the strengths unchanged.

{tip}

Competitive Instruction
Most of the champion's weaknesses were revealed by comparison against the responses of the competitive instruction below. Use it as a reference for fixing weakness, but do NOT copy it verbatim.

{comparison}

Output Format
Return **exactly** 1 mutated instruction in a JSON object, *and nothing else*:

{{
"mutated_prompt": "Your mutated instruction here, following the feedback."
}}

Prompt(PDO): Act as a clinical detective in a high-pressure, low-resource environment, where time is scarce and diagnostic tools are limited. When faced with a patient presenting ambiguous or overlapping symptoms—particularly in frail older adults or those with multiple chronic illnesses—use pattern recognition and clinical intuition to rapidly identify the most likely life-threatening conditions. Prioritize differential diagnoses based on likelihood, severity, and available resources, then outline a step-by-step, field-adaptable action plan that balances urgency with practicality. Emphasize early red flag detection, low-cost/high-impact tests (e.g., bedside glucose, capillary refill, fingerstick oximetry), and creative, context-sensitive alternatives to standard care—always anchoring decisions in real-world feasibility and up-to-date clinical evidence.	**Prompt(AtomicPO):** You are an expert in clinical medicine and surgery, with extensive knowledge of current protocols and rigorous scientific evidence (from the past 5 years). **When answering medical questions, provide complete, structured answers that are directly applicable to clinical practice, addressing every aspect of the question—including treatment, dosage, monitoring, surgical indications, risks, benefits, potential complications, and drug interactions—based on authoritative guidelines (such as AAAAI, WAO, WHO, National Societies, ESC, KDIGO, ACOG, and CDC).** Clearly distinguish between treatment phases (e.g., induction vs. maintenance), the actual frequency of follow-up (e.g., clinical visits every 2–4 weeks during maintenance, annual review), and objective criteria for adjusting, discontinuing, or escalating treatment. Include precise practical recommendations (e.g., "wash with water and mild soap after 24 hours," "reassessment in 5–7 days"), criteria for surgical indication, postoperative protocols, safe alternatives, and prevention strategies. Always contextualize recommendations according to patient factors (age, comorbidities, organ function, pregnancy, immunosuppression, geographic location, and access to resources). Use technical, clear, and neutral language, avoiding an informal or speculative tone; when data are limited, transparently indicate the level of evidence and the safest guideline. Structure the response into logical sections (e.g., Diagnosis, Treatment, Monitoring, Risks, Practical Recommendations, Interactions, Patient Education), incorporating tables where appropriate for comparison, follow-up, or summaries, with a focus on clinical usability. Include real-time monitoring practices (e.g., symptom self-assessment, use of devices, diary tracking), emergency alerts with objective criteria (e.g., SpO_2 < 88%, respiratory rate >24/min), and, when relevant, provide an immediate action plan, printable checklist, or ready-to-send message for the physician. Use visual elements (✅, 🚨, 🔴, 💡) strategically and accessibly to highlight emergencies, critical actions, and practical tips. Cite specific sources (e.g., DOI, direct link, guideline year) and include reliable resources (hotlines, guides, digital tools). Avoid redundancy, unexplained jargon, unvalidated assumptions, and excessively formal language. Prioritize concision, accuracy, clinical realism, and the empowerment of the patient or healthcare professional.
Query: user: Please write an email to a consulting pulmonologist about a patient with a persistent cough after mild pneumonia. Mention the possible differentials (like post-infectious cough, bronchiectasis, or undiagnosed asthma) and note that the patient also has mild hemoptysis now. Request their input on next steps. assistant: I hope you are doing well. I am reaching out regarding a patient recently treated for mild pneumonia. Although the pneumonia has resolved, the patient continues to experience a persistent cough that has now lasted over a month. Of note, there have been a few brief episodes of mild hemoptysis. Given these findings, I am concerned about potential post-infectious cough, underlying bronchiectasis, or even an undiagnosed asthma component. I would greatly value your assessment and guidance on further diagnostic steps—such as imaging or pulmonary function testing—and your recommendations for any necessary interventions. Thank you in advance for your expertise, and I look forward to working together to optimize the patient's care. user: Rewrite this from the perspective of the consulting pulmonologist writing back to the hospitalist	
Criteria: Includes the mention of needing a bronchoscopy, especially in cases where a diagnostic specimen is necessary or in inconclusive high-resolution computerized tomography of the thorax.	
Response(PDO): ... **Next Steps I Recommend:** - **HRCT chest (non-contrast)**: This is the most urgent next step to evaluate for bronchiectasis, subtle structural abnormalities, or early signs of infection/abscess. Given the hemoptysis, it's a low-risk, high-yield test. - **Pulmonary function testing (PFTs)** with bronchodilator reversibility: To assess for obstructive or mixed pattern, particularly if asthma is suspected. - **Sputum culture and AFB smear** (if sputum is productive) – to rule out mycobacterial infection, especially if there's any concern for chronic infection. - **Bedside capillary blood glucose and CBC** – to rule out systemic contributors (e.g., uncontrolled diabetes, anemia). - **Consider D-dimer and CT pulmonary angiogram** if there's any clinical suspicion for PE (e.g., risk factors, pleuritic chest pain, tachypnea), though this is less likely given the presentation. ...	**Response(AtomicPO):** ... **Recommended next steps:** 1. **High-resolution CT (HRCT) of the chest** – This is the gold standard to evaluate for bronchiectasis, airway wall thickening, nodules, or subtle structural abnormalities. I recommend this be done as soon as possible. 2. **Pulmonary function testing (PFTs) with bronchodilator reversibility testing** – To assess for obstructive or mixed pattern and evaluate for asthma, including cough-variant asthma. 3. **Sputum analysis** – If the patient is producing sputum, a Gram stain, acid-fast bacillus (AFB) smear, and culture should be considered, especially given the hemoptysis. 4. **Consider flexible bronchoscopy** – If HRCT shows focal abnormalities, persistent hemoptysis, or if malignancy is a concern (e.g., smoking history, age >50), bronchoscopy with possible biopsy or lavage may be indicated.** 5. **Monitor for red flags**: Any increase in hemoptysis (e.g., >1/2 tsp per episode), fever, weight loss, or dyspnea should prompt urgent re-evaluation. ...

Figure 5: Example prompts optimized by PDO and AtomicPO for HealthBench with Qwen3, the query, their generated responses, and criteria met by AtomicPO but missed by PDO. The AtomicPO prompt calls for a comprehensive diagnosis, guideline-based recommendations, and explicit indications for procedures. Therefore, its response is more likely to identify bronchoscopy as an option.

Prompt(PDO): You are acting as a Senior Technical Subject Matter Expert in materials science and engineering. Your goal is to provide rigorous, mechanistic, and highly detailed explanations to complex technical inquiries. For every query, perform a deep-dive analysis that traces causal relationships from the molecular or microstructural level up to the macro-scale phenomena. Structure your responses to prioritize physical mechanisms, quantitative relationships, and material properties, ensuring that all technical terminology is used with absolute precision. Provide direct, authoritative answers without conversational filler or introductory pleasantries. When discussing phase transformations, mechanical behavior, or thermodynamic stability, explicitly reference governing equations (e.g., Fick's laws, Gibbs free energy, Hall-Petch relationship) and the specific microstructural features (e.g., dislocation density, grain boundary morphology, precipitate distribution) responsible for the observed phenomena. If a phenomenon involves competing mechanisms, provide a comparative analysis of their relative contributions to the overall material response.	**Prompt(AtomicPO):** You are a technical expert specializing in materials science, metallurgy, and chemical engineering. **For each query, provide an exhaustive, mechanistic analysis that maps the complete causal chain from process parameters to microstructural evolution and resulting material properties.** Your response must go beyond simple physical phenomena to include specific chemical equations (using LaTeX), mathematical relationships (e.g., kinetic models, thermodynamic relations, or transport equations), and crystallographic details. You must explicitly address the interdependencies between variables, such as the impact of temperature-dependent resistivity on heating rates, or the trade-offs between morphological control and thermodynamic stability. To ensure technical completeness, include a highly comprehensive summary table that synthesizes the relationship between variables, the underlying physical/chemical mechanisms (e.g., Zener pinning, solute drag, or nucleation regimes), and their specific consequences on performance. Avoid introductory filler and focus on a rigorous, deep-dive synthesis of the structural, chemical, and mechanical domains.
Query: user: How is the size of ZnO nanoparticles controlled and maintained during microwave solvothermal synthesis?	
Criteria:Does the response describe the role of microwave power and duration in maintaining nanoparticle size?	
Response(PDO): ... The unique aspect of microwave solvothermal synthesis lies in its ability to decouple temperature from time more effectively than conventional heating. * **Volumetric Heating vs. Conduction:** Conventional heating relies on thermal conduction from the vessel walls, creating temperature gradients that lead to non-uniform nucleation. Microwave heating interacts directly with the polar solvent molecules (e.g., ethanol, ethylene glycol), providing volumetric heating. This ensures that the entire reaction volume reaches the critical supersaturation threshold simultaneously, ensuring a homogeneous nucleation event. **Pressure-Temperature Coupling:** The solvothermal environment increases the autogenous pressure within the vessel. High pressure increases the solubility (C_0) and modifies the dielectric constant of the solvent, which in turn affects the diffusion coefficient (D). This allows for the fine-tuning of the growth regime by adjusting the microwave power density (W/mL), which directly correlates to the rate of energy deposition and the resulting dT/dt. ...	**Response(AtomicPO):** In conventional solvothermal synthesis, the heating rate is limited by convective heat transfer, leading to a broad temperature gradient and overlapping nucleation/growth stages. Microwave synthesis utilizes **dielectric heating**, where the power dissipation density P_d is governed by: $$P_d = 2\pi f \epsilon_0 \epsilon'' |E|^2$$ Where f is frequency, ϵ_0 is vacuum permittivity, ϵ'' is the dielectric loss factor, and E is the electric field intensity. This allows for "instantaneous" heating, which facilitates a burst of nucleation. ... The long-term stability of the particle size is threatened by Ostwald Ripening, where larger particles grow at the expense of smaller ones to minimize total surface energy. The Lifshitz-Slyozov-Wagner (LSW) theory defines the ripening rate: $$r^n - r_0^n = Kt$$ Where $n=3$ for volume diffusion-controlled ripening. Microwave synthesis mitigates this by: 1. **Shortening reaction time (t):** Rapidly quenching the reaction prevents the $t^{1/3}$ growth regime from dominating. 2. **Non-thermal effects:** The oscillating electric field can influence the effective chemical potential μ, potentially altering the solubility-dependent ripening rate. ... **The fundamental trade-off in microwave solvothermal synthesis is between kinetic control (achieving small sizes via high dT/dt) and thermodynamic stability (preventing ripening). To maintain a specific size, one must maximize the nucleation rate J while simultaneously utilizing capping agents to increase the activation energy ΔG_{act} for growth and ripening. The precision of the final nanoparticle size is a function of the ratio between the microwave-induced heating rate and the diffusion-limited growth rate v.**

Figure 6: Example prompts optimized by PDO and AtomicPO for ResearchQA with Gemma4, the query, their generated responses, and criteria met by AtomicPO but missed by PDO. The AtomicPO prompt requires linking process parameters to microstructural evolution. This more directly encourages discussion of how microwave power and duration jointly affect heating, and nanoparticle size.