Multi-View Self-Optimization for Language Agents

Anonymous ACL submission

Abstract

Language agents often produce a plausible first attempt that still contains weaknesses visible from alternative viewpoints. Human writers and problem solvers rarely revise from a single undifferentiated perspective; instead, they separately inspect correctness, completeness, constraint satisfaction, and presentation before rewriting. We study whether a single text model can benefit from the same inductive bias at inference time. We propose **Multi-View Self-Optimization** (MVSO), a lightweight single-backbone framework that generates a draft, critiques it through task-adaptive viewpoints, synthesizes the critiques into one revision plan, and then rewrites once. We further introduce **MVSO-Bench**, a reproducible text-first benchmark suite for multi-view agent optimization, spanning easy pilot tasks, controllable rewriting, and harder reasoning subsets with exact-match and constraint-aware metrics. On a hard 80-example `MATH-500` subset with a local `Qwen3-4B-Instruct-2507` checkpoint, MVSO improves exact match from 0.1125 (single-pass) and 0.1625 (generic self-refine) to 0.3125. However, the same structure does not help a weaker `Gemma-4-E2B-it` checkpoint on the same benchmark, and a fast `bbh_hard` validation instead favors generic self-refinement over MVSO. These results suggest that explicit viewpoint decomposition is a real optimization axis for agents, but its benefit is backbone-dependent and task-dependent, with the clearest gains appearing when tasks expose multiple failure modes that a single critique prompt does not reliably separate.

1 Introduction

The first answer from a language agent is often "locally plausible" but globally under-optimized. A math solution may have the right plan but one arithmetic mistake. A constrained rewrite may be fluent while quietly violating a hard lexical restriction. A long-form answer may be mostly correct while underserving a particular audience or evaluation rubric. These are not failures of raw generation alone; they are failures of *inspection*.

Human revision is naturally multi-view. When we improve a draft, we do not merely say "reflect and revise." We switch viewpoints. One pass checks logic. Another checks whether all requirements are satisfied. Another pass edits style or compresses the answer. This perspective-switching behavior is especially common in tasks with competing objectives, where improving one dimension can easily damage another.

This observation motivates our central research question: *can a single language model improve its own outputs more reliably when we require it to inspect the draft from several explicit task-adaptive viewpoints before revision?* We focus on a practical inference-time setting rather than a new training regime. The goal is a drop-in mechanism that is cheaper than multi-agent debate and more structured than one generic self-correction prompt.

We propose **Multi-View Self-Optimization** (MVSO), a four-stage framework: produce a draft, critique the draft from several specialized viewpoints, synthesize the resulting critiques into a single revision plan, and then rewrite once. The method uses one backbone model throughout. Its complexity comes not from external tools or search, but from forcing the model to decompose revision into several perspectives that target different failure modes.

The empirical story that emerges is not "multi-view always wins." Instead, it is more interesting. On easy reasoning and rewriting pilots, generic self-refinement is already strong. On a harder `MATH-500` subset, however, MVSO substantially improves the stronger Qwen backbone while barely helping Gemma. This means the main question is not simply whether more reflective turns help, but whether explicit viewpoint allocation changes what the model can notice and repair.

Contributions. We make four contributions:

1. We introduce MVSO, a simple single-backbone framework for explicit multi-view critique and revision at inference time.

2. We introduce **MVSO-Bench**, a text-first benchmark suite for multi-view agent optimization, spanning sampled reasoning, hard symbolic reasoning, and controllable rewriting with automatic metrics for exact match, hard constraints, and source preservation.

3. We report the first local empirical study showing that MVSO can produce large gains on hard reasoning for one backbone while failing to transfer uniformly across models.

4. We show that the method remains testable under tight compute budgets through fast validation subsets, which is useful for small-scale but reproducible agent research.

2 Related Work

Self-refinement and reflective prompting are the closest starting points for this work. Madaan et al. (2023) showed that models can critique and improve their own outputs through iterative feedback. Reflexion (Shinn et al., 2023) extended this style of verbal feedback to agent settings, while later work questioned how robust intrinsic self-correction really is without additional structure (Liu et al., 2024). MVSO belongs to this family, but asks a narrower design question: can we improve self-correction by replacing one generic critique with several targeted viewpoints?

Another related line explores deliberate search over reasoning states. Tree of Thoughts (Yao et al., 2023), Graph of Thoughts (Besta et al., 2024), and related search-style methods improve reasoning by expanding and selecting among intermediate trajectories. These approaches are powerful but often more expensive than the local single-model workflow we target here. MVSO is intentionally lighter-weight: it does not search over many candidate answers, but it does ask the model to inspect one candidate through several lenses before revising.

The third connection is multi-agent role decomposition and debate. Multi-agent debate (Du et al., 2024) and ChatEval (Chan et al., 2024) show that disagreement and role specialization can improve reasoning and evaluation. Surveys of LLM multi-agent systems emphasize role allocation as a major source of performance variation (Guo et al., 2024). MVSO borrows that intuition but collapses it into a single-model pipeline. Our method is therefore closer to viewpoint prompting than to full decentralized debate.

Finally, perspective-taking itself has been identified as a useful prompting primitive. Think Twice (Wilf et al., 2024) argues that explicit perspective-taking can improve theory-of-mind reasoning. We generalize that idea from social reasoning to broader agent optimization: rather than treating viewpoint changes as a task-specific trick, we treat them as a general mechanism for revising outputs with multiple latent objectives.

3 MVSO Overview

MVSO uses a single model f and a small set of viewpoint prompts $v_1, \ldots, v_k$. For an input x, the model first generates a draft $d = f(x)$. It then produces viewpoint-specific critiques

$$c_i = f(v_i, x, d), \quad i = 1, \ldots, k.$$

These critiques are combined into a synthesized revision plan

$$s = f(x, d, c_1, \ldots, c_k),$$

and the model finally rewrites the answer once:

$$y = f(x, d, s).$$

This structure is deliberately minimal. We do not use retrieval, external tools, tree search, or multiple independent models. The design goal is to test whether *viewpoint decomposition itself* adds value over a strong generic self-refinement baseline. Put differently, MVSO attempts to spend extra inference budget in a more structured way.

The choice of viewpoints is task-adaptive rather than global. For reasoning, we use perspectives such as logical validity, arithmetic checking, and final-answer formatting. For constrained rewriting, we use perspectives such as meaning preservation, exact constraint satisfaction, and fluency editing. This distinction matters because many agent tasks fail not for one reason but for several partially competing reasons.

4 MVSO-Bench

4.1 Benchmark suite

We evaluate MVSO on **MVSO-Bench**, a benchmark suite designed specifically for multi-view

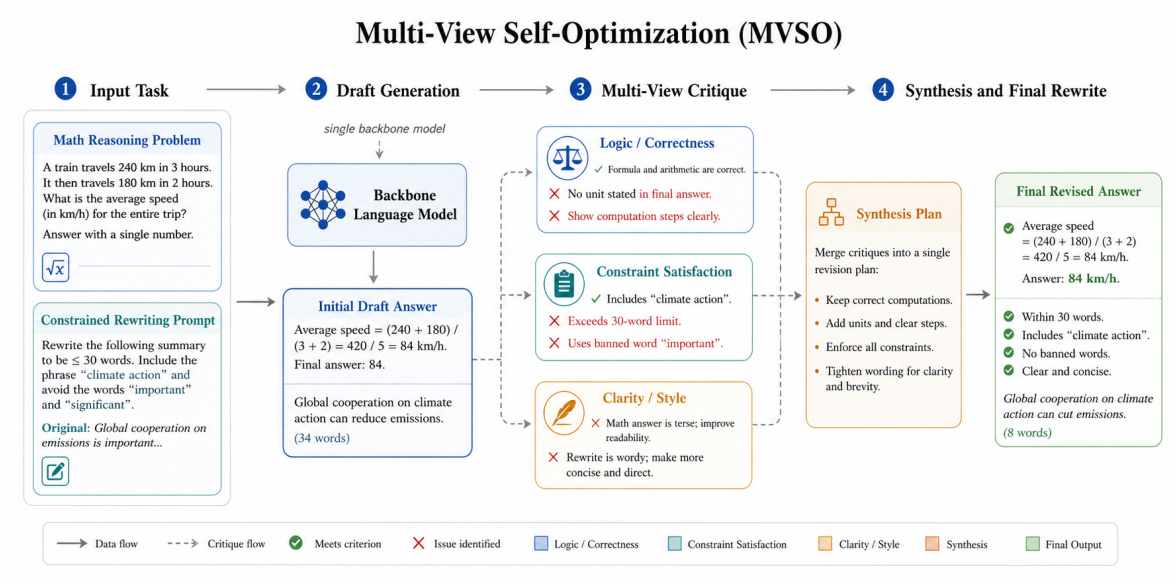

Figure 1: Overview of Multi-View Self-Optimization (MVSO). A single backbone model first produces a draft, then inspects it through several task-adaptive viewpoints, synthesizes the resulting critiques into one revision plan, and finally rewrites the answer once.

agent optimization. The first component is an easy pilot reasoning benchmark, `gsm8k_sample`, consisting of 20 sampled GSM8K test problems. The second is a 20-example controllable rewriting benchmark, `rewrite_constraints`, where the model must preserve meaning while satisfying explicit requirements such as sentence count, banned words, required phrases, and word budgets. The third is a harder reasoning suite built from public resources: `math500_hard` (80 filtered `MATH-500` problems) and `bbh_hard` (180 BBH examples across six reasoning-heavy configurations).

This benchmark design is intentionally asymmetric. The pilot tasks are not meant to be final leaderboard-style tests; they are there to expose whether multi-view prompting changes behavior on small, fully reproducible tasks. The harder subsets are where we expect backbone-sensitive differences to become more visible. In this sense, the benchmark is not just a passive evaluation set, but a targeted stress test for whether explicit viewpoint decomposition helps under different objective structures and difficulty regimes.

4.2 Metrics

For GSM8K, `MATH-500`, and BBH, we report exact match after extracting the model's final answer according to the project evaluator. For constrained rewriting, we report overall score, explicit constraint score, hard-pass rate, keyword recall, and source-overlap ROUGE-L. This mixture of metrics reflects the central motivation of the paper: some tasks are scalar and single-objective, while others expose trade-offs between correctness, faithfulness, and surface quality.

4.3 Fast validation protocol

In addition to full measured runs on the main benchmarks, we also use small validation subsets when rapid iteration is necessary. These subsets are not intended to replace the full benchmarks, but they are useful for checking whether an observed trend survives under tighter wall-clock budgets. In the current draft, we use this protocol for a short BBH validation run with reduced decoding length and a smaller number of MVSO viewpoints.

5 Experimental Setup

5.1 Models and decoding

Our primary text-first comparison uses two local checkpoints that were fully downloaded and made runnable in the workspace: `Qwen3-4B-Instruct-2507` and `Gemma-4-E2B-it`. We also retain earlier pilot runs with `Qwen3-VL-8B-Instruct` in text-only mode to stress-test the method on a stronger reasoning-capable model. Across experiments, we use greedy decoding with `temperature=0`, with `max_new_tokens` typically set between 220 and

256 for the main runs.

5.2 Methods

We compare three inference-time strategies:

- **Single-pass**: direct generation only.
- **Self-refine**: draft, generic critique, and one revision.
- **MVSO**: draft, several viewpoint critiques, one synthesis step, and one revision.

5.3 Implementation notes

All experiments run from the local workspace and use the same evaluator. Because this project evolved under real local resource constraints, we distinguish between full benchmark runs and explicit fast-validation subsets, but all numbers reported in the paper are measured rather than simulated. This makes the draft more conservative, while still allowing us to study whether MVSO behaves consistently across task difficulty regimes.

5.4 Compute and reproducibility

All runs are inference-only on local GPUs with no additional fine-tuning. The codebase, benchmark constructors, and evaluation scripts are shared in one workspace so each reported number can be traced to a concrete prediction file and summary file. For fast validation, we explicitly report reduced settings (sample count, decoding length, and MVSO view count) to avoid mixing full and reduced-budget results.

6 Main Results

6.1 Hard reasoning: MVSO helps the stronger Qwen backbone

Table 1 reports the cleanest result in the project: on `math500_hard`, MVSO yields a large improvement for Qwen but not for Gemma. This is the most persuasive current evidence that viewpoint decomposition can matter.

The Qwen result is large enough to shape the paper's central claim. MVSO does not merely add a tiny gain over self-refinement; it nearly doubles exact match relative to self-refinement and nearly triples it relative to single-pass. This indicates that for a sufficiently capable backbone, explicit perspective allocation can unlock error-correction behavior that a generic "reflect" prompt does not consistently elicit.

Model	Method	Exact Match
Qwen3-4B	Single-pass	0.1125
	Self-refine	0.1625
	MVSO	**0.3125**
Gemma-4-E2B	Single-pass	0.0000
	Self-refine	0.0125
	MVSO	0.0125

Table 1: Measured results on `math500_hard` (80 examples). MVSO substantially improves the stronger Qwen backbone, but does not help Gemma on the same benchmark.

Model	Method	GSM8K EM	Rewrite Score
Qwen3-4B	Single-pass	0.90	0.9473
	Self-refine	0.95	**0.9590**
	MVSO	**1.00**	0.9580
Gemma-4-E2B	Single-pass	0.45	0.9417
	Self-refine	**0.65**	**0.9522**
	MVSO	0.60	0.9513

Table 2: Measured pilot results on 20-example GSM8K and rewriting subsets. On easier tasks, generic self-refinement is already very competitive.

The Gemma result is just as important conceptually. MVSO is not a universal free improvement. If the backbone is too weak to produce reliable viewpoint-specific critiques, then adding more structured reflective stages does not buy much. This backbone dependence gives the paper a sharper and more honest thesis than "multi-view is always better."

6.2 Easy pilot tasks: self-refine remains a strong baseline

Table 2 summarizes easier pilot experiments. On the 20-example GSM8K sample, the best method depends on the model family: Qwen3-4B slightly favors MVSO, while Gemma slightly favors self-refine. On the rewriting benchmark, self-refine remains strongest overall, though MVSO remains competitive.

These pilot results are useful because they prevent overclaiming. If we only reported the Qwen hard-math win, the method might look uniformly superior. The pilot tasks show a more realistic picture: when tasks are easier or already well served by one generic critique, MVSO may tie or only marginally improve on self-refinement.

6.3 Fast second-hard-benchmark validation

To test whether the Qwen `math500_hard` trend extends beyond symbolic math, we also report a

Benchmark	Method	Exact Match
BBH hard subset	Single-pass	0.05
	Self-refine	**0.15**
	MVSO	0.05

Table 3: Fast measured validation on a 20-example `bbh_hard` subset with shortened decoding and two MVSO viewpoints. Unlike `math500_hard`, this second hard setting favors generic self-refinement over MVSO.

fast measured validation on a 20-example subset of `bbh_hard` using shortened decoding and two MVSO viewpoints. This subset is intentionally small, but it is useful because it probes a second reasoning family under the same single-pass, self-refine, and MVSO comparison.

Although smaller than the full benchmark, this second hard setting is important because it prevents us from overgeneralizing from the Qwen math win. Even on the stronger Qwen backbone, multi-view prompting does not automatically transfer to every hard reasoning family: under this fast BBH setting, one generic critique is stronger than decomposing the revision into two viewpoints.

Uncertainty and sample-size caution. For exact-match tasks, the corresponding success counts are: Qwen on `math500_hard` is 9/80 (single-pass), 13/80 (self-refine), and 25/80 (MVSO); the fast Qwen BBH subset is 1/20, 3/20, and 1/20 respectively. Using Wilson 95% intervals, Qwen `math500_hard` is $[0.060, 0.200]$, $[0.098, 0.258]$, and $[0.222, 0.421]$, while fast BBH is $[0.009, 0.236]$, $[0.052, 0.360]$, and $[0.009, 0.236]$. This reinforces two points: the `math500_hard` gain is sizeable, and the 20-example BBH validation should be interpreted as directional rather than final.

7 Analysis

Three conclusions stand out. First, viewpoint decomposition can be a genuinely strong intervention when the task has several separable failure modes and the backbone is capable enough to exploit the decomposition. The Qwen `math500_hard` result is too large to dismiss as noise from one additional revision step.

Second, the gains are not explained by "more inference" alone. If extra inference budget were the whole story, we would expect generic self-refinement to close most of the gap. Instead, Qwen still improves substantially when we replace one generic critique with several targeted ones. This suggests that *how the extra inference is organized* matters.

Third, the method is backbone-sensitive. Gemma's flat result on `math500_hard` implies that MVSO does not magically create missing internal competence. Multi-view prompting appears to work best when the model can already generate moderately useful specialized critiques but needs help coordinating them.

There is also a task-shape interpretation. Easy GSM8K and controlled rewriting often reward one strong correction pass. The contrast between Qwen's `math500_hard` gain and the fast BBH validation suggests that MVSO helps most when errors decompose into several partially independent checks, such as plan validity, arithmetic consistency, and final-answer formatting. In BBH-style multiple-choice reasoning, a generic revise step may already capture most of the useful correction under a short budget, while extra viewpoint prompts can diffuse attention.

8 Discussion

The most defensible claim from the current evidence is not that MVSO is universally better, but that *explicit viewpoint decomposition is a promising inference-time mechanism whose benefit is most visible on hard reasoning and depends strongly on backbone quality*. This is a sharper and more credible claim than a broad universal improvement statement.

That positioning is likely better for review because it ties the contribution to a concrete mechanism. The Qwen hard-math result serves as the main positive case, while the Gemma result clarifies an important boundary condition: multi-view prompting amplifies existing competence better than it creates missing competence from scratch. The fast BBH validation adds a second boundary condition: even for a capable backbone, viewpoint decomposition does not automatically help every hard reasoning family.

9 Potential Risks

Our method optimizes model outputs through additional critique-and-revision steps, which can improve quality but can also improve the persuasiveness of incorrect or misleading content if deployed

without safeguards. In high-stakes settings, this could increase over-trust in polished but wrong answers. We therefore view MVSO as an assistant-side quality mechanism that should be paired with task-appropriate verification, especially for factual or safety-critical domains.

There is also a benchmarking risk. Multi-stage prompting can overfit to benchmark-specific answer formats if prompts are tuned too narrowly. To reduce this risk, we evaluate across multiple task families and report both positive and negative transfer cases, rather than only selecting favorable tasks.

10 Limitations

This project is still a local text-first study rather than a large-scale conference-final campaign. Several benchmark slices are small, the second hard benchmark is currently only a 20-example fast validation rather than a full multi-method sweep, and the current backbone pool is narrower than what a final EMNLP submission should ideally report. In addition, the current multi-view prompts are manually designed rather than learned, so the paper does not yet isolate which viewpoint inventory is optimal.

Finally, MVSO is an inference-time coordination strategy rather than a training-time solution. If a backbone cannot produce useful viewpoint-specific critiques, then the framework may not help much. This means the method should be understood as a capability amplifier under certain conditions, not as a universal repair mechanism.

11 Conclusion

We presented MVSO, a simple single-backbone framework for improving language-agent outputs through explicit viewpoint switching before revision. The current study already supports one clear claim: for a sufficiently capable text model, multi-view revision can substantially outperform both single-pass generation and generic self-refinement on a hard symbolic-math benchmark. At the same time, the method is not uniformly effective across backbones or reasoning families, which sharpens rather than weakens the paper's contribution. The broader lesson is that agent self-improvement may depend less on adding undifferentiated reflective turns and more on choosing when and which perspectives the model is forced to instantiate before rewriting.

References

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and 1 others. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 17682–17690.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2024. Chateval: Towards better llm-based evaluators through multi-agent debate. In *International conference on learning representations*, volume 2024, pages 9079–9093.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first international conference on machine learning*.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*.

Dancheng Liu, Amir Nassereldine, Ziming Yang, Chenhui Xu, Yuting Hu, Jiajie Li, Utkarsh Kumar, Changjae Lee, Ruiyang Qin, Yiyu Shi, and 1 others. 2024. Large language models have intrinsic self-correction ability. *arXiv preprint arXiv:2406.15673*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652.

Alex Wilf, Sihyun Lee, Paul Pu Liang, and Louis-Philippe Morency. 2024. Think twice: Perspective-taking improves large language models' theory-of-mind capabilities. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8292–8308.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.