GUIDEDEVOLVE: FEEDBACK-GUIDED PROMPT EVOLUTION FOR DISCOVERING IMPLICIT DECISION CRITERIA

Anonymous authors
Paper under double-blind review

ABSTRACT

Reliable decision-making with Vision-Language Models (VLMs) often requires adherence to implicit, domain-specific decision policies that go beyond generic reasoning instructions. Prompt optimization provides a practical way to elicit such behavior, yet existing methods largely rely on stochastic search or scalar performance feedback, offering limited insight into why a prompt fails or how it should be improved. To address this limitation, we propose **GuidedEvolve**, a feedback-guided prompt evolution framework for decision-making that transforms case-level feedback analysis into actionable decision policies. GuidedEvolve identifies fine-grained failure patterns, distills them into transferable decision strategies, and uses these strategies to guide subsequent prompt evolution, turning prompt optimization from metric-driven exploration into interpretable and directed refinement. We evaluate GuidedEvolve on multimodal misinformation detection across public benchmark datasets. Experimental results show that GuidedEvolve consistently outperforms existing prompting and prompt-optimization methods, demonstrating the effectiveness of explicit feedback-guided evolution for complex multimodal decision-making.

1 INTRODUCTION

Social media has become a major channel for news dissemination, where textual claims are often accompanied by visual content. While such multimodal communication enriches online information exchange, it also facilitates the rapid spread of misinformation, which may arise not only from manipulated content within a single modality, but also from inconsistencies between textual claims and visual evidence (Liu et al., 2025a). The emergence of AI-generated media further exacerbates this problem by making fabricated or manipulated content increasingly realistic and difficult to distinguish from authentic information (X-Force, 2026; Xue et al., 2025). Recent advances in vision-language models (VLMs) (Google DeepMind, 2025; OpenAI, 2026; Bai et al., 2025b) provide a promising foundation for this task. In practical applications, their detection behavior is largely determined by prompts that specify how the model should inspect claims, compare cross-modal evidence, resolve conflicts, and produce final predictions. As a result, prompt design plays a crucial role in determining multimodal reasoning behavior (Yang et al., 2023). This sensitivity further motivates automatic prompt optimization for task-specific multimodal decision-making.

Despite their potential, effectively optimizing prompts for multimodal misinformation detection remains challenging. A central difficulty is that multimodal misinformation can arise through diverse forms of evidence manipulation and semantic inconsistency rather than a single, uniform failure mode. It may stem from false textual claims, manipulated visual content, or inconsistencies between the claim and its supporting context. Across these diverse forms, effective detection often relies on recurring decision principles, including verifying evidential support, checking cross-modal consistency, and handling uncertainty appropriately. Existing prompt optimization methods, however, typically optimize prompts as monolithic text strings and provide little mechanism for explicitly discovering and reusing such transferable decision strategies. Learning such decision principles is difficult because multimodal evidence is often incomplete or ambiguous. A discrepancy between text and image, for example, does not necessarily imply misinformation; the model must determine whether it reflects a genuine contradiction or merely insufficient evidence. A model may

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.

correctly recognize the content of both modalities yet still misjudge whether the observed discrepancy constitutes evidence of misinformation. It may also mistake insufficient evidence for evidence of falsehood. These errors indicate that improving misinformation detection requires more than lexical prompt refinement: the optimization process must identify why particular cases fail and extract decision rules that can guide subsequent reasoning.

Existing prompt optimization methods have primarily followed two directions. Static approaches (Kojima et al., 2022; Sun et al., 2025; Nguyen et al., 2026) manually or automatically augment prompts with fixed reasoning instructions, whereas evolutionary approaches (Fernando et al., 2024; Novikov et al., 2025) search over candidate prompts using task-level performance as feedback. However, many existing methods still rely primarily on aggregate task-level signals and provide limited mechanisms for systematically converting case-level failures into reusable decision guidance. Scalar metrics can indicate which prompt performs better overall, but they provide little insight into which types of cases fail, why those failures occur, or how the prompt should be revised. For example, a model may fail because it overlooks a subtle contradiction between a caption and an image, confuses missing evidence with falsifying evidence, or relies too heavily on superficial visual cues. Treating these heterogeneous failures merely as changes in an aggregate accuracy score makes it difficult to identify meaningful optimization directions and may reduce prompt evolution to weakly guided search.

To address this limitation, we propose GuidedEvolve, an automated prompt evolution mechanism for complex multimodal decision-making that derives actionable decision rules from case analysis and iteratively refines prompts using case-level diagnostic feedback. Unlike conventional evolutionary methods that rely solely on numerical fitness signals, GuidedEvolve introduces an error-case–analysis–optimization mechanism. Within this paradigm, the framework first conducts fine-grained analyses of individual cases—for example, attributing errors to factors such as irony or insufficient evidence—and generates targeted suggestions. Through multiple rounds of suggestion generation, the framework progressively distills transferable experience and strategic insights, which are then used to guide the creation of new prompts. In this way, prompt optimization is elevated from a purely metric-driven search process to a strategy-guided evolutionary mechanism.

GuidedEvolve retains the general evolutionary search paradigm while augmenting it with structured diagnostic feedback and distilled decision guidance. We instantiate this mechanism within an evolutionary prompt optimization framework and evaluate it on multiple public multimodal misinformation detection benchmarks covering diverse content forms, multimodal evidence patterns, and both binary and multi-class prediction settings. Extensive experiments across different VLM backbones demonstrate consistent improvements over existing prompting and prompt-evolution baselines under comparable optimization budgets.

The primary contributions of this work are summarized as follows:

- We propose GuidedEvolve, a feedback-guided prompt evolution framework that moves beyond conventional metric-driven search by incorporating structured reasoning over concrete model behaviors into the evolutionary optimization process.
- We introduce a structured diagnostic feedback mechanism that accumulates fine-grained signals from individual failure cases and leverages the resulting transferable decision principles to guide the generation of subsequent prompt candidates.
- Extensive experiments across multiple multimodal misinformation benchmarks and VLM backbones demonstrate that GuidedEvolve consistently improves over existing prompting and evolutionary optimization baselines under comparable LLM-call budgets.

2 Related Work

2.1 Multimodal Misinformation Detection

Multimodal misinformation detection aims to identify misleading content by jointly analyzing textual claims, visual evidence, and their consistency. Early approaches primarily rely on task-specific architectures for multimodal representation learning and feature fusion (Wang et al., 2018; Singhal et al., 2019), while recent large vision-language models (LVLMs) enable more flexible reasoning

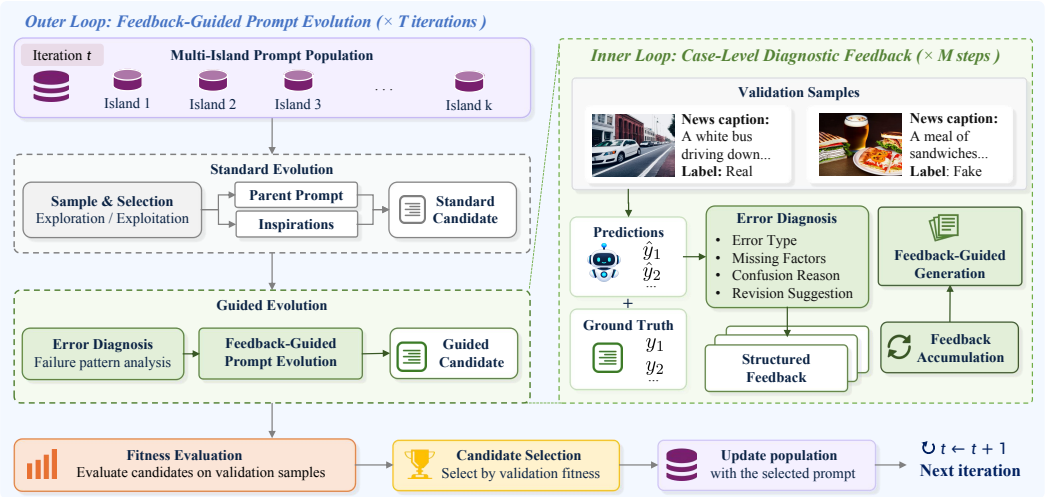


Figure 1: Overview of the GuidedEvolve framework.

over cross-modal consistency and information veracity (Cekinel et al., 2025; Hu et al., 2025). Building on these capabilities, recent research has increasingly focused on more comprehensive misinformation settings that go beyond simple modality fusion, requiring models to distinguish diverse forms of manipulation, verify cross-modal evidence, and perform structured reasoning for final judgments. MMFakeBench (Liu et al., 2025a), for example, establishes a mixed-source setting encompassing textual distortion, visual distortion, and cross-modal inconsistency, while TRUST-VL (Yan et al., 2025) jointly models different distortion types with structured reasoning chains aligned with human fact-checking workflows. These efforts reflect a broader shift toward more generalizable and reasoning-aware multimodal misinformation detection. Most existing approaches improve multimodal misinformation detection through specialized architectures, additional training data, or model adaptation. Meanwhile, relatively less attention has been paid to the decision strategies encoded in prompts. These strategies are often manually specified or optimized using only aggregate task-level feedback, leaving their automatic discovery and refinement largely underexplored.

2.2 Prompt Optimization

Prompt optimization provides a lightweight way to adapt model behavior by refining task instructions without updating model parameters. Existing approaches broadly fall into manually designed reasoning strategies and automatic prompt optimization methods. Reasoning-based prompting methods, such as Zero-shot Chain-of-Thought (Kojima et al., 2022) and Tree of Thoughts (Yao et al., 2023), introduce predefined reasoning procedures but rely on largely static templates. Automatic Prompt Optimization (APO) instead formulates prompt design as a search problem and automatically explores candidate instructions based on task feedback. Evolutionary approaches further maintain and iteratively refine a population of prompts using task performance as a fitness signal. Methods such as PromptBreeder (Fernando et al., 2024) and AlphaEvolve (Novikov et al., 2025) expand the search space through mutation and selection, but their optimization is still primarily driven by scalar rewards. Such signals indicate which prompts perform better overall, but provide limited guidance on why individual cases fail or how subsequent mutations should be directed. Recent feedback-based methods begin to incorporate richer execution-level information (Wang et al., 2024; Agrawal et al., 2026), yet systematically converting case-level failures into reusable decision guidance remains underexplored. GuidedEvolve targets this gap by using structured error diagnosis to guide subsequent prompt evolution.

3 Method

GuidedEvolve builds on evolutionary prompt optimization by coupling population-based search with case-level diagnostic feedback. The framework consists of two nested loops with complementary

Algorithm 1 GuidedEvolve Optimization Process

Require: Validation set $\mathcal{D}_{\text{val}}$, initial prompt p_0, outer iterations T, inner diagnostic steps M
Ensure: Optimized prompt p^*

Initialize prompt database $\mathcal{I} \leftarrow \{p_0\}$, organized into islands $\mathcal{I}_1, \ldots, \mathcal{I}_K$
for $t = 1$ **to** T **do**
  Outer Loop: Feedback-Guided Prompt Evolution
  Select parent prompt $p_{\text{parent}} \leftarrow \text{Sample}(\mathcal{I})$
  Initialize accumulated suggestions $\mathcal{S} \leftarrow \emptyset$
  for $m = 1$ **to** M **do**
    Inner Loop: Case-Level Diagnostic Feedback
    Sample validation cases $\mathcal{B}_m \subset \mathcal{D}_{\text{val}}$
    Obtain predictions $\hat{y}_i \leftarrow \text{Inference}(x_i, p_{\text{parent}}), \forall (x_i, y_i) \in \mathcal{B}_m$
    Construct diagnostic suggestions $\mathcal{S}_m \leftarrow \{\text{Diagnose}(x_i, y_i, \hat{y}_i, p_{\text{parent}})\}_{(x_i,y_i)\in\mathcal{B}_m}$
    Accumulate suggestions $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}_m$
  end for
  Retrieve top-performing exemplars $\mathcal{E}_{\text{top}}$ from $\mathcal{I}$
  Generate standard candidate $p_{\text{standard}} \leftarrow \text{StandardEvolve}(p_{\text{parent}}, \mathcal{E}_{\text{top}})$
  Generate guided candidate $p_{\text{guided}} \leftarrow \text{GuidedEvolve}(p_{\text{parent}}, \mathcal{S}, \mathcal{E}_{\text{top}})$
  Select better candidate $p_{\text{candidate}} \leftarrow \arg \max_{p \in \{p_{\text{standard}}, p_{\text{guided}}\}} \mathcal{F}(p, \mathcal{D}_{\text{val}})$
  Update prompt database $\mathcal{I} \leftarrow \mathcal{I} \cup \{p_{\text{candidate}}\}$ and perform island migration
end for

return $p^* \leftarrow \arg \max_{p \in \mathcal{I}} \mathcal{F}(p, \mathcal{D}_{\text{val}})$

roles: the outer loop evolves candidate prompts based on validation performance, while the inner loop analyzes model failures and accumulates structured guidance for prompt refinement. The overall framework is illustrated in Figure 1, with the detailed optimization procedure provided in Algorithm 1.

At each outer-loop iteration, GuidedEvolve samples a parent prompt from the evolving population and generates candidate prompts for evaluation. Within this iteration, the inner diagnostic loop repeatedly examines validation cases, identifies failure causes, and produces actionable revision suggestions. The accumulated feedback is then used to generate a guided candidate alongside a standard evolutionary candidate. By complementing scalar fitness with case-level diagnostic feedback, GuidedEvolve directs prompt evolution toward concrete failure modes and revision needs.

3.1 Problem Formulation

Let $\mathcal{P}$ denote the prompt space. Given a validation set $\mathcal{D}_{\text{val}} = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i is a multimodal input and y_i is the corresponding ground-truth label, our goal is to find a prompt p^* that maximizes a task-specific fitness function $\mathcal{F}$:

$$p^* = \arg \max_{p \in \mathcal{P}} \mathcal{F}(p, \mathcal{D}_{\text{val}}). \tag{1}$$

Here, $\mathcal{F}$ can be instantiated as accuracy, F1-score, balanced accuracy, or a task-specific utility metric depending on the target detection task.

GuidedEvolve maintains a prompt database $\mathcal{I}$ organized into multiple islands $\mathcal{I}_1, \ldots, \mathcal{I}_K$. Each island stores a subpopulation of prompts and their evaluation records. The multi-island structure encourages diverse search trajectories: high-performing prompts can be locally refined within each island, while periodic migration allows useful reasoning strategies to spread across islands and prevents premature convergence.

3.2 Inner Loop: Case-Level Diagnostic Feedback

The inner loop performs repeated case-level diagnosis over validation samples within each evolutionary iteration. At each inner step $m = 1, \ldots, M$, GuidedEvolve samples a validation batch

$\mathcal{B}_m \subset \mathcal{D}_{\text{val}}$ and evaluates the selected parent prompt p_{parent} on these cases. For each sample $(x_i, y_i) \in \mathcal{B}_m$, the model first obtains $\hat{y}_i = \text{Inference}(x_i, p_{\text{parent}})$. An Analyst VLM then compares the prediction $\hat{y}_i$ with the ground-truth label y_i and produces a structured diagnostic suggestion:

$$s_i = \text{Diagnose}(x_i, y_i, \hat{y}_i, p_{\text{parent}}). \tag{2}$$

Each diagnostic suggestion s_i comprises: (a) *Error Type*, which categorizes the observed failure (e.g., "cross-modal contradiction" or "insufficient evidence"); (b) *Missing Factors*, which identifies overlooked evidence or decision cues (e.g., "temporal inconsistency" or "unsupported textual claim"); (c) *Confusion Reason*, which explains why the current prompt leads to an incorrect judgment (e.g., "the model treats missing supporting evidence as direct evidence of falsity"); and (d) *Revision Suggestion*, which provides actionable guidance for refining the prompt's reasoning procedure, evidence requirements, or decision boundaries.

The suggestions produced from batch $\mathcal{B}_m$ form

$$\mathcal{S}_m = \{s_i \mid (x_i, y_i) \in \mathcal{B}_m\}. \tag{3}$$

These suggestions are accumulated across inner steps:

$$\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}_m. \tag{4}$$

After M diagnostic steps, the accumulated feedback set $\mathcal{S}$ captures diverse failure patterns and actionable revision suggestions observed across validation cases, providing structured guidance for subsequent prompt evolution.

3.3 Outer Loop: Feedback-Guided Prompt Evolution

The outer loop uses the diagnostic feedback $\mathcal{S}$ to guide evolutionary search over prompts. At each iteration, GuidedEvolve first selects a parent prompt p_{parent} from the prompt database $\mathcal{I}$. The selection strategy balances exploitation and exploration: higher-fitness prompts are more likely to be selected for refinement, while diverse prompts from different islands are also sampled to maintain search diversity. Given the selected parent prompt, GuidedEvolve generates two candidate prompts through a standard evolutionary branch and a guided evolution branch.

Standard Evolution Candidate. An LLM generator mutates the parent prompt using top-performing prompts and their fitness scores as inspiration:

$$p_{standard} = \text{StandardEvolve}(p_{parent}, \mathcal{E}_{top}), \tag{5}$$

where $\mathcal{E}_{top}$ denotes a set of high-performing prompts retrieved from the prompt database.

Guided Evolution Candidate. This branch additionally incorporates the diagnostic feedback produced by the inner loop. The generator revises the parent prompt by considering both top-performing exemplars and the structured feedback set $\mathcal{S}$:

$$p_{guided} = \text{GuidedEvolve}(p_{parent}, \mathcal{S}, \mathcal{E}_{top}). \tag{6}$$

This guided branch encourages the new prompt to address concrete failure modes observed in the current batch. Instead of merely producing stylistic or random variations, it attempts to refine the prompt's reasoning procedure, evidence requirements, and decision boundaries according to the diagnosed errors.

Both candidates are evaluated under the same fitness function:

$$p_{candidate} = \arg \max_{p \in \{p_{standard}, p_{guided}\}} \mathcal{F}(p, \mathcal{D}_{val}). \tag{7}$$

After evaluation, the selected candidate and its fitness score are added to the prompt database. Competitive candidates are retained for future sampling, while lower-performing candidates may be discarded in favor of stronger existing prompts. This population update mechanism helps preserve high-quality solutions while maintaining diversity throughout the evolutionary search.

After T iterations, GuidedEvolve returns the prompt with the highest validation-set fitness:

$$p^* = \arg \max_{p \in \mathcal{I}} \mathcal{F}(p, \mathcal{D}_{val}). \tag{8}$$

Table 1: Performance comparison of different prompt evolution methods on multimodal misinformation detection benchmarks.

Method	MMFakeBench				AMG				DeceptionDecoded			
	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.
Standard	35.64	45.73	60.55	38.48	23.38	38.58	26.85	29.18	66.11	66.05	66.73	66.05
PromptBreeder (Fernando et al., 2024)	39.37	44.72	53.05	40.80	25.38	36.94	29.21	32.11	66.23	66.13	66.73	66.13
CFPO (Liu et al., 2025b)	40.76	48.33	49.35	41.23	24.48	43.71	32.22	29.66	68.89	68.64	70.17	68.64
PromptAgent (Wang et al., 2024)	41.69	51.73	49.92	44.15	22.07	52.01	26.53	26.34	57.98	58.23	59.93	58.23
Promptolution (Zehle et al., 2026)	44.75	47.76	50.99	47.99	25.38	40.80	29.22	30.98	66.49	66.37	67.05	66.37
GEPA (Agrawal et al., 2026)	53.41	55.96	57.07	54.65	26.10	40.53	28.72	29.73	74.12	74.20	74.53	74.21
ThetaEvolve (Wang et al., 2026)	39.97	41.10	44.82	40.75	25.16	40.22	27.71	29.53	67.53	67.50	68.32	67.50
ShinkaEvolve (Lange et al., 2026)	47.17	49.21	48.91	49.15	23.21	39.09	25.19	26.03	69.82	69.73	70.50	69.73
AlphaEvolve (Novikov et al., 2025)	51.74	55.48	55.59	51.74	24.14	42.81	28.29	28.64	70.53	70.58	71.52	70.58
GuidedEvolve (Ours)	**58.15**	**59.58**	**61.66**	**57.84**	**26.94**	**48.12**	**29.85**	**30.17**	**75.02**	**75.22**	**75.17**	**75.22**

4 EXPERIMENT

4.1 EXPERIMENTAL SETUP

4.1.1 DATASETS

To comprehensively evaluate the effectiveness of feedback-guided prompt evolution for multimodal misinformation detection, we conduct experiments on three representative benchmarks: MMFakeBench (Liu et al., 2025a), AMG (Guo et al., 2025), and DeceptionDecoded (Wu et al., 2026). MMFakeBench (Liu et al., 2025a) is a large-scale multimodal benchmark consisting of 11K image–text pairs collected from diverse sources, including Politifact, GossipCop, Fakeddit, and AI-generated content. It covers fine-grained misinformation patterns involving textual distortion, visual manipulation, and cross-modal inconsistency. AMG (Guo et al., 2025) focuses on fine-grained attribution of multimodal misinformation, distinguishing multiple types of image–text inconsistency and manipulation beyond binary fake-news detection. For DeceptionDecoded (Wu et al., 2026), we use the subset corresponding to the source task, which focuses on identifying whether multimodal news content is misleading based on the relationship between the presented content and its underlying source. Together, these benchmarks cover diverse misinformation patterns and provide a comprehensive evaluation of multimodal misinformation detection.

4.1.2 BASELINES

We compare *GuidedEvolve* with representative prompt optimization and evolutionary search methods. *PromptBreeder* (Fernando et al., 2024) formulates prompt optimization as a self-referential evolutionary process, jointly evolving task prompts and the mutation prompts used to improve them. *PromptAgent* (Wang et al., 2024) casts prompt optimization as a strategic planning problem and searches for improved prompts through Monte Carlo Tree Search guided by model errors and feedback. *CFPO* (Liu et al., 2025b) iteratively optimizes both prompt content and structural form, exploring complementary content-level and format-level modifications. *Promptolution* (Zehle et al., 2026) is a unified modular framework that integrates multiple discrete prompt optimization methods. In our experiments, we instantiate its CAPO optimizer, which performs evolutionary prompt optimization with cost-aware racing to efficiently allocate evaluations. *ShinkaEvolve* (Lange et al., 2026) performs sample-efficient evolution with LLM-based mutation and population-based selection, maintaining high-quality and diverse candidates through evolutionary archives; we adapt its search procedure from program evolution to prompt evolution. *GEPA* (Agrawal et al., 2026) performs reflective prompt evolution using natural-language feedback and Pareto-aware selection to iteratively improve candidate prompts. *ThetaEvolve* (Wang et al., 2026) extends evolutionary search into a test-time learning framework, combining inference-time candidate evolution with reinforcement-learning-based adaptation of the proposal model. *AlphaEvolve* (Novikov et al., 2025) combines LLM-based proposal generation with evolutionary search and iterative evaluation to progressively discover higher-performing solutions. In contrast to these general-purpose optimization methods, *GuidedEvolve* augments evolutionary search with structured diagnostic feedback, lever-

Table 2: Comparison of total LLM-call budgets for different prompt optimization methods. Δ denotes the relative difference in total calls compared with GuidedEvolve on the same benchmark.

Method	MMFakeBench		AMG		DeceptionDecoded	
	Total	Δ	Total	Δ	Total	Δ
PromptBreeder (Fernando et al., 2024)	203,127	-0.04%	105,078	-0.98%	203,127	-0.04%
CFPO (Liu et al., 2025b)	200,199	-1.48%	103,599	-2.37%	200,199	-1.48%
PromptAgent (Wang et al., 2024)	203,200	0.00%	106,117	0.00%	203,200	0.00%
Promptolution (Zehle et al., 2026)	203,396	$+0.10\%$	106,178	$+0.06\%$	203,312	$+0.06\%$
GEPA (Agrawal et al., 2026)	203,300	$+0.05\%$	105,992	-0.12%	203,193	0.00%
ThetaEvolve (Wang et al., 2026)	201,200	-0.98%	104,117	-1.88%	201,200	-0.98%
ShinkaEvolve (Lange et al., 2026)	203,202	0.00%	106,189	$+0.07\%$	203,202	0.00%
AlphaEvolve (Novikov et al., 2025)	203,202	0.00%	105,153	-0.91%	203,202	0.00%
GuidedEvolve (Ours)	**203,200**	–	**106,117**	–	**203,200**	–

aging case-level failure analysis to direct prompt refinement toward more effective task-specific decision strategies.

4.1.3 Implementation Details

For MMFakeBench and AMG, prompt optimization uses the official validation data, while the official test sets are reserved for final evaluation. For the DeceptionDecoded source task, we construct a 1,000-example validation set by balanced sampling and a separate non-overlapping 1,000-example test set. We adopt Qwen3-VL-8B as the default base vision-language model (VLM) for all experiments, as it represents a strong and efficient modern VLM. Unless stated otherwise, the decoding parameters were set to a temperature of 0 and a top-p value of 1. To facilitate reliable evaluation, we constrain model outputs using predefined schemas (e.g., JSON format), which improves consistency and enables automatic parsing of results. To ensure a fair comparison, we approximately match the total LLM-call budgets across prompt optimization methods. Despite different internal call allocations, all methods use comparable overall budgets across datasets, with deviations from GuidedEvolve within 2.4%.

4.2 Main Results

As summarized in Table 1, GuidedEvolve achieves the highest F1 score across all three benchmarks. It obtains F1 scores of 58.15, 26.94, and 75.02 on MMFakeBench, AMG, and DeceptionDecoded, respectively. Compared with the strongest competing baseline, GEPA, these results correspond to improvements of 4.74, 0.84, and 0.90 points. Relative to AlphaEvolve, the evolutionary framework most closely related to our implementation, the corresponding gains are 6.41, 2.80, and 4.49 points. GuidedEvolve also achieves the best accuracy, precision, and recall on MMFakeBench and DeceptionDecoded. On AMG, it achieves the highest F1, although the best accuracy, precision, and recall are obtained by other baselines. Overall, these results show that incorporating structured case-level feedback consistently improves prompt evolution across heterogeneous multimodal misinformation benchmarks. Table 1 reports mean results over three random seeds, with the corresponding standard deviations provided in Appendix A.

4.3 Computational Budget

To ensure a fair comparison, we control the total number of LLM calls used by each prompt optimization method. Since different methods distribute computation differently across evaluation, prompt generation, feedback construction, and diagnostic analysis, we compare them under closely matched overall LLM-call budgets rather than enforcing identical internal allocations.

Table 2 reports the total LLM calls and their relative differences. Across all three benchmarks, GuidedEvolve operates under a computational budget closely matched to those of the compared baselines, with the deviation in total LLM calls remaining within 2.4%. Although GuidedEvolve introduces additional calls for case-level diagnosis and feedback-guided evolution, this overhead is

Table 3: Ablation study of GuidedEvolve on multimodal misinformation detection benchmarks.

Method	MMFakeBench				AMG				DeceptionDecoded			
	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.
w/o Guided Candidate	51.74	55.48	55.59	51.74	24.14	42.81	28.29	28.64	70.53	70.58	71.52	70.58
w/o Diagnostic Feedback	45.61	46.37	50.60	47.94	23.92	44.10	25.34	30.91	68.75	68.94	70.59	68.96
GuidedEvolve (Ours)	**58.15**	**59.58**	**61.66**	**57.84**	**26.94**	**48.12**	**29.85**	**30.17**	**75.02**	**75.22**	**75.17**	**75.22**

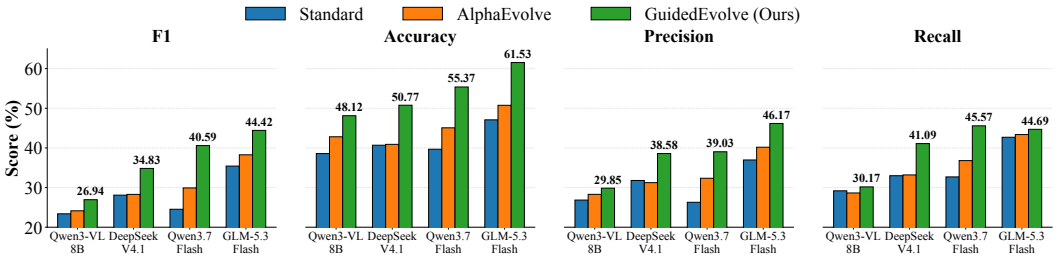


Figure 2: Performance comparison of Standard prompting, AlphaEvolve, and GuidedEvolve across different VLMs on AMG.

offset by reducing the number of evolutionary iterations. These closely matched budgets indicate that the performance gains of GuidedEvolve are unlikely to result simply from using more model queries.

4.4 Ablation Study

We further examine the contributions of the guided candidate branch and diagnostic feedback in GuidedEvolve. We compare the full method with two variants: *w/o Guided Candidate*, which removes the guided branch and reduces the framework to AlphaEvolve-style evolution, and *w/o Diagnostic Feedback*, which retains guided candidate generation but bypasses the diagnosis and directly generates a candidate prompt. As shown in Table 3, removing either component degrades performance. Notably, *w/o Diagnostic Feedback* achieves lower F1 than AlphaEvolve on all three benchmarks, indicating that simply introducing an additional unguided prompt revision is less effective than standard evolutionary search. In contrast, the full GuidedEvolve consistently performs best, suggesting that its gains stem from using case-level diagnostic feedback to direct candidate generation rather than from additional prompt generation alone.

4.5 Comparison with Various VLMs

To assess whether the effectiveness of GuidedEvolve transfers across different VLM backbones, we evaluate the proposed method across different vision-language models (VLMs). Specifically, we apply GuidedEvolve to four representative models spanning multiple model families, including Qwen3-VL-8B (Bai et al., 2025a), Qwen3.7-Flash (Alibaba Cloud, 2026), DeepSeek-V4.1-Flash (DeepSeek-AI, 2026), and GLM-5.3-Flash (GLM-5 Team et al., 2026). Figure 2 reports the results in terms of F1, Accuracy, Precision, and Recall. Across the tested models, GuidedEvolve consistently outperforms both the *Standard* prompting baseline and AlphaEvolve, showing consistent gains across VLM backbones. These results indicate that GuidedEvolve's effectiveness is not specific to a particular base model but transfers across models with different capability profiles, demonstrating that the gains of GuidedEvolve are not specific to a single VLM backbone.

4.6 Convergence Analysis

To further investigate the optimization behavior of GuidedEvolve, we visualize the best validation F1 score achieved throughout the evolutionary process and compare it with AlphaEvolve. As shown in Figure 3, GuidedEvolve achieves consistently higher best-so-far validation performance across the evolutionary process. On MMFakeBench, GuidedEvolve improves rapidly during the

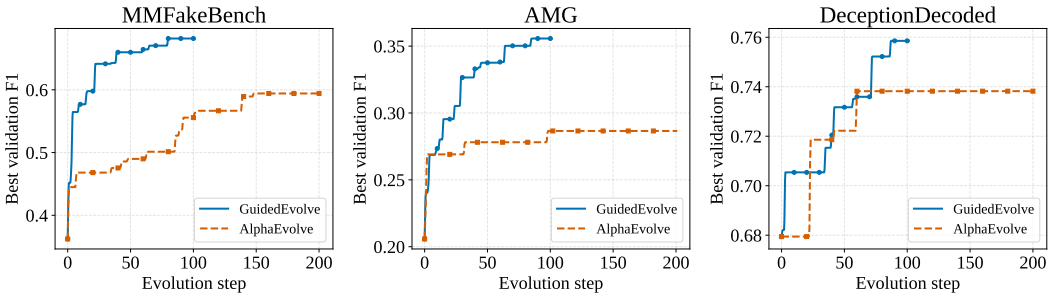


Figure 3: Convergence comparison between GuidedEvolve and AlphaEvolve on MMFakeBench, AMG, and DeceptionDecoded. Each curve shows the best validation F1 achieved up to each evolution step. GuidedEvolve consistently converges to higher-performing prompts across all three benchmarks.

early evolution stages and continues to obtain substantial gains before converging to a markedly higher validation F1 than AlphaEvolve. A similar pattern is observed on AMG, where GuidedEvolve steadily improves throughout optimization, whereas AlphaEvolve reaches an earlier plateau with only limited subsequent gains. On DeceptionDecoded, both methods improve during the initial stages, but GuidedEvolve continues to discover stronger prompts and ultimately converges to a higher validation performance.

Overall, GuidedEvolve not only reaches a better final solution but also exhibits a more effective optimization trajectory across different datasets. In particular, the diagnostic feedback provides explicit directions for prompt refinement, allowing the evolutionary process to progressively address observed failure patterns rather than relying primarily on performance-driven exploration. GuidedEvolve uses fewer evolutionary iterations than AlphaEvolve because each iteration additionally generates and evaluates a guided candidate alongside the standard evolutionary candidate, while the two methods are evaluated under closely matched total LLM-call budgets.

5 Conclusion

In this paper, we introduce GuidedEvolve, a feedback-guided prompt evolution framework that reconceptualizes prompt optimization from a heuristic search process toward a more strategically guided optimization process. By automating the distillation of implicit decision rules from error patterns, GuidedEvolve transforms case-level failures into structured and actionable guidance for prompt refinement. Experiments across three multimodal misinformation benchmarks demonstrate that GuidedEvolve consistently improves over standard prompting and existing prompt optimization methods under comparable LLM-call budgets. Additional evaluations across different VLM backbones further show that the effectiveness of GuidedEvolve is not limited to a single base model. Overall, our results demonstrate the value of incorporating structured case-level diagnostic feedback into evolutionary prompt optimization. By using concrete model failures to provide explicit directions for prompt refinement, GuidedEvolve offers a simple and modular approach for optimizing prompts in multimodal decision-making tasks that require nuanced, task-specific decision criteria.

6 Discussion

Beyond benchmark evaluation, GuidedEvolve is also well suited to large-scale data quality control in industrial foundation-model development. Recurring quality failures can be distilled into reusable control instructions, reducing repeated manual prompt engineering and highlighting the potential of feedback-guided prompt evolution in practical model-development workflows. Future work may investigate more reliable feedback aggregation, tighter integration with agent-based data curation and quality-control pipelines, and broader applications to complex decision-making tasks.

AI USE STATEMENT

Generative AI tools were used in two distinct capacities in this work. First, LLMs and VLMs constitute part of the proposed methodology and experimental evaluation, including prompt generation, diagnostic feedback construction, prompt evolution, and multimodal inference. Their roles, model configurations, and evaluation protocols are described in the corresponding methodological and experimental sections. Second, generative AI tools were used during manuscript preparation for limited assistance with language polishing, wording refinement, code debugging, and LaTeX formatting. All research ideas, experimental designs, implementations, analyses, interpretations, and scientific claims were developed, reviewed, and verified by the authors. The authors take full responsibility for the accuracy and integrity of the manuscript.

ETHICS STATEMENT

This work focuses on improving prompt optimization for multimodal misinformation detection using existing public benchmark datasets. We do not collect new personal data, conduct human-subject experiments, or introduce additional user profiling beyond the information already contained in the benchmark datasets. We follow the original dataset licenses and intended research-use settings.

Multimodal misinformation detection is inherently sensitive to ambiguity, cultural context, incomplete evidence, and model bias. Automated systems may produce both false positives and false negatives, potentially misclassifying legitimate content or failing to identify misleading information. The proposed method is therefore intended as a research framework for improving and studying VLM decision behavior rather than as a standalone mechanism for high-stakes content moderation or fact-checking. Human oversight and additional domain-specific verification remain necessary for real-world deployment.

In addition, GuidedEvolve relies on feedback generated by large models, which may inherit biases or systematic errors from the underlying models. Although the feedback-guided optimization process can improve task performance, it does not eliminate such limitations. We therefore encourage future work to evaluate fairness, robustness, and cross-domain behavior before applying the framework in sensitive real-world settings.

REPRODUCIBILITY STATEMENT

We provide detailed experimental configurations to facilitate reproduction of the reported results. To account for stochasticity in prompt optimization and evolutionary search, all main experiments are repeated using fixed seeds, and the corresponding mean and standard deviation results are reported in Appendix A. We additionally control the optimization cost by approximately matching the total number of LLM calls across prompt optimization methods. Upon publication, we will release the implementation, prompts, configuration files, and evaluation scripts to facilitate further reproduction and extension of our results.

REFERENCES

LA Agrawal, S Tan, D Soylu, N Ziems, R Khare, K Opsahl-Ong, et al. Gepa: reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations*, 2026.

Alibaba Cloud. Qwen3.7-flash model information. Alibaba Cloud Model Studio Documentation, 2026. Accessed September 2026.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-vl technical report. *ArXiv preprint*, abs/2511.21631, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, et al. Qwen2.5-vl technical report. *CoRR*, abs/2502.13923, 2025b.

Recep Firat Cekinel, Pinar Karagoz, and Çağrı Çöltekin. Multimodal fact-checking with vision language models: A probing classifier based solution with embedding strategies. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 4622–4633. ACL, 2025.

DeepSeek-AI. Deepseek-v4.1-flash: Pushing the limits of kv cache compression, 2026.

Chrisantha Fernando, Dylan Sunil Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: Self-referential self-improvement via prompt evolution. In *Proceedings of the International Conference on Machine Learning*, volume 235, pp. 13481–13544, 2024.

GLM-5 Team et al. Glm-5: From vibe coding to agentic engineering, 2026.

Google DeepMind. Frontier safety framework report: Gemini 3 pro. Technical report, Google DeepMind, November 2025. URL https://storage.googleapis.com/deepmind-media/gemini/gemini_3_pro_fsf_report.pdf.

Hao Guo, Zihan Ma, Zhi Zeng, Minnan Luo, Weixin Zeng, Jiuyang Tang, and Xiang Zhao. Each fake news is fake in its own way: An attribution multi-granularity benchmark for multimodal fake news detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 228–236, 2025.

Shuguo Hu, Jun Hu, and Huaiwen Zhang. Synergizing LLMs with global label propagation for multimodal fake news detection. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 1426–1440. ACL, 2025.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Annual Conference on Neural Information Processing Systems*, 2022.

Robert Tjarko Lange, Yuki Imajuku, and Edoardo Cetin. Shinkaevolve: Towards open-ended and sample-efficient program evolution. In *International Conference on Learning Representations*, 2026.

Xuannan Liu, Zekun Li, Peipei Li, Huaibo Huang, Shuhan Xia, Xing Cui, Linzhi Huang, Weihong Deng, and Zhaofeng He. Mmfakebench: A mixed-source multimodal misinformation detection benchmark for lvlms. In *International Conference on Learning Representations*, 2025a.

Yuanye Liu, Jiahang Xu, Li Lyna Zhang, Qi Chen, Xuan Feng, Yang Chen, Zhongxin Guo, Yuqing Yang, and Peng Cheng. Beyond prompt content: Enhancing LLM performance via content-format integrated prompt optimization. *CoRR*, abs/2502.04295, 2025b.

Tin Nguyen, Logan Bolton, Mohammad Reza Taesiri, Trung Bui, and Anh Totti Nguyen. HoT: Highlighted chain of thought for referencing supporting facts from inputs. *Transactions on Machine Learning Research*, 2026.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. *ArXiv preprint*, abs/2506.13131, 2025.

OpenAI. Openai GPT-5 system card. *CoRR*, abs/2601.03267, 2026.

Shivangi Singhal, Rajiv Ratn Shah, Tanmoy Chakraborty, Ponnurangam Kumaraguru, and Shin'ichi Satoh. Spotfake: A multi-modal framework for fake news detection. In *IEEE Fifth International Conference on Multimedia Big Data*, pp. 39–47. IEEE, 2019.

Yiliu Sun, Yanfang Zhang, Zicheng Zhao, Sheng Wan, Dacheng Tao, and Chen Gong. Fast-Slow-Thinking: Complex Task Solving with Large Language Models. *ArXiv preprint*, abs/2504.08690, 2025.

Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric P. Xing, and Zhiting Hu. Promptagent: Strategic planning with language models enables expert-level prompt optimization. In *International Conference on Learning Representations*, 2024.

Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. Eann: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 849–857. ACM, 2018.

Yiping Wang, Shao-Rong Su, Zhiyuan Zeng, Eva Xu, Liliang Ren, Xinyu Yang, Zeyi Huang, Xuehai He, Luyao Ma, Baolin Peng, et al. Thetaevolve: Test-time learning on open problems. In *Proceedings of the International Conference on Machine Learning*, 2026.

Jiaying Wu, Fanxiao Li, Zihang Fu, Min-Yen Kan, and Bryan Hooi. Seeing through deception: Uncovering misleading creator intent in multimodal news with vision-language models. In *International Conference on Learning Representations*, 2026.

IBM X-Force. Ibm 2026 x-force threat index: Ai-driven attacks are escalating. *IBM Newsroom*, 2026.

Qi Xue, Minrui Jiang, Runjia Zhang, Xiurui Xie, Pei Ke, and Guisong Liu. Falcon: A cross-modal evaluation dataset for comprehensive safety perception. *CoRR*, abs/2509.23783, 2025.

Zehong Yan, Peng Qi, Wynne Hsu, and Mong-Li Lee. Trust-vl: An explainable news assistant for general multimodal misinformation detection. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, 2025.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. MM-REACT: prompting chatgpt for multimodal reasoning and action. *CoRR*, abs/2303.11381, 2023.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Annual Conference on Neural Information Processing Systems*, 36:11809–11822, 2023.

Tom Zehle, Timo Heiß, Moritz Schlager, Matthias Aßenmacher, and Matthias Feurer. promptolution: A unified, modular framework for prompt optimization. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pp. 282–296, 2026.

Table 4: Performance comparison of different prompt evolution methods across three independent runs with random seeds 42, 43, and 44. Results are reported as mean ± standard deviation.

Method	MMFakeBench				AMG				DeceptionDecoded			
	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.
Standard	35.64 ± 0.04	45.73 ± 0.04	60.55 ± 0.22	38.48 ± 0.03	23.38 ± 0.49	38.58 ± 0.39	26.85 ± 0.39	29.18 ± 0.86	66.11 ± 0.12	66.05 ± 0.11	66.73 ± 0.14	66.05 ± 0.11
PromptBreeder Fernando et al. (2024)	39.37 ± 3.05	44.72 ± 0.72	53.05 ± 4.90	40.80 ± 2.07	25.38 ± 0.64	36.94 ± 4.05	29.21 ± 0.83	32.11 ± 0.81	66.23 ± 0.31	66.13 ± 0.31	66.73 ± 0.26	66.13 ± 0.31
CFPO Liu et al. (2025b)	40.76 ± 0.05	48.33 ± 0.03	49.35 ± 0.08	41.23 ± 0.03	24.48 ± 0.57	43.71 ± 2.02	32.22 ± 5.52	29.66 ± 1.24	68.89 ± 1.07	68.64 ± 1.05	70.17 ± 1.16	68.64 ± 1.05
PromptAgent Wang et al. (2024)	41.69 ± 2.11	51.73 ± 1.32	49.92 ± 4.53	44.15 ± 1.43	22.07 ± 2.02	52.01 ± 4.15	26.53 ± 5.24	26.34 ± 1.54	57.98 ± 9.19	58.23 ± 8.82	59.93 ± 10.29	58.23 ± 8.79
Promptolution Zehle et al. (2026)	44.75 ± 0.57	47.76 ± 1.86	50.99 ± 4.13	47.99 ± 3.37	25.38 ± 0.79	40.80 ± 0.41	29.22 ± 1.07	30.98 ± 0.51	66.49 ± 0.32	66.37 ± 0.32	67.05 ± 0.26	66.37 ± 0.32
GEPA Agrawal et al. (2026)	53.41 ± 1.47	55.96 ± 2.12	57.07 ± 3.20	54.65 ± 0.37	26.10 ± 2.17	40.53 ± 2.42	28.72 ± 2.11	29.73 ± 2.52	74.12 ± 1.93	74.20 ± 1.73	74.53 ± 1.75	74.21 ± 1.73
ThetaEvolve Wang et al. (2026)	39.97 ± 1.64	41.10 ± 2.22	44.82 ± 2.67	40.75 ± 2.96	25.16 ± 0.67	40.22 ± 3.65	27.71 ± 1.79	29.53 ± 1.89	67.53 ± 4.14	67.50 ± 3.92	68.32 ± 4.28	67.50 ± 3.91
ShinkaEvolve Lange et al. (2026)	47.17 ± 1.61	49.21 ± 1.68	48.91 ± 2.49	49.15 ± 1.11	23.21 ± 2.86	39.09 ± 5.81	25.19 ± 2.20	26.03 ± 3.79	69.82 ± 2.02	69.73 ± 2.17	70.50 ± 1.80	69.73 ± 2.18
AlphaEvolve Novikov et al. (2025)	51.74 ± 2.58	55.48 ± 1.78	55.59 ± 2.08	51.74 ± 2.24	24.14 ± 1.04	42.81 ± 3.42	28.29 ± 2.15	28.64 ± 1.16	70.53 ± 0.78	70.58 ± 0.78	71.52 ± 0.38	70.58 ± 0.78
GuidedEvolve (Ours)	58.15 ± 4.29	59.58 ± 3.12	61.66 ± 2.79	57.84 ± 4.47	26.94 ± 1.25	48.12 ± 3.05	29.85 ± 1.41	30.17 ± 0.92	75.02 ± 0.39	75.22 ± 0.20	75.17 ± 0.36	75.22 ± 0.21

A Repeated-Run Results

To account for the stochasticity of prompt optimization and evolutionary search, we repeat all experiments three times using random seeds 42, 43, and 44. Table 4 reports the mean and standard deviation across the three runs. Overall, GuidedEvolve achieves the highest average F1 score across all three misinformation benchmarks. On MMFakeBench, it obtains an F1 score of 58.15 ± 4.29, outperforming the strongest baseline, GEPA (53.41 ± 1.47). On AMG, GuidedEvolve achieves 26.94 ± 1.25 F1, compared with 26.10 ± 2.17 for GEPA. On DeceptionDecoded, it reaches 75.02 ± 0.39 F1, exceeding GEPA (74.12 ± 1.93). These repeated-run results indicate that the average improvements of GuidedEvolve are preserved across different random seeds, although the magnitude of run-to-run variation differs across datasets.