Agentic Prompt Optimization with Visual Contrastive Reasoning for Fine-Grained Classification

Lucas Goncalves Vidit Bansal Karim Bouyarmane Robert A. Barton
Amazon, USA
{sglucas, bansalv, bouykari, rab}@amazon.com

Abstract

*Fine-grained visual recognition demands attention to subtle, localized differences that current multimodal large language models (MLLMs) often overlook when guided by generic prompts. We propose **APO-Pair**, a prompt-optimization framework that learns classification rules by contrasting image pairs. A multimodal agent views these pairs, judges whether they depict the same fine-grained class, and iteratively updates a single verification prompt using structured, interpretable YAML-style patches. Each prompt is treated as an arm in a Thompson-sampling bandit whose Beta posterior is updated from pairwise successes and failures, balancing exploration with exploitation without touching model weights. Experiments on seven challenging 3-class fine-grained splits derived from CUB-200, iNaturalist, and VegFru, APO-Pair improves top-1 accuracy over prompt-optimization baselines by up to 17 percentage points. These results suggest that pairwise visual contrast is a useful signal for prompt optimization in controlled fine-grained settings.*

1. Introduction

Multimodal large language models (MLLMs) have demonstrated impressive image classification capabilities on broad datasets [1]. Models such as Flamingo [4] and Qwen-VL [31] can recognize generic object categories by leveraging both vision and language understanding. However, when these models are prompted on fine-grained classification tasks that require distinguishing subtle visual differences between similar subcategories, they often confuse similar classes [16, 39]. For example, an MLLM may confuse bird species or plant varieties that share visual traits, as it tends to overlook minute details without explicit guidance. Closing this gap is critical; fine-grained recognition of species, consumer products, or medical pathology relies on subtle distinctions where generic MLLMs currently fall short [37]. A conventional solution is to further fine-tune vision-language models on large, labeled fine-grained datasets [15, 19, 41]. Indeed, prior efforts have shown that carefully supervised training can improve fine-grained accuracy [15]. Unfortunately, this approach is highly resource-intensive: collecting and annotating fine-grained data is expensive and time-consuming, due to the subtle distinctions between classes [13, 34, 39]. For instance, labeling thousands of bird images by species requires expert knowledge [30] and yields only incremental improvements. Moreover, fine-tuning massive MLLMs (often with billions of parameters) is computationally costly or even infeasible when using proprietary API-based models. A recent study analyzed MLLMs' fine-grained abilities and introduced additional training to boost their recognition power, but at the cost of large-scale supervision [13]. This motivates exploring alternative, lightweight strategies that do not rely on new labeled data or model fine-tuning.

One promising direction is to optimize prompts for frozen MLLMs without requiring model fine-tuning or large-scale data collection [14, 28]. Even with limited labeled examples from the target domain, prompt optimization can provide valuable signal about which visual features are important for discrimination. However, effectively optimizing prompts in a black-box setting (with no model gradients or internal training) is non-trivial. A recent work demonstrated that MLLMs can improve themselves by iterative prompt refinement [16]. This work introduces an intermediate image description step: the MLLM is first prompted to generate a detailed textual description of an image's visual attributes and then classifies the image using that description as additional context. By optimizing the description prompt through self-supervised feedback, this approach highlights the promise of prompt-based self-improvement: with the right prompting strategy, even a fixed model like GPT [27] or Qwen-VL [31] can be steered to focus on discriminative features it previously ignored.

Yet even with a well-crafted description prompt, the model still reasons about each image in isolation. This becomes brittle when subclasses live inside a visually homogeneous parent class—e.g., *Arctic lupine*, *Silvery lupine*,

and *Arizona lupine* differ only in minute petal venation and foliage tint [29]. In such cases the decisive evidence is not an absolute attribute ("this leaf is green") but a relative one ("leaf A is bluer-green than leaf B"). Motivated by this nuance, we introduce APO-Pair—an agentic prompt-optimization loop. Overall, our approach recast prompt-optimization for fine-grained recognition as a pairwise contrastive visual reasoning problem: instead of asking the MLLM to label a single image or describe it separately, we ask it to contrast and visually reason between two images and determine whether belong to the same sub-category. This contrastive framing obliges the model to align and compare fine details, sharpening its sensitivity to subtle cues that generic prompts overlook.

APO-Pair (**A**gentic **P**rompt **O**ptimization with **Pair**wise Contrastive Reasoning) refines the prompt through a multimodal LLM agent that interacts in a closed loop with a frozen vision model. At each iteration the current prompt is scored on a batch of image pairs, producing successes and failures. A balanced subset of these outcomes, together with the images and textual rationales, is given to the agent. The agent proposes targeted YAML-style edits (line additions, modifications, or deletions); an edit is accepted only if it improves the pairwise performance, after which the new prompt joins a repository of candidates. This structured approach enables targeted modifications and maintains an interpretable audit trail. Beyond academic reproducibility, this lineage tracking is critical for deploying dynamic prompts in production environments, where transparency and the ability to roll back changes are essential. To traverse the prompt space efficiently, APO-Pair treats each prompt as a Thompson-sampling bandit [3, 6] arm with a Beta posterior over its success rate. Every cycle samples these posteriors, selects a promising prompt, re-evaluates it, and allows the agent to refine it. This uncertainty-aware exploration guards against local optima by keeping multiple prompts active. A rejection log of failed edits and a list of image pairs prevent repetitions and forgetting of previous failed attempts, since the agent learns not to repeat unhelpful changes. Through multiple ablation studies, we demonstrate that each of these components is beneficial to fine-grained classification performance. Our contributions are as follows:

1. **Contrastive visual reasoning for prompt optimization.** We introduce multimodal prompt-optimization for fine-grained recognition in which an agent examines image pairs and reasons contrastively over successes and failures.
2. **Bayesian score-based prompt scheduling.** We model prompt selection as a Thompson-sampling bandit whose Beta posteriors are updated from pairwise prediction outcomes. Providing uncertainty-aware balance between exploration and exploitation over prompt trajectory.
3. **Interpretable YAML edits and lineage tracking.** We represent prompt updates as targeted YAML additions, modifications, and deletions recorded in a patch log. This structured evolution process supports auditability, ablation, and rollback, while rejection memory discourages repeated unsuccessful edits and encourages more diverse prompt exploration.

2. Related Work

Fine-Grained Recognition with MLLMs: The difficulty of fine-grained image classification for multimodal models is well documented [19, 22, 36]. Previous evaluation of large vision-language models (VLM) on fine-grained categories found significant performance gaps compared to general object recognition [10], as MLLMs trained on web-scale data often lack specialized knowledge for expert-level discrimination. Some works address this through domain-specific prompts or fine-tuning [13, 20], but these supervised approaches are impractical when labeling data or retraining models is expensive.

Our work aligns with recent efforts to improve fine-grained classification in extremely similar visual classes. The AutoSEP framework [16] uses images to self-optimize prompts that elicit more discriminative descriptions from MLLMs, conceptually related to chain-of-thought prompting [33]. We instead use pairwise comparison as the auxiliary task, analogous to contrastive learning [7, 23], where the model learns by comparing similar versus dissimilar pairs. By prompting an MLLM to decide if two images are the same, we guide it to identify discriminative features.

Automatic Prompt Optimization (APO): There is growing interest in algorithms that automatically optimize prompts for large language models [35]. Early approaches treated this as discrete search over token space using gradients or reinforcement learning [8, 25], requiring white-box access and labeled data. Recent black-box methods harness LLMs' own capabilities: some use LLMs to analyze errors and suggest revisions [40], while others adopt evolutionary paradigms where populations of prompts undergo mutation and selection [9, 12]. Building on this, recent frameworks like GEPA demonstrate that reflective prompt evolution can even outperform traditional reinforcement learning [2]. This underscores a shift toward continuous learning loops driven by iterative self-reflection, bypassing the heavy overhead of RL. Collectively, these evolutionary approaches demonstrate that iterative prompt optimization can significantly boost NLP performance without human intervention.

Several contemporaneous works have introduced more structured or strategic ways for LLMs to act as prompt engineers. PromptAgent views prompt optimization as a planning problem and uses Monte Carlo Tree Search (MCTS) guided by an LLM's critique of intermediate

prompts [32]. This strategic exploration enabled finding expert-level prompts on challenging language tasks beyond what greedy improvements achieved. CriSPO takes a multi-aspect critique-and-suggest approach: the LLM evaluates a candidate prompt along different dimensions (e.g. correctness, clarity, completeness) and then proposes focused edits for each aspect, which are aggregated into a refined prompt [14]. By iterating this critique-suggestion loop, CriSPO improved prompt quality for text generation tasks over single-step baselines. Our APO-Pair framework shares the spirit of using an LLM as an autonomous prompt editor, but extends it to the multimodal realm where textual critiques alone are insufficient. Our agent actually sees images and can thereby ground its suggestions in visual evidence, a capability beyond the scope of text-only optimizers like PromptAgent or CriSPO.

Bayesian Optimization and Bandits: Bayesian optimization (BO) is renowned as a sample-efficient strategy for black-box optimization, traditionally applied to hyperparameter tuning and experimental design [11, 26]. In the era of LLMs, BO principles have been adapted for prompt optimization [24] and multi-task soft prompt transfer [18], while frameworks like LLAMBO leverage LLMs to warm-start BO surrogates [21]. However, optimizing discrete text prompts for expensive, closed-source multimodal models presents a unique challenge where standard continuous BO surrogates struggle. Therefore, our work aligns with the Multi-Armed Bandit (MAB) formulation of Bayesian inference. While recent APO methods like PromptAgent [32] rely on Monte Carlo Tree Search (MCTS) for directed planning, APO-Pair treats prompt selection as a Thompson-sampling bandit. This Bayesian scheduling explicitly tracks the uncertainty of each prompt's success rate via Beta posteriors. In the context of vision-language tasks where iterative visual evaluation is computationally expensive, this provides a mathematically principled balance between exploiting high-performing prompts and exploring novel evolutionary branches.

3. Methodology

APO–Pair (***A**utomatic **P**rompt **O**ptimization with **Pair**wise Contrastive Reasoning*) is an agent-driven framework that casts fine-grained recognition as a pairwise contrastive task and optimizes a prompt for a frozen vision–language model (VLM). Following previous work [16], we use Qwen-2VL-7B [31] as our evaluator for direct comparison. The framework decides whether to accept or reject each prompt edit strictly according to the resulting pairwise performance. A multimodal LLM agent, built with Claude 3.7 Sonnet [5], proposes interpretable YAML patches after contrasting its own successes and failures, while a Thompson-sampling bandit allocates the evaluation budget across competing prompt versions. Figure 1 gives a schematic overview; Algorithm 1 formalizes an optimization cycle. A complete working example can be found in Appendix A. Additionally, specific hyperparameters are detailed in Appendix B.

3.1. Problem Setup

We start from a labeled fine-grained dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ with class set $[K]$. From $\mathcal{D}$ we sample a pool of *image pairs* $\mathcal{P} = \{(x_i, x_j, g_{ij})\}$, where $g_{ij} = 1$ (a *positive* pair) if the two images share the same class label and $g_{ij} = 0$ (a *negative* pair) otherwise. [1]

A frozen VLM f_θ is queried only through *inference*: given a text prompt $P \in \mathcal{T}$ and a pair (x_i, x_j), the model produces

$$\hat{g}_{ij}(P) \in \{0, 1\}, \quad e_{ij}(P) \in \mathcal{E},$$

where $\hat{g}_{ij}$ is the binary decision ("`same`" or "`different`"). Along with the binary verdict, the evaluator returns a short natural-language rationale e_{ij} explaining the decision (30 words max). These rationales are stored in the context memory and later shown to the agent as auxiliary feedback during prompt updates. Initial base prompt is shown in Appendix C.

3.2. Optimization Loop

Each iteration t proceeds through four stages:

1. **Prompt selection.** Thompson sampling draws a candidate $P^{(t)}$ from the prompt repository $\mathcal{R}$ by sampling its Beta posterior and picking the arm with the highest sampled accuracy.
2. **Context aggregation.** From the Context Memory DB we retrieve: (i) a balanced batch $\mathcal{B}^{(t)}$ mixing n^- misclassified and n^+ correctly classified pairs from $\mathcal{P}_{\text{train}}$ with explanations (the explanations in the feedback bundle are the evaluator VLM's previously generated short rationales e_{ij} for those image pairs under the current prompt) ; (ii) rejected patch log for $P^{(t)}$. We use a rotation mechanism where previously used examples are moved to the back of the sampling list to ensure diverse context across iterations.
3. **Agentic patch proposal.** A multimodal LLM agent receives $\big(P^{(t)}, \mathcal{B}^{(t)}, \text{rejected-patch log}\big)$ and returns a targeted YAML patch $\Delta P^{(t)}$ (insert / modify / delete lines). Applying $\textsc{Merge}\big(P^{(t)}, \Delta P^{(t)}\big)$ yields a candidate prompt $\tilde{P}^{(t)}$.
4. **Acceptance test and update.** We run the frozen VLM on the pair dataset $\mathcal{P}$ (reusing cached responses whenever possible) to compute $\text{F1}\big(\tilde{P}^{(t)}; \mathcal{P}\big) = \frac{2PR}{P+R}$ where $P = TP/(TP + FP)$ and $R = TP/(TP + FN)$. If this score is $\geq \text{F1}\big(P^{(t)}; \mathcal{P}\big)$ the patch is accepted:
   - insert $\tilde{P}^{(t)}$ into $\mathcal{R}$ and initialize / warm-start its Beta counts;

[1] The pair pool is created once, off-line.

- update the running confusion-matrix statistics for every prompt affected;
- log the new explanations and outcomes to the context DB.

Otherwise we discard $\tilde{P}^{(t)}$ and append $\Delta P^{(t)}$ to the rejected-patch log so the agent will not retry it. The acceptance/rejection of a candidate patch is decided on the validation set.

3.2.1. Prompt Repository and Thompson Sampler

Every prompt version $P^{(k)}$ is treated as an arm in a Thompson-sampling bandit. After each full-pool evaluation we record its confusion-matrix counts (TP_k, FP_k, FN_k, TN_k) over *all* pairs in $\mathcal{P}$. Interpreting every pair decision as a Bernoulli trial (success = correct prediction), we maintain a conjugate $\mathrm{Beta}(\alpha_k, \beta_k)$ posterior:

$$\alpha_k = 1 + TP_k + TN_k, \qquad \beta_k = 1 + FP_k + FN_k.$$

At iteration t we draw $\tilde{\rho}_k^{(t)} \sim \mathrm{Beta}(\alpha_k, \beta_k)$ and select $P^{(t)} = \arg\max_k \tilde{\rho}_k^{(t)}$. Under standard i.i.d. assumptions, Thompson sampling enjoys $\tilde{O}(K \log T)$ Bayesian regret.

3.2.2. Balanced Contrastive Batch Selection

The Context Memory DB logs tuples $(P, x_i, x_j, g_{ij}, \hat{g}_{ij}, e_{ij})$ from $\mathcal{P}_{\text{train}}$. Each batch $\mathcal{B}$ contains $n^- = 2$ misclassified pairs and $n^+ = 2$ correctly classified pairs, stratified by error type. To prevent immediate re-use and ensure diverse learning signals, we maintain rotation lists moving recently sampled pairs to the back of their respective lists.

3.2.3. Visual-Aware Agent Updates and Patch Merge

The evolution agent A_Φ is given:

(i) the current prompt $P^{(t)}$, (ii) the balanced batch $\mathcal{B}$ (images *and* explanations), and (iii) the log of previously rejected patches for $P^{(t)}$.

The agent is instructed to modify only the evaluation guidelines within the EVAL block (between `### BEGIN_EVAL_GUIDELINES` and `### END_EVAL_GUIDELINES`) of the prompt template. It produces a YAML patch ΔP, example:

```
+ 3: "Emphasise beak curvature and eye-
    ring color"
~ 1: "Replace 'overall color' with '
    plumage pattern and wing bars'"
- 5: "Remove background-based cues"
```

where +, ~, and − denote insertion, modification, and deletion. Applying $\textsc{Merge}(P^{(t)}, \Delta P)$ yields a candidate prompt $\tilde{P}$. The complete agent instructions are provided in Appendix D.

Algorithm 1 APO–PAIR

Require: Training pairs $\mathcal{P}_{\text{train}}$, validation pairs $\mathcal{P}_{\text{val}}$, eval VLM f_θ, agent A_Φ, max iterations $T_{\max}$

Evaluate base prompt $P^{(0)}$ on $\mathcal{P}_{\text{val}}$; initialize $\mathcal{R}$ with Beta counts
$t \leftarrow 1$, no_improv_count $\leftarrow 0$
while $t \le T_{\max}$ **and** no_improv_count < 10 **do**
  (1) Thompson draw: $P^{(t)} \leftarrow \arg\max_k$ sample from $\mathrm{Beta}(\alpha_k, \beta_k)$
  (2) Build context: contrastive batch $\mathcal{B}^{(t)}$ from $\mathcal{P}_{\text{train}}$ (with rotation), rejected patch log
  (3) Agent proposes patch: $\Delta P^{(t)} \leftarrow A_\Phi(P^{(t)}, \mathcal{B}^{(t)}, \text{rejected log})$
  Candidate $\tilde{P}^{(t)} \leftarrow \textsc{Merge}(P^{(t)}, \Delta P^{(t)})$
  (4) Evaluate $\tilde{P}^{(t)}$ on $\mathcal{P}_{\text{val}} \rightarrow \mathrm{F1}_{\text{new}}$
  if $\mathrm{F1}_{\text{new}} \ge \mathrm{F1}(P^{(t)})$ **then**
    Accept: add $\tilde{P}^{(t)}$ to $\mathcal{R}$, update Beta counts, log predictions
    no_improv_count $\leftarrow 0$
  else
    Reject: log $\Delta P^{(t)}$ only
    no_improv_count $\leftarrow$ no_improv_count $+ 1$
  end if
  $t \leftarrow t + 1$
end while
return best prompt in $\mathcal{R}$

We adopt an anchor-based protocol to enable contrastive visual comparisons: for every task we randomly select an anchor image per class from the training set with a seed for consistency across experiments, let the model compare each test image against all anchors, and choose the class whose anchor yields the highest $\log\big(P(\texttt{Same})/P(\texttt{Different})\big)$ margin. This converts pairwise judgments into standard top-1 accuracy metrics while providing interpretable confidence scores.

3.3. Datasets

Following Hong et al. [16], we evaluate on three fine-grained benchmarks, each reduced to small subsets that are notoriously hard to distinguish. This evaluation protocol enables direct comparison with the most relevant baseline for fine-grained classification with VLMs. For **CUB-200** [30] we use three three-class splits: CUB_CUCKOO, containing the Black-billed, Mangrove, and Yellow-billed cuckoos; CUB_ORIOLE, with Hooded, Orchard, and Scott's orioles; and CUB_VIREO, composed of Philadelphia, Red-eyed, and Warbling vireos. From **iNaturalist-2021** [29] we derive INAT_BUTTERFLY (Symbrenthia lilaea, Claudina crescent, Elada checkerspot) and INAT_LUPINE (Arctic, Silvery, and Arizona lupines). Finally, in **VegFru** [17] we utilize VEGFRU_GREENS—Dandelion, Shepherd's

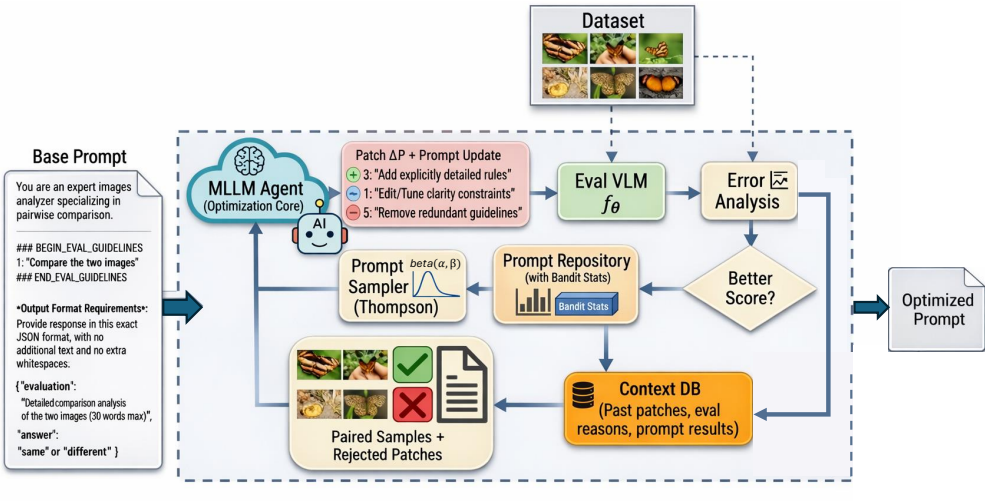


Figure 1. APO-Pair overview. The optimization loop starts from the base prompt, which initializes the prompt repository. At each iteration, the Thompson sampler selects a prompt variant; the context DB retrieves a balanced set of previously correct and incorrect image-pair decisions together with their rationales; the multimodal agent proposes a YAML patch; and the frozen evaluation VLM scores the updated prompt on the validation pair set. The resulting score determines whether the candidate is accepted into the repository and lineage or rejected and logged in memory. This loop repeats until the iteration budget or early-stopping criterion is reached.

purse, Prickly lettuce—and VEGFRU_ALLIUM—Leek, Green Chinese onion, Bunching onion. These challenging subsets represent scenarios where even subtle visual differences make a big difference, making them ideal testbeds for our pairwise contrastive approach.

3.4. Baselines

We compare seven variants. *Base* is the initial prompt we optimize from; its entire text is reproduced in Appendix A for reference. *One-shot* is APO-Pair with one-shot initialization: one labeled exemplar per class is used to derive initial class-specific comparison rules, which are then evolved for four generations on the small support set before inference. *APO-Pair (Neg.)* runs the full APO-Pair loop but feeds the agent only misclassified (negative) pairs, providing a basis for comparison against our balanced method with both positive and negative evidence. *APO-Pair (text)* supplies both positive and negative pairs but removes the images, so the agent sees only the VLM's textual rationales; this isolates the value of visual feedback. *APO-Pair* is the complete APO-Pair system. We also compare two existing prompt optimization methods: AutoSEP [16], a visual prompt optimization method for fine-grained recognition, and GEPA [2], a reflective prompt optimizer that uses natural language feedback and Pareto-based candidate selection to evolve prompts. Using Qwen-2VL-7B consistently across all methods ensures a controlled comparison.

4. Experiments

4.1. Results

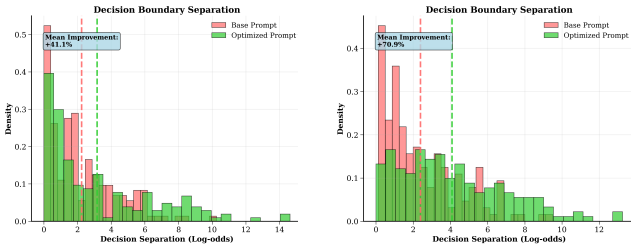
Table 1 presents top-1 accuracy across all seven fine-grained evaluation sets. APO-Pair achieves the highest performance on six of seven tasks, with substantial improvements over the base prompt: +15.2pp on iNat_butterfly, +15.6pp on iNat_lupine, and +20.8pp on VegFru_greens. The method consistently outperforms the one-shot baseline, demonstrating that iterative optimization surpasses static initialization. The results validate our key design choices. Balanced feedback proves essential: APO-Pair (Neg.) underperforms on five of seven tasks, confirming that learning from both successes and failures stabilizes optimization. Visual evidence is equally critical: APO-Pair (text) excels on some bird tasks but struggles on plant recognition, where subtle morphological differences require direct visual analysis.

Compared to existing optimizers, APO-Pair demonstrates clear advantages. It outperforms AutoSEP on the iNaturalist (+15.4pp butterfly, +4.4pp lupine) and VegFru tasks (+3.1pp greens, +10.7pp allium), while AutoSEP performs better on CUB_vireo. Both methods optimize universal prompts without fine-tuning, but differ in strategy: AutoSEP optimizes description-generation prompts, while APO-Pair directly optimizes evaluation guidelines for pairwise visual contrast. Similarly, APO-Pair outperforms GEPA on six of the seven tasks. While GEPA effectively optimizes prompts using natural language reflection, its reliance on text-based feedback limits its ability to capture the subtle morphological distinctions required for these domains, further underscoring the necessity of visual con-

Table 1. Top-1 accuracy across all fine-grained datasets. iNat_butterfly, iNat_lupine, CUB_oriole, CUB_vireo, CUB_cuckoo, VegFru_greens, and VegFru_allium.

	iNaturalist		CUB			VegFru	
Method	iNat_butterfly	iNat_lupine	CUB_cuckoo	CUB_oriole	CUB_vireo	VegFru_greens	VegFru_allium
Base	0.556	0.435	0.558	0.600	0.482	0.545	0.615
One-shot	0.667	0.509	0.558	0.625	0.500	0.611	0.522
GEPA	0.684	0.554	0.658	0.593	0.575	0.667	0.679
AutoSEP	0.554	0.547	0.634	0.551	**0.655**	0.722	0.624
APO-Pair (Neg.)	0.632	0.427	0.671	0.588	0.463	0.657	0.664
APO-Pair (text)	0.684	0.421	0.656	0.718	0.512	0.606	0.717
APO-Pair	**0.708**	**0.591**	**0.687**	**0.724**	0.529	**0.753**	**0.731**

trastive reasoning.



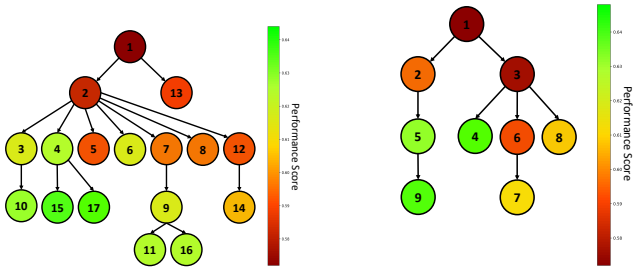
(a) iNat_lupine (b) iNat_butterfly

Figure 2. Distribution of decision-separation margins (log-odds of best vs. second-best anchor) for the base prompt (red) and the optimized APO-Pair prompt (green). Dashed vertical lines mark the respective means; call-outs report the relative improvement. Optimized prompts (green) achieve significantly higher separation compared to base (red). Statistical significance: $p < 0.001$.

4.2. Ablation Studies

All ablations are run on the two iNaturalist splits, iNat_butterfly (Symbrenthia lilaea, Claudina crescent, Elada checkerspot) and/or iNat_lupine (Arctic, Silvery, and Arizona lupines).

Decision-boundary separation. To gauge how clearly the optimized prompt separates confusable subclasses, we plot the confidence margin (the log-odds gap between the best anchor and the runner-up) for every image at inference. Figure 2 shows the distributions for both iNat splits. On iNat_lupine the mean margin moves from 2.24 to 3.16 log-odds, a $+41.1\%$ increase; on iNat_butterfly it rises by $+70.9\%$. Although the curves still overlap, highlighting the intrinsic difficulty of fine-grained recognition, the shift demonstrates that APO-Pair not only fixes borderline cases but broadens the internal gap between the correct class and its closest competitor, yielding more decisive predictions and fewer near-ties.



(a) iNat_lupine (b) iNat_butterfly

Figure 3. Prompt-evolution trees generated by APO-Pair with Thompson sampling. Each circle is an accepted prompt version during optimization, labelled by its version ID; circle color encodes the pair-wise score at the moment of acceptance (green=higher, red=lower).

Prompt-selection strategy The default optimizer controller relies on Thompson sampling: after each full-pool evaluation the confusion-matrix counts update a Beta posterior and the next prompt is sampled in proportion to its posterior success probability. We compare against a *purely greedy* rule that always picks the highest-scoring prompt so far, and an ε-*greedy* rule with $\varepsilon = 0.1$ that occasionally selects a random prompt to force exploration.

Table 2. Impact of the sampling rule on iNaturalist accuracy.

Strategy	iNat_butterfly	iNat_lupine
Thompson sampling	**0.708**	**0.591**
Greedy	0.638	0.497
ε-greedy	0.684	0.458

Thompson sampling achieves the best top-1 accuracy on both subsets, outperforming the greedy baseline by $+7.0$pp on butterflies and $+9.4$pp on lupines. The greedy rule converges quickly but often to a sub-optimal prompt, while ε-greedy expends valuable evaluation budget on uninformed random choices. The Bayesian controller there-

fore provides a superior trade-off between exploitation and uncertainty-driven exploration, justifying its use in the full APO-Pair pipeline.

Prompt-evolution dynamics. Figure 3 plots the accepted prompt lineages for iNAT_LUPINE (3a) and iNAT_BUTTERFLY (3b). The lineages never collapse into a single chain; Thompson sampling maintains multiple active branches, investing in prompts with uncertain potential. Many of the highest-scoring variants (bright-green circles) appear two or three hops from the root, demonstrating that initially weak ancestors can serve as valuable stepping stones toward better prompts, a cumulative improvement pattern observed across optimization domains [38]. These represent opportunities that a greedy approach would miss.

Evolution depth varies by domain: iNAT_LUPINE develops 17 versions (best v17 from v4) while iNAT_BUTTERFLY stabilizes at 9 versions (best v4 from v3), suggesting plant species' subtle distinctions require more iterative refinement than butterflies' distinctive features. The branching structure reveals hierarchical refinement rather than random exploration, validating uncertainty-aware optimization.

Structured vs. Freeform Prompt Evolution To validate the effectiveness of our YAML-style structured editing approach, we compare it against a freeform optimization baseline where the LLM agent can rewrite the entire prompt without structural constraints. Both methods use the same visual feedback and Thompson sampling framework, differing only in how the agent modifies prompts.

On the iNat_butterfly dataset, our structured YAML approach achieves superior performance (0.708) compared to the freeform baseline (0.684) while producing significantly more concise prompts (89 vs. 342 words; see Appendices E.2 and F.2 for full prompts). The YAML format enables precise ablation studies, as individual guidelines can be easily removed or modified to assess their contribution. In contrast, the freeform approach produces verbose, hierarchical text where concepts are intertwined across multiple sections, making targeted modifications difficult (see Fig. 4).

This comparison demonstrates that structured, interpretable prompt evolution achieves superior performance with more analyzable and efficient prompts. Complete prompts and additional analysis with iNat_lupine are provided in Appendix F.

YAML

```
1: Compare wing patterns, colors, markings
    , and body structure only between
    organisms in the same life stage.
2: Focus on distinctive taxonomic features
     rather than similarities in pose,
    background, or activity.
3: Note that similar coloration alone
    doesn't indicate same species; examine
     pattern arrangement and wing shape.
...
```

Freeform

```
# Species Identification Protocol for
    Butterfly Image Comparison
## Primary Analysis Framework
1. Focus exclusively on the butterfly's
    distinctive morphological features:
   - Wing pattern: Examine the specific
       arrangement of colors, bands, spots
       , and borders
   - Wing shape: Note the outline, edges,
       and proportional dimensions
   - Body structure: Observe thorax and
       abdomen characteristics when
       visible
   - Antennae: Note length and shape if
       visible
2. Disregard environmental factors that
    don't indicate species:
...
```

Figure 4. Comparison of structured YAML and Freeform prompt outputs. The YAML enforces concise, modular rules, whereas the freeform baseline tends toward verbose, hierarchical text.

Table 3. Impact of rejection memory on prompt evolution dynamics and performance.

Dataset	Rejection Memory	Total Evolutions	Best Version	Top-1 Acc.
lupine	✗	7	v3	0.463
	✓	18	v17	0.591
butterfly	✗	2	v3	0.649
	✓	8	v4	0.708

Rejection Memory Ablation To validate our rejection memory mechanism, we compare optimization runs with and without providing previously rejected patches as context to the agent. Table 3 shows that rejection memory significantly impacts exploration and performance. Without rejection context, iNat_butterfly achieves only 0.649 accuracy with 2 evolutions, while with rejection memory it reaches 0.708 accuracy across 8 evolutions (+5.9pp). On iNat_lupine, the effect is more pronounced: 0.463 accuracy with 7 evolutions versus 0.591 accuracy with 18 evolutions (+12.8pp). This demonstrates that rejection memory prevents the agent from repeatedly proposing similar failed modifications, enabling more diverse exploration by focusing creative efforts on genuinely novel prompt variations

rather than revisiting unsuccessful attempts.

Table 4. Studying the impact of modality and batch size (number of pairs) on top-1 accuracy (iNaturalist splits). "Reasoning Only" feeds the agent the VLM's textual rationales without images, while "Image Desc. Only" relies on independent LLM image descriptions. "W/ Images" represents the full multimodal APO-Pair pipeline, which maxes out at 4 pairs due to API image limits.

Pairs	Reasoning Only		Image Desc. Only		W/ Images (APO-Pair)	
	butterfly	lupine	butterfly	lupine	butterfly	lupine
4	0.684	0.421	0.649	0.497	**0.708**	**0.591**
8	0.695	0.485	0.643	0.418	-	-
16	0.667	0.508	0.690	0.395	-	-
32	0.729	0.456	0.632	0.537	-	-

Text-only feedback and batch-size exploration. Here, we evaluate the optimization dynamics when the agent receives *only textual cues* rather than images. Beyond isolating the value of visual feedback, this setting allows us to explore the impact of larger batch sizes. Because our multimodal agent is constrained to a maximum of ten images per API call, the visual pipeline is bottlenecked at four pairs (eight images) per iteration. By relying solely on text, this constraint is lifted, allowing us to sweep batch sizes up to 32 pairs (Table 4). *Reasoning Text* feeds the agent the VLM's natural-language rationales without images; *Image Desc. Only* inserts an auxiliary step that asks the LLM to describe each image first, passing those descriptions in place of the pixels. Both settings allow larger batches because no images are transmitted.

A primary observation from these results is that visual context provides a significantly stronger and more sample-efficient optimization signal than simply scaling up text-based examples. As shown in Table 4, the full multimodal APO-Pair system with its maximum capacity of 4 image pairs (0.708 on butterfly, 0.591 on lupine) is vastly more efficient and robust than text-only variants. While the "Reasoning Only" variant edges out the multimodal system on iNat_butterfly at 32 pairs (0.729 vs. 0.708), it requires eight times the amount of evaluation data to achieve this gain. Furthermore, this text-only scaling proves highly brittle across domains: on the challenging iNat_lupine split, the 32-pair text variant collapses to 0.456, falling far behind the 4-pair multimodal system (0.591). The description variant shows similarly inconsistent patterns. This instability suggests that text-only signals are fragile; independently generated descriptions may omit the fine details the agent needs, while raw rationales expose only the features the evaluator was able to focus on, without painting the whole picture. Images give the agent the full visual context, letting it decide which cues matter and yielding steadier improvements and cross-domain reliability. Closing this gap remains an open direction: more structured or multi-turn descriptions might capture the fine-grained cues now missing from single-prompt text. Future work could explore richer visual caption pipelines or joint reasoning over sets of descriptions to give the agent a fuller picture without exceeding image input limits.

5. Conclusion, Limitations and Future Work

We introduced APO-Pair, a framework that treats fine-grained classification as a pairwise task and optimizes a single class-agnostic prompt for a frozen multimodal LLM. A Thompson-sampling controller maintains multiple prompt lineages in parallel, while a multimodal agent refines each prompt through interpretable YAML patches grounded in both successes and failures. Experiments on seven challenging fine-grained splits drawn from CUB, iNaturalist 2021, and VegFru show that APO-Pair consistently outperforms a generic prompt, one-shot baseline, and the recent AutoSEP zero-shot optimizer, achieving an average improvement of 10.4pp. Ablations reveal that balanced positive–negative feedback with visual evidence are critical, and that Bayesian exploration yields higher accuracy and wider decision margins than greedy or ε-greedy alternatives. Qualitative lineage analysis further demonstrates that the method promotes open-ended, hierarchical knowledge accumulation rather than a single greedy walk.

While promising, our anchor-based evaluation scales quadratically with the number of classes, limiting direct applicability to large taxonomies with hundreds or thousands of categories. Practical deployment would require efficiency strategies such as hierarchical coarse-to-fine classification that first narrows to a subtree, embedding-based pre-filtering to reduce candidate anchors to a top-k shortlist, or learned anchor selection—directions we leave for future work. Additionally, our text-only feedback experiments reveal brittle performance patterns, and the impact of pair pool size on optimization dynamics warrants further investigation. See Appendix G for detailed discussion.

The framework is generalizable to any dataset, requires no gradient updates, and produces a transparent audit trail of prompt edits. Looking ahead, this interpretable lineage lays the groundwork for continuous learning loops where feedback from human domain experts or specialized AI agents could dynamically refine the prompt repository over time. Furthermore, while our method relies on large VLMs, the high-quality reasoning generated by APO-Pair could serve as a valuable supervision signal for model distillation. This would streamline the transition from an optimized prompt to a lightweight, production-ready model, reducing inference latency and throughput bottlenecks. Ultimately, we hope these results encourage broader adoption of agentic prompt optimization as a scalable, transparent alternative to conventional fine-tuning for fine-grained visual understanding.

References

[1] Abdelrahman Abdelhamed, Mahmoud Afifi, and Alec Go. What do you see? enhancing zero-shot image classification with multimodal large language models. *arXiv preprint arXiv:2405.15668*, 2024. 1

[2] Lakshya A Agrawal et al. Gepa: Reflective prompt evolution can outperform reinforcement learning, 2025. 2, 5

[3] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Proceedings of the 25th Annual Conference on Learning Theory*, pages 39.1–39.26, Edinburgh, Scotland, 2012. PMLR. 2

[4] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millicah, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2022. Curran Associates Inc. 1

[5] Anthropic. Claude 3.7 sonnet. https://www.anthropic.com/news/claude-3-7-sonnet, 2025. 3

[6] Rin Ashizawa, Yoichi Hirose, Nozomu Yoshinari, Kento Uchida, and Shinichi Shirakawa. Bandit-based prompt design strategy selection improves prompt optimizers. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20799–20817, Vienna, Austria, 2025. Association for Computational Linguistics. 2

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020. 2

[8] Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3369–3391, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics. 2

[9] Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: self-referential self-improvement via prompt evolution. In *Proceedings of the 41st International Conference on Machine Learning*. JMLR.org, 2024. 2

[10] Gregor Geigle, Radu Timofte, and Goran Glavaš. African or European swallow? benchmarking large vision-language models for fine-grained object classification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2653–2669, Miami, Florida, USA, 2024. Association for Computational Linguistics. 2

[11] Stewart Greenhill, Santu Rana, Sunil Gupta, Pratibha Vellanki, and Svetha Venkatesh. Bayesian optimization for adaptive experimental design: A review. *IEEE access*, 8: 13937–13948, 2020. 3

[12] 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 *The Twelfth International Conference on Learning Representations*, 2024. 2

[13] Hulingxiao He, Geng Li, Zijun Geng, Jinglin Xu, and Yuxin Peng. Analyzing and boosting the power of fine-grained visual recognition for multi-modal large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. 1, 2

[14] Han He, Qianchu Liu, Lei Xu, Chaitanya Shivade, and Yi Yang. Crispo: Multi-aspect critique-suggestion-guided automatic prompt optimization for text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025. 1, 3

[15] Junwen He, Yifan Wang, Lijun Wang, Huchuan Lu, Jun-Yan He, Jin-Peng Lan, Bin Luo, and Xuansong Xie. Multi-modal instruction tuned llms with fine-grained visual perception. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13980–13990, 2024. 1

[16] Yunqi Hong, Sohyun An, Andrew Bai, Neil Y.C. Lin, and Cho-Jui Hsieh. Unlabeled data improves fine-grained image zero-shot classification with multimodal llms. *arXiv preprint arXiv:2506.03195*, 2025. 1, 2, 3, 4, 5

[17] Saihui Hou, Yushan Feng, and Zilei Wang. Vegfru: A domain-specific dataset for fine-grained visual categorization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 541–549, 2017. 4

[18] Haeju Lee, Minchan Jeong, Se-Young Yun, and Kee-Eung Kim. Bayesian multi-task transfer learning for soft prompt tuning. *arXiv preprint arXiv:2402.08594*, 2024. 3

[19] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 1, 2

[20] Mingxuan Liu, Subhankar Roy, Wenjing Li, Zhun Zhong, Nicu Sebe, and Elisa Ricci. Democratizing fine-grained visual recognition with large language models. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024. 2

[21] Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mihaela van der Schaar. Large language models to enhance bayesian optimization. *arXiv preprint arXiv:2402.03921*, 2024. 3

[22] Yexin Liu, Zhengyang Liang, Yueze Wang, Muyang He, Jian Li, and Bo Zhao. Seeing clearly, answering incorrectly: A multimodal robustness benchmark for evaluating mllms on leading questions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9087–9097, 2025. 2

[23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2

[24] Lennart Schneider, Martin Wistuba, Aaron Klein, Jacek Golebiowski, Giovanni Zappella, and Felice Antonio

Merra. Hyperband-based bayesian optimization for black-box prompt selection. *arXiv preprint arXiv:2412.07820*, 2024. 3

[25] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2020. 2

[26] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012. 3

[27] OpenAI Team. Gpt-4 technical report. 2023. 1

[28] Prashant Trivedi, Souradip Chakraborty, Avinash Reddy, Vaneet Aggarwal, Amrit Singh Bedi, and George K. Atia. Align-pro: A principled approach to prompt optimization for llm alignment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(26):27653–27661, 2025. 1

[29] Grant Van Horn, Elijah Cole, Sara Beery, Kimberly Wilber, Serge Belongie, and Oisin Mac Aodha. Benchmarking representation learning for natural world image collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12884–12893, 2021. 2, 4

[30] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Computation & Neural Systems Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 1, 4

[31] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Ke-Yang Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *ArXiv*, abs/2409.12191, 2024. 1, 3

[32] Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric Xing, and Zhiting Hu. Promptagent: Strategic planning with language models enables expert-level prompt optimization. In *The Twelfth International Conference on Learning Representations*, 2024. 3

[33] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 2

[34] Xiu-Shen Wei, Yi-Zhe Song, Oisin Mac Aodha, Jianxin Wu, Yuxin Peng, Jinhui Tang, Jian Yang, and Serge Belongie. Fine-Grained Image Analysis With Deep Learning: A Survey . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 44(12):8927–8948, 2022. 1

[35] Jinyu Xiang, Jiayi Zhang, Zhaoyang Yu, Fengwei Teng, Jinhao Tu, Xinbing Liang, Sirui Hong, Chenglin Wu, and Yuyu Luo. Self-supervised prompt optimization. *ArXiv*, abs/2502.06855, 2025. 2

[36] Dingchen Yang, Bowen Cao, Guang Chen, and Changjun Jiang. Pensieve: Retrospect-then-compare mitigates visual hallucination. *ArXiv*, abs/2403.14401, 2024. 2

[37] Hong-Tao Yu, Xiu-Shen Wei, Yuxin Peng, and Serge Belongie. Benchmarking large vision-language models on fine-grained image tasks: A comprehensive evaluation. *ArXiv*, abs/2504.14988, 2025. 1

[38] Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin godel machine: Open-ended evolution of self-improving agents. *ArXiv*, abs/2505.22954, 2025. 7

[39] Yuhui Zhang, Alyssa Unell, Xiaohan Wang, Dhruba Ghosh, Yuchang Su, Ludwig Schmidt, and Serena Yeung-Levy. Why are visually-grounded language models bad at image classification? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 1

[40] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[41] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 1