Saliency-Aware Multi-Route Thinking: Grounding and Reasoning on Vision-Language Agents

Mingjia Shi[1,*] **Yinhan He**[1] **Yaochen Zhu**[1] **Cassie Dong**[2] **Jundong Li**[1,†]

[1]University of Virginia [2]Meta

[*]nzh3ru@virginia.edu [†]jundong@virginia.edu

Abstract

Vision-language (VL) agents increasingly combine extended reasoning with external visual tools. Yet long traces can drift from the image as generated text accumulates, while directly inserting tool outputs can turn one incorrect detection into a persistent premise. We ask how inference-time computation can improve reasoning without surrendering control to the tool. We propose *Saliency-Aware Principle Selection* (SAP), a training-free method that evolves compact reasoning principles rather than lengthening a single token trajectory. Each principle generates independent routes from the original image and prompt. A frozen grounding tool scores whether objects cited along these routes are visually supported; agreement, diversity, and uncertainty further control drift, collapse, and overconfidence. The tool supplies selection evidence but never writes the reasoning or answer. Across object-hallucination benchmarks, SAP consistently improves visually grounded prediction under a unified model and decoding budget. Broader evaluations spanning reasoning, knowledge, OCR, documents, diagrams, and real-world imagery show that this gain does not come at the expense of general VL capability. Cross-architecture and model-scale experiments further demonstrate that the principle-selection mechanism transfers beyond a single backbone, while controlled ablations confirm that both multi-route exploration and evolutionary selection are necessary. Because routes run in parallel, SAP also reduces serial latency relative to an equally long chain. Together, these results show that scaling over governed reasoning behaviors can preserve visual evidence more effectively than simply extending the chain. Code is available at `https://github.com/BDeMo/Saliency-Aware-Principle-Selection-NeurIPS-2026-Official`.

1 Introduction

Vision–language (VL) agents increasingly use images when answering visual questions [Bai et al., 2025], following language instructions in embodied navigation [Cheng et al., 2025], and calling external visual tools [Zhao et al., 2024, Wang et al., 2026b]. Across these settings, the image is encoded only once, at the beginning of autoregressive reasoning, while textual states are regenerated at every step. As the chain grows, an early visual error, such as mistaking a toy for a task-relevant object, can be repeated and elaborated as if it were grounded in the image. Figure 1 shows this failure developing: attention to image tokens falls as a single reasoning chain extends. Consequently, an unsupported textual hypothesis becomes harder to correct and can propagate through later

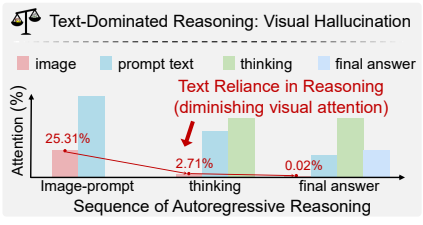


Figure 1: Image attention collapses along Qwen3-VL-8B-Thinking on MS-COCO.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

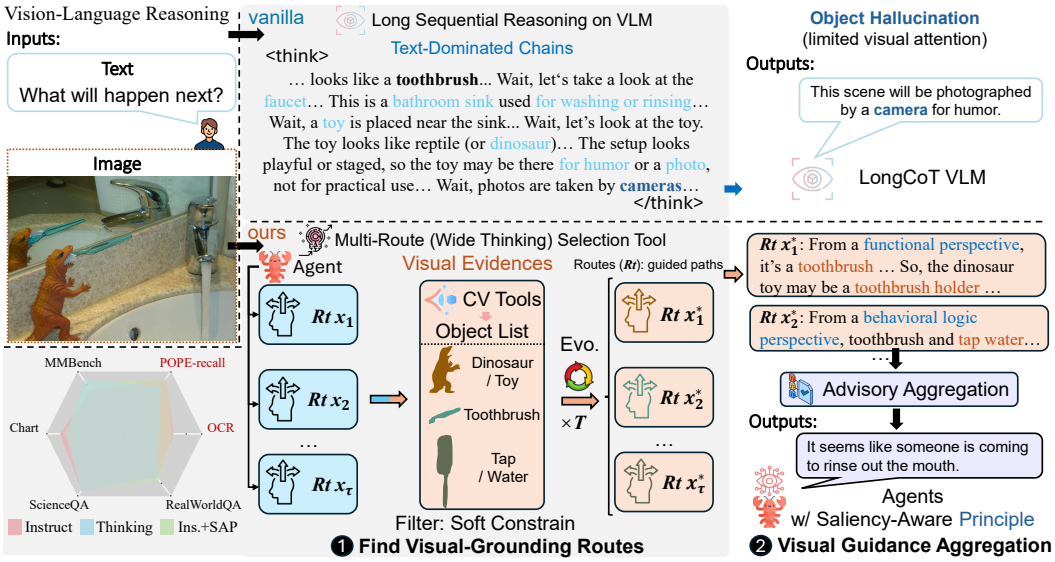


Figure 2: The motivation and pipeline of **Saliency-Aware Principle (SAP)** route selection tool. Given an image $\mathcal{I}$ and a prompt p, a frozen grounding tool (SAM + captioner) produces a textual grounding summary g that is fed to the agent only as an *advisory* signal. The frozen language model φ maintains a population of $\mu + \lambda$ reasoning principles $\{x_i\}$; each principle instantiates τ short parallel reasoning routes $\{r_{i,j}\}$. Four ordinal signals—consensus c, diversity d, evidence validity e, uncertainty u—form a discrete fitness f_i, and a $(\mu+\lambda)$ elite-selection scheme keeps the top principles while sampling λ new ones from the elites. Iterating this loop T times realizes inference-time scaling through *principle-level evolution*, rather than through extending a single reasoning chain.

reasoning into the final answer. For an agent, this can produce a wrong answer or an action directed toward an object that is not present.

Long chain-of-thought adds inference compute by extending one textual trajectory, but it does not restore visual evidence once that trajectory becomes text dominated [Lu et al., 2019]. Sampling more trajectories has the same limitation [Wang et al., 2023b, Yao et al., 2023]. Grounded reasoning instead inserts boxes, captions, or object lists into the VLM context [Wang et al., 2026b,a, Zhao et al., 2024]; an incorrect tool output can then affect every later step. In our controlled SAM-augmented LongCoT study, direct insertion reduces POPE absent-object recall from 79.6 to 14.8, and advisory wording recovers only part of the loss (Appendix B.3). The remaining open question is *when and how an agent should call its tools while remaining able to disagree with them.*

We introduce *Saliency-Aware Principle Selection* (SAP), a training-free search over short reasoning principles (Fig. 2). A principle specifies how the VLM should inspect and verify visual evidence without prescribing a token-level trace. The frozen VLM reads each principle and generates several independent routes from the original image and prompt. Searching over principles therefore changes reasoning behavior while leaving image interpretation and answer generation with the VLM.

A frozen grounding tool evaluates the resulting behavior. For each principle, SAP measures whether objects cited by its routes appear in the grounding output and combines this evidence-validity signal with route agreement, within-principle diversity, and uncertainty. Evolutionary selection retains stronger principles and produces the next population through mutation and recombination. Here, saliency means consistency between cited and grounded objects; it does not model human visual attention or assume that every detected object is task-relevant. The tool affects which principles survive but does not write the routes or answers. SAP changes no model parameters, stores no state across queries, and allows routes within a generation to run in parallel.

We evaluate the targeted behavior on POPE, HallusionBench, and AMBER-d under a unified Qwen3-VL-8B setup. Sixteen additional vision, reasoning, OCR, document, diagram, and real-world benchmarks test whether SAP sacrifices capabilities outside object grounding. We match route-token budgets, include grounding and selection overhead, and vary the search components, backbone architecture, and model scale. SAP improves the three object-hallucination evaluations while remaining

competitive on the broader suite; the additional analyses identify the source and inference cost of these gains. Our contributions are:

1. We identify two failures in VL inference: long textual trajectories lose visual support, while direct tool insertion can propagate grounding errors through later reasoning.
2. We develop SAP, which uses grounded object support to select high-level reasoning principles without training the VLM or treating tool output as the answer.
3. SAP improves three object-hallucination evaluations; broader capability, transfer, ablation, and cost experiments characterize the scope and inference trade-offs.

2 Problem Formulation

In this paper, we formulate inference-time scaling as an optimization problem over reasoning routes executed during the inference phase. Given a prompt p and a language model φ, a reasoning route

$\rho \in \mathcal{R}$ specifies how inference unfolds across steps—determining the structure and progression of intermediate reasoning—where $\mathcal{R}$ denotes the space of feasible reasoning routes. The objective of inference-time scaling is to identify a reasoning route that maximizes a task utility $f(\rho; p, \varphi)$, which scores the outputs that route ρ induces under prompt p and language model φ. Suppressing (p, φ) when they are held fixed, we write

$$\rho^\star = \arg\max_{\rho \in \mathcal{R}} \; f(\rho). \tag{1}$$

Here, inference-time scaling does not refer solely to generating a longer chain of reasoning along a single trajectory. Instead, it refers to allocating additional inference-time computation to *explore multiple alternative reasoning routes* and to select higher-quality ones. As the inference budget increases, more candidates in $\mathcal{R}$ can be instantiated and evaluated, improving the likelihood of identifying a reasoning route with higher utility. In this sense, inference-time scaling corresponds to scaling the *search over reasoning behaviors* rather than merely extending a single route.

3 Methodology

To address the two failure modes of Section 1, we propose *Saliency-Aware Principle Selection* (SAP), a training-free inference-time method that replaces both rigid recipes with a search over a discrete principle space. Concretely, SAP combines ***(i)*** principle-guided reasoning generation, parameterizing reasoning behaviors via high-level principles; ***(ii)*** evolutionary principle refinement, performing population-based selection under ordinal feedback; and ***(iii)*** saliency-aware evaluation, supplying a modality-aware fitness signal. An overview is illustrated in Fig. 2.

3.1 Principle-Guided Reasoning Generation

3.1.1 Intractable Route Search

Direct optimization of the task utility f over the reasoning route space $\mathcal{R}$ as shown in Eq. (1) to achieve test-time scaling for VLM is intractable due to both theoretical and practical reasons. First, we note that a reasoning route is not a fixed-length variable but the emergent output of autoregressive generation under the language model φ. As inference-time computation is scaled, reasoning routes become longer and increasingly sparse, making direct sampling and evaluation more time-consuming in practice. In addition, the space $\mathcal{R}$ is inherently discrete and redundant.

Small local modifications to token sequences often do not lead to meaningful changes in the final reasoning outcome, while semantically distinct reasoning behaviors may be separated by large combinatorial gaps. Therefore, information in the route space is sparsely distributed: many distinct routes induce similar final outputs, while high-quality routes are rare and difficult to identify. This discreteness and sparsity make it difficult to efficiently compare, rank, or refine reasoning routes, rendering direct optimization over $\mathcal{R}$ impractical. To make this problem amenable to optimization, we parameterize the routes using a set of high-level *reasoning principles*.

3.1.2 Principle-guided reasoning

We start by defining *principle reasoning* as a form of inference in which the reasoning process is guided by explicit, high-level principles that specify *how* the model should reason, rather than the exact intermediate tokens to be generated. Compared to a specific route, a reasoning principle provides guidance on reasoning behavior—such as how visual evidence should be revisited during the generation process, how intermediate hypotheses should be verified, or how different modalities should be balanced—without constraining to a particular token-level realization.

For example, in a vision–language reasoning task, a principle may instruct the model to *explicitly re-examine visual evidence whenever an intermediate textual conclusion is formed* or to *verify spatial relations by checking the relative positions of all relevant objects in the image*. We note that such a principle x is itself a compact textual description, but it does not prescribe a single reasoning trace. Instead, under a fixed language model, the same principle $x \in \mathcal{X}$ may induce multiple concrete reasoning trajectories, depending on how the model instantiates the principle at the token level. We denote each such instantiation as a reasoning route $\rho(x)$ given principle x. This parameterization yields the equivalent optimization objective

$$x^\star = \arg\max_{x \in \mathcal{X}} \; f\big(\rho(x)\big), \tag{2}$$

which shifts inference-time optimization from unstructured routes to a compact principle space.

3.1.3 Principle Selection

Principle-guided reasoning shifts inference-time control from token-level trajectories to high-level reasoning directives. This shift raises a central question: *How should reasoning principles be selected and refined during inference?*

In VLMs, principle selection is challenging for two reasons closely tied to multimodal inference. First, during long-horizon generation, visual evidence becomes increasingly difficult to revisit, making it necessary to favor principles that explicitly preserve visual grounding throughout reasoning. In addition, feedback available at inference time is inherently discrete and ordinal, providing only relative comparisons between reasoning behaviors (e.g., preferring A to B) rather than forcing them into scalar scores.

These properties indicate that principle selection should operate directly over discrete candidates, rely on relative comparisons instead of absolute scores, and remain robust under noisy and inconsistent feedback. This principle-level view also separates behavioral search from token-level realization: selection decides which reasoning directive should survive, while the frozen model remains free to instantiate that directive through different routes. Maintaining a population further preserves competing behaviors long enough to compare their visual support, rather than committing early to a single chain that may already have drifted. Motivated by these requirements, we adopt a population-based evolutionary selection strategy to refine reasoning principles. The next subsection instantiates this strategy over the principle space, with visual saliency providing the only modality-aware supervision.

3.2 Saliency-Aware Principle Evolution for Vision–Language Reasoning

p, x	prompt [input], reasoning principle [variable]
r, s	reasoning [thinking] and summary [output]
μ, λ, τ	#elites, #new ones, #routes [hyperparam.]
φ, f	LLM, utility and fitness metric on (r, s)

Built on the principle-level formulation in Eq. (2), we cast inference-time scaling as a saliency-guided evolutionary search over reasoning principles. The framework combines ***(i)*** principle-level reasoning that discovers routes, ***(ii)*** visual saliency as the modality-aware supervision signal, and ***(iii)*** population-based evolution as the optimization mechanism.

3.2.1 Approach overview

Given an image $\mathcal{I}$ and prompt p, a frozen grounding tool extracts salient elements $\mathcal{O} = \textsc{Ground}(\mathcal{I})$, used as a modality-aware reference for evaluation rather than injected into the language model. Conditioned on $(p, \mathcal{O})$, we initialize a population of $\mu + \lambda$ principles $\{x_i\}_{i \in [\mu+\lambda]}$. Each principle x_i instantiates τ routes via $(r_{i,j}, s_{i,j}) \sim \varphi(r, s \mid x_i, p, \mathcal{I}), \quad j \in [\tau]$, realizing the reasoning routes

Algorithm 1 Initialization: sample principles and τ routes

Require: images $\mathcal{I}$, prompt p, #init. N, #routes τ

$\mathcal{O} \leftarrow \textsc{Ground}(\mathcal{I})$, $x \leftarrow [\,]$
for $i = 1$ **to** N **do**
  $x_i \leftarrow \varphi_x(p, \mathcal{O})$ {sample one principle}
  $r_i, s_i \leftarrow [\,], [\,]$
  for $j = 1$ **to** τ **do**
    $r_{i,j}, s_{i,j} \leftarrow \varphi_{rs}(p, \mathcal{I}, x_i)$
    $\textsc{append}(r_i, r_{i,j})$
    $\textsc{append}(s_i, s_{i,j})$
  end for
  $\textsc{append}(x, x_i)$
end for
return x, $\{r_{i,j}, s_{i,j}\}_{i\in[N],\, j\in[\tau]}$

Algorithm 2 $(\mu + \lambda)$ Evolution: select elites and resample

Require: principles x, images $\mathcal{I}$, prompt p, #routes τ

$N \leftarrow |x|$ $\{N = \mu + \lambda\}$, $f \leftarrow [\,]$
for $i = 1$ **to** N **do**
  $\{(r_{i,j}, s_{i,j})\}_{j\in[\tau]} \leftarrow \textsc{Inference}(p, \mathcal{I}, x_i, \tau)$
  $\ell_i \leftarrow \textsc{DiscreteEval}(\{(r_{i,j}, s_{i,j})\}_{j\in[\tau]})$
  $f_i \leftarrow \textsc{MapLevelToScore}(\ell_i)$ {internal only}
  $\textsc{append}(f, f_i)$
end for
$\texttt{idx} \leftarrow \textsc{TopK}(f, k = \mu)$
$x^\mu \leftarrow x[\texttt{idx}]$; $x^\lambda \leftarrow [\,]$
for $n = 1$ **to** $N - \mu$ **do**
  $x_n \leftarrow \varphi_x(p, x^\mu)$ {offspring from elites}
  $\textsc{append}(x^\lambda, x_n)$
end for
$x^+ \leftarrow \textsc{concat}(x^\mu, x^\lambda)$
return x^+ new population of principles

$\rho(x_i)$ induced by the principle. Visual information thus remains available throughout inference, while different principles induce distinct reasoning behaviors.

3.2.2 Saliency-aware fitness and evolution

The primary supervision is **evidence validity** e_i: the fraction of cited visual references in $\{r_{i,j}\}_{j\in[\tau]}$ that match a grounded object in $\mathcal{O}$, capturing how consistently the principle's routes refer to detected visual evidence. We use "saliency" operationally to mean this object-grounding consistency; it is neither a model of human visual attention nor a claim that the detected objects are necessarily task-relevant. This is the only signal derived from the grounding tool and drives the search toward visually supported principles. At *initialization only*, the language model is given a textual report about $\mathcal{O}$ returned by the grounding tools—a compact list of object labels with image- and object-ids—used to seed the principle population; subsequent generations resample principles conditioned on elites and never re-receive $\mathcal{O}$. The e_i score itself is computed *externally* by canonicalizing each cited reference into the format `img#k_obj#j` ($\ell_{k,j}$) and matching it against the underlying label map. Thus, $\mathcal{O}$ enters the model only as an initialization-time textual prior, never as an authoritative answer or task supervision (Appendix C, Step 5).

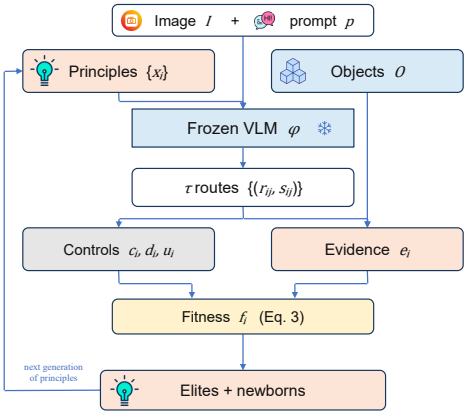


Figure 3: **Saliency-aware evolutionary loop.** Each principle generates parallel routes. Grounded object support and route-level control signals define fitness; elite principles and newborns form the next population for T rounds.

Three further *control* signals keep the search stable, each obtained from a lightweight external probe rather than from the reasoning trace itself: **(a) consensus** c_i—a lightweight external judge compares each principle's representative answer against the population majority $\bar{y}$ and emits an ordinal agreement label (anchor against drift); **(b) within-principle diversity** d_i—the same judge counts the number of distinct topics covered by the τ routes under x_i and discretizes that count into an ordinal level (anchor against collapse to one route); **(c) uncertainty** u_i—after each route the reasoning model is required to self-emit an ordinal certainty rating reflecting how confident it is in its own answer (suppressing over-confident or ambiguous outputs). All four signals are ordinal in

$\{\texttt{low}, \texttt{medium}, \texttt{high}\}$, reflecting the inherently ordinal and contrastive nature, and combine into a scalar fitness (full procedure in Appendix C, Steps 3–6):

$$f_i \;=\; \underbrace{w_e\,\mathrm{Score}(e_i)}_{\text{saliency-driven}} \;+\; \underbrace{w_c\,\mathrm{Score}(c_i) + w_d\,\mathrm{Score}(d_i) - w_u\,\mathrm{Pen}(u_i)}_{\text{control: anti-drift, anti-collapse, anti-overconfidence}}, \tag{3}$$

where Score and Pen are ordinal mappings $(\texttt{low}, \texttt{medium}, \texttt{high}) \to (0, 1, 2)$.

A $(\mu + \lambda)$ selection scheme retains the top-μ principles as elites and resamples λ new principles conditioned on the elites; iterating T times yields a population-based approximation to the principle-level objective in Eq. (2), allocating inference-time computation to progressively favor visually supported principles. Details: Alg. 1, 2. We next analyze the loop convergence and computation.

3.3 Framework Analysis

In this part, we provide theoretical analysis that quantifies how inference-time scaling operates in SAP by explicitly characterizing, in terms of (μ, λ, T), the probability of optimization improvement and the coverage of effective reasoning principles achieved by the evolutionary search, where T is the total number of iterations. Under the evolutionary algorithm with elite selection, iterative filtering ensures that the population fitness is non-decreasing across generations, such that the best-performing principles monotonically improve as optimization proceeds.

Theorem 3.1 (Optimization and generalization with $(\mu + \lambda)$ evolution). *Consider the $(\mu + \lambda)$ evolutionary procedure over a principle space $\mathcal{X}$. Let F_t denote the best fitness in the population at iteration t, and let T be the total number of iterations. Then the following properties hold:*

1. *(**Optimization stability**) Due to elite selection, the best fitness is non-decreasing: $F_{t+1} \geq F_t,\ \forall t$. Moreover, if each offspring sample has probability at least q_t of instantiating a principle with fitness strictly greater than F_t, then the probability of improvement in one iteration satisfies $\Pr(F_{t+1} > F_t) \geq 1 - (1 - q_t)^{\lambda}$.*
2. *(**Generalization as coverage**) Suppose there exists a set of effective principles $\mathcal{X}_{\mathrm{good}} \subset \mathcal{X}$, and each offspring sample has probability at least $q > 0$ of belonging to $\mathcal{X}_{\mathrm{good}}$. Then the probability that the evolutionary procedure discovers at least one effective principle within T iterations is lower bounded by $1 - (1 - q)^{\lambda T}$.*

Theorem 3.1 explains how inference-time scaling emerges in SAP. The optimization guarantee ensures stability: elite preservation prevents degeneration by retaining the best-performing principles across iterations. At the same time, the improvement bound shows that increasing λ raises the probability of discovering strictly better principles at each iteration. Beyond optimization, the generalization bound characterizes the growing *coverage of the principle space*. Since each iteration evaluates λ newly sampled principles, after T iterations the search explores on the order of λT distinct candidates. As λ or T increases, the probability of encountering effective principles outside the current population grows monotonically. Together, these results show that inference-time scaling in SAP is achieved by allocating additional computation to explore and select among alternative reasoning principles, rather than extending a single reasoning trajectory. Formal proofs are provided in Appendix E.

3.3.1 Comparison with long chain-of-thought

SAP further exhibits a scaling advantage over traditional long chain-of-thought (CoT) reasoning by operating in a parallel and population-based manner. Consider a fixed inference-time token budget L. In long CoT reasoning, tokens are generated sequentially along a single trajectory, and transformer-based decoding incurs a quadratic attention cost w.r.t. the sequence length, i.e., $\mathcal{O}(L^2)$, together with strict sequential dependencies that limit parallelism across decoding steps. In contrast, SAP allocates the same *route-token* budget across multiple shorter reasoning routes evaluated in parallel; complete system accounting, including grounding and selection overhead, is reported in Appendix B.12. Let $L = (\mu + \lambda)\tau\bar{\ell}$, where $\bar{\ell}$ denotes the average length of each reasoning route. Each route incurs an attention cost of $\mathcal{O}(\bar{\ell}^2)$, yielding a total attention cost of $\mathcal{O}\big((\mu + \lambda)\tau\,\bar{\ell}^2\big)$, which is strictly lower than $\mathcal{O}(L^2)$ when $\bar{\ell} \ll L$. Moreover, reasoning routes across different principles and routes are independent and can be executed in parallel, avoiding the sequential bottleneck inherent to long CoT. Therefore, SAP scales inference-time reasoning by parallel exploration and selection rather than by elongating a single reasoning chain, providing a more computationally efficient alternative.

3.3.2 Training-free scaling

SAP is a plug-to-play, fully training-free method: it requires no gradient updates, no manually labeled data, and no fine-tuning of either the language model or the external grounding service. The grounding tool is consulted only at inference time and its outputs are advisory, so no external knowledge is absorbed into the language model's parameters (cf. Appendix B.2). The evolutionary search is performed directly in the space of reasoning principles, using the fixed knowledge already encoded in the pretrained model, without introducing additional training data or external supervision. Rather than acquiring new knowledge, SAP explores organizing and applying existing knowledge during inference. This aligns closely with the nature of reasoning, which concerns how information is structured, combined, and verified, rather than what new information is learned [Zhu et al., 2025]. Consequently, SAP improves reasoning quality by reallocating computation to search for better reasoning behaviors within the model's capabilities, avoiding the cost and potential bias of data collection or retraining.

4 Empirical Study

We evaluate SAP on 16 vision–language benchmarks via VLMEvalKit [Duan et al., 2024], grouped into *Reasoning / Knowledge / General* and *OCR / Document / Diagram / Real-world* (Appendix D). We separately evaluate object hallucination on POPE [Li et al., 2023b], HallusionBench [Guan et al., 2024], and AMBER-d [Wang et al., 2023a]. The benchmarks target single-pass VL grounding and reasoning, matching the scope declared in Section 1; multi-turn conversational agent benchmarks include many non-perception turns and would dilute the perception signal we measure. All experiments are pure inference: no parameter updates, no per-dataset tuning.

Models. Qwen3-VL-8B [Bai et al., 2025] is the default, frozen backbone for all reasoning, principle generation, and aggregation steps; cross-architecture transfer additionally uses InternVL3.5-8B [Wang et al., 2025a] and DeepSeek-VL2 (4.5B MoE) [Wu et al., 2024b], and the model-scale sweep covers Qwen3-VL-Instruct at 2B/4B/8B/30B. Visual grounding uses a frozen Segment-Anything Model (SAM) [Kirillov et al., 2023] with labels from a lightweight captioner; the same backbone serves as the consensus/diversity judge (details in Appendix C).

Inference procedure. Given prompt p and image(s) $\mathcal{I}$, SAP keeps a population of $\mu + \lambda$ principles. Each principle x_i generates τ reasoning routes in one forward call; fitness follows Eq. (3); the population evolves by $(\mu + \lambda)$ selection. After the last generation an optional single aggregation call synthesizes the final answer.

Hyperparameters. $\mu = \lambda = 2$, $\tau = 2$, $T = 2$ generations, fitness weights $w_c = w_d = w_e = w_u = 1$, fixed across all 16 benchmarks. The number of grounded objects per image is capped; all discretization thresholds are fixed.

Evaluation. Each benchmark uses its official protocol (exact match for free-form, label mapping for multiple choice). *Average performance* (avg. perf.) is the mean across the 16 benchmarks excluding MME, whose scale differs.

Baselines. **(i)** *Instruct*: direct answer from $(\mathcal{I}, p)$; **(ii)** *Thinking*: long chain-of-thought (LongCoT) with explicit reasoning traces; **(iii)** *SAP* (ours). All share the same checkpoint and decoding settings. To isolate principle search from the grounding signal itself, we additionally compare against *SAM-Augmented LongCoT* (with and without our advisory prompt) and representative inference-time search / visually-grounded baselines: Self-Consistency [Wang et al., 2023b], Tree of Thoughts [Yao et al., 2023], Multimodal-CoT [Zhang et al., 2024b], VGR [Wang et al., 2026b], and TreeVGR [Wang et al., 2026a]; see Appendix B.3 and B.9.

4.1 Comparison

Tab. 1 compares SAP with training-free decoding baselines under a unified Qwen3-VL-8B evaluation stack. SAP obtains the strongest POPE precision and absent-object recall, and improves HallusionBench and AMBER-d while keeping POPE accuracy and F1 close to the Instruct backbone. This pattern supports the targeted claim that SAP reduces unsupported object assertions rather than uniformly improving every VLM capability. Table 2 reports the full 16-benchmark comparison across perception, reasoning, OCR, document, diagram, and real-world tasks. The averages in the

subsequent ablations summarize their corresponding expanded appendix experiments; they are not averages over the columns of either main table.

Table 1: Object-hallucination results under a unified Qwen3-VL-8B stack. All methods use the same benchmark items, evaluator interfaces, and decoding budget. All comparison methods are training-free decoding baselines; error bars are shown for Instruct and SAP over three seeds. R' denotes recall on absent-object queries.

method	POPE				HallusionBench	AMBER-d
	precision	R'	accuracy	F1	score	score
Instruct	89.6±2.2	90.0±2.4	**88.0±1.9**	**87.8**	72.7±2.5	88.7±2.5
MFCD [Liu et al., 2025]	87.4	87.3	87.7	87.7	72.0	87.3
M3ID [Favero et al., 2024]	81.5	80.0	84.0	84.6	65.3	83.3
ICD [Wang et al., 2024]	79.0	76.7	82.3	83.3	64.3	64.0
MaskCD [Deng and Yang, 2025]	86.3	86.7	85.3	85.1	73.3	78.7
SAP (ours)	**91.2±1.3**	**92.0±1.2**	87.6±1.1	87.0±1.1	**80.6±1.3**	**92.4±1.9**

Table 2: Comparison of longer and wider reasoning across 16 vision–language benchmarks under the same Qwen3-VL-8B backbone and inference-time decoding budget. Set I emphasizes perception and reasoning; Set II covers specialized visual inputs. Thinking denotes sequential LongCoT, whereas SAP scales over parallel routes. Shaded cells identify the entries included in each method's average; colors distinguish Instruct, Thinking, and SAP, while the differently scaled MME tasks are excluded. Δ is SAP minus Instruct.

benchmarks (↑)	Instruct	Thinking	**SAP (ours)**	Δ
avg. except MME	75.4	75.9	**76.6**	**+1.2**
Set I. Perception / Reasoning / General Knowledge				
POPE-recall [Li et al., 2023b]	83.9	79.6	**89.9**	+6.0
MME-p [Fu et al., 2025a]	1744.7	1663.2	**1755.5**	+10.8
MME-r [Fu et al., 2025a]	656.7	**722.5**	689.9	+33.2
MMBench [Liu et al., 2024a]	70.7	**87.0**	82.9	+12.2
SEEDBench [Li et al., 2023a]	68.5	**78.3**	77.3	+8.8
MMMU [Yue et al., 2024]	**62.4**	62.3	62.3	–0.1
Set II. Science / Text / OCR / Document / Diagram / Real-world				
ScienceQA [Lu et al., 2022]	88.0	**92.5**	92.2	+4.2
TextVQA [Singh et al., 2019]	**82.3**	77.2	81.3	–1.0
OCRVQA [Mishra et al., 2019]	**63.2**	44.1	62.8	–0.4
OCRBench [Liu et al., 2024b]	56.8	**65.9**	60.2	+3.4
ChartQA [Masry et al., 2022]	**82.7**	77.2	78.1	–4.6
DocVQA [Mathew et al., 2021]	**92.2**	91.6	90.6	–1.6
InfoVQA [Mathew et al., 2022]	82.6	**84.1**	83.1	+0.5
AI2D [Kembhavi et al., 2016]	79.2	**82.2**	80.1	+0.9
BLINK [Fu et al., 2024]	**68.2**	60.3	62.6	–5.6
RealWorldQA [xAI, 2024]	70.3	72.8	**73.3**	+3.0

4.2 Ablation Study

All ablations are reported relative to the default SAP configuration and the corresponding *Instruct* and LongCoT *Thinking* baselines.

Principle gain vs. depth (Tab. 3a). The marginal of the (τ, L) grid in Appendix B.7 shows that the no-principle POPE collapses from 84.6 at $L/8$ to 78.9 at L=4096, while the with-principle score holds at 87.9—so the absolute POPE gap is largest at full length, the opposite of what instruction-following degradation would produce. Principles compensate most where visual decay is most severe. Means over 3 seeds; error bars in Appendix B.5. **Modules of SAP (Tab. 3b).** Adding routes without evolution does not improve upon Instruct—route diversity alone is not enough. Adding evolution without multi-route collapses into single-trajectory optimization and degrades performance. Only

Table 3: Five SAP ablations study principle depth, modules, architecture transfer, evolutionary hyperparameters, and model scale. Light gray marks panel references/readouts; Dark gray marks default full SAP (76.6). Averages are appendix-specific, not main-table averages.

(a) **Principle gain vs. chain length.** POPE gap widens with depth w/o vs. w/ principles.

		POPE		OCRVQA		MME-R	
Length	Tokens	w/o	w/	w/o	w/	w/o	w/
$L/8$	512	84.6	87.7	61.3	62.5	399.7	568.3
$L/4$	1024	84.6	88.4	62.0	62.6	533.6	628.5
$L/2$	2048	83.3	88.8	63.4	63.4	625.1	683.2
L	4096	78.9	**87.9**	62.9	63.7	643.3	674.5

(b) **Module ablation.** Both evolution and multi-route are necessary.

settings	#gen. evo.	#routes τ	decision	avg.
instruct	0	1	–	75.4
w/o evolution	0	τ	elite	73.4
w/o multi-route	T	1	elite	73.0
full SAP	T	τ	elite	**76.6**
w/ summarizer	T	τ	agg.	74.8

(c) **Cross-architecture (reasoning).**

	InternVL3.5		DeepSeek-VL2	
	base	+SAP	base	+SAP
MMMU	74.6	**75.3**	51.2	**56.7**
MMBench	83.4	**84.6**	79.5	**79.8**

(d) (μ, λ) **sweep**. Balanced exploration > maximizing survival.

λ	2	2	4	4	6
μ	2	4	2	4	6
avg.	76.6	74.8	**76.8**	76.5	**76.8**

(e) Qwen3-VL-Instruct, 2/4/8/30B. SAP improves all **scales**.

sizes	2B	4B	8B	30B
instruct	60.4	75.4	75.4	77.3
+SAP	66.6	75.6	76.6	78.1

the full configuration recovers the gain; replacing elite selection with naive aggregation undoes it, supporting the claim that explicit principle filtering—not route averaging—is the operative mechanism. **Fitness weights (Fig. 4).** Zeroing out any single fitness component drops performance, so the four ordinal signals encode necessary, non-redundant constraints. A smooth w_d/w_c interpolation (Appendix B.11) is flat within $\sim$1.5 points across the interior, so SAP keeps $w_c{=}w_d{=}w_e{=}w_u{=}1$ across all 16 benchmarks without per-dataset tuning. **Cross-architecture (Tab. 3c).** On the two non-Qwen backbones listed under *Models* above, SAP improves both on MMMU and MMBench, indicating that the gain is backbone-agnostic. **Evolutionary hyperparameters (Tab. 3d).** Within the explored range, increasing the elite size μ does not consistently help while increasing the offspring count λ produces small gains. This matches Theorem 3.1, which predicts improvement to scale with λ rather than μ. **Scales (Tab. 3e).** Sweeping Qwen3-VL-Instruct from 2B to 30B, SAP improves consistently; the largest absolute gain is observed on the smallest backbone—the regime where unconstrained generation is most prone to drift.

Robustness to grounding quality. Three controlled perturbations target the input, representation, and output of the grounding pipeline (Appendix B.8). SAP remains above the ungrounded Instruct baseline under substantial image noise, up to 70% feature-token drop, and even complete removal of the grounding summary. In contrast, directly appending clean grounding to LongCoT fails sharply. Thus, the gain does not require unusually favorable detections: the advisory interface and route-level cross-validation matter more than exposing the model to additional object tokens.

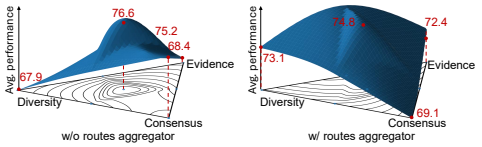


Figure 4: Fitness weights $w_d/w_c/w_e$ ($w_u{=}1$). Red values are appendix ablation averages, not main-table averages. Each term is necessary; the interior is stable.

Search controls and statistical stability. Under matched backbone and token budgets, SAP outperforms Self-Consistency, Tree of Thoughts, and Multimodal-CoT on the perception-sensitive comparison, while remaining competitive with trained visual-grounding methods (Appendix B.9). Three-seed deviations are smaller than the main inter-method gaps (Appendix B.10). On MathVista and MathVerse, longer SAP routes match or exceed LongCoT, showing that grounding-aware selection does not preclude reasoning depth (Appendix B.6). Finally, independent routes reduce wall-clock latency once parallel instances are available, although total token compute is unchanged (Appendix B.12). Together with the complete (τ, L) grid and smooth weight sweep, these controls rule out chain shortening, random variation, and a narrowly tuned operating point as explanations for the gain.

5 Related Works

Inference-time search for VL reasoning. For multimodal trajectories, Lin et al. [2025] sample and tree-search multimodal chain-of-thought with a consistency verifier and Wang et al. [2025b] train a vision value model that biases best-first decoding toward visually grounded continuations; for general prompts and programs, Kojima et al. [2022], Pryzant et al. [2023], Wu et al. [2024a], van Stein and Bäck [2024] evolve a single prompt or LLM-generated program offline on a curated set and reuse it at every query. **Main difference:** all of these either deepen one trajectory or commit to one fixed artifact. SAP runs evolution *per query at inference time* over multiple short principle-level routes launched in parallel from the original image and prompt. Each generation evaluates the resulting routes against the same frozen grounding reference and feeds that evidence into principle selection.

Tool-augmented grounded VL agents (the regime we target). Wang et al. [2026b] fine-tunes a VLM on (region, reasoning) pairs so it emits bounding boxes the next step replays; Wang et al. [2026a] extends this to RL that jointly supervises localization and reasoning; Zhao et al. [2024] feeds external object lists directly into decoding. **Main difference:** all three *inject* tool outputs into the VLM's trajectory (by training the VLM to follow them, or by hard-conditioning decoding on them), so a single noisy call can dominate. SAP instead uses a frozen grounding reference to evaluate routes and select principles. The tool influences which reasoning behavior survives, but it does not write the routes or their answers.

Training-free hallucination mitigation. Recent decoding methods suppress visually unsupported tokens by contrasting the original distribution with an image-perturbed or masked distribution [Liu et al., 2025, Deng and Yang, 2025], grounding generation in visual information [Favero et al., 2024], or contrasting instruction-conditioned predictions [Wang et al., 2024]. These methods intervene at token selection within one trajectory. SAP is complementary: it leaves token decoding unchanged and selects among complete routes induced by different reasoning principles. Consequently, its grounding signal can jointly assess object support, cross-route agreement, diversity, and uncertainty, while rejected routes do not contribute tokens to the surviving answer. It also permits black-box grounding without retraining the VLM.

6 Conclusions

Long serial reasoning can weaken visual support, while directly inserting tool outputs can turn a noisy detection into a persistent premise. SAP instead scales inference over compact reasoning principles and uses grounding only to select among independent routes. Across 16 benchmarks, it improves perception-sensitive and object-hallucination results without sacrificing broader reasoning capability; ablations, matched-budget baselines, controlled grounding perturbations, and cross-model transfer identify principle selection as the source of the gain. More broadly, tools can provide bounded evidence without becoming authorities over the answer. SAP still requires multiple model calls, benefits most from parallel hardware, and is evaluated with one grounding tool in single-pass question answering. Multiple tools and memory-aware selection for multi-turn agents are natural next steps.

Impact statement. Improving visual grounding may make VLM outputs more reliable, but SAP's multi-route inference requires more total model computation than a single short response and can therefore increase energy use and its associated environmental footprint. Parallel execution reduces wall-clock latency but does not eliminate this additional computation, so the impact depends on deployment scale, hardware efficiency, and the electricity source.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Science Foundation (NSF) under Grants IIS-2144209, IIS-2223769, CMMI-2411248, and IIBR-2601942; by the Office of Naval Research (ONR) under Grant N000142412636; and by grant support from Meta. Research was also supported by the NVIDIA Academic Grant Program using DGX Spark.

The authors declare no competing interests.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Bıyık, Hongxu Yin, Sifei Liu, and Xiaolong Wang. NaVILA: Legged robot vision-language-action model for navigation. In *Robotics: Science and Systems*, 2025. arXiv:2412.04453.

Jingyuan Deng and Yujiu Yang. MaskCD: Mitigating LVLM hallucinations by image head masked contrastive decoding. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 18854–18866. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.1025.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 11198–11201, 2024.

Alessandro Favero, Luca Zancato, Matthew Trager, Siddharth Choudhary, Pramuditha Perera, Alessandro Achille, Ashwin Swaminathan, and Stefano Soatto. Multi-modal hallucination control by visual information grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14303–14312, 2024.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. MME: A comprehensive evaluation benchmark for multimodal large language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025a. doi: 10.52202/085713-4899. NeurIPS 2025 Datasets and Benchmarks Track.

Ling Fu, Biao Yang, Zhebin Kuang, Jiajun Song, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, Mingxin Huang, et al. OCRBench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning. *arXiv preprint arXiv:2501.00321*, 2025b.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pages 148–166. Springer, 2024.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European conference on computer vision*, pages 235–251. Springer, 2016.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023a.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023b. arXiv:2305.10355.

Yujie Lin, Ante Wang, Moye Chen, Jingyao Liu, Hao Liu, Jinsong Su, and Xinyan Xiao. Investigating inference-time scaling for chain of multi-modal thought: A preliminary study. *arXiv preprint arXiv:2502.11514*, 2025.

Bingqian Liu, Fu Zhang, Guoqing Chen, and Jingwei Cheng. Multi-frequency contrastive decoding: Alleviating hallucinations for large vision-language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28568–28584. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1452.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer, 2024a.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024b.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. arXiv:2209.09513.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.02255.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pages 2263–2279, 2022.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2019.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. arXiv:2305.03495.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019.

Niki van Stein and Thomas Bäck. Llamea: A large language model evolutionary algorithm for automatically generating metaheuristics. *IEEE Transactions on Evolutionary Computation*, 2024.

Haochen Wang, Xiangtai Li, Zilong Huang, Anran Wang, Jiacong Wang, Tao Zhang, Jiani Zheng, Sule Bai, Zijian Kang, Jiashi Feng, Zhuochen Wang, and Zhaoxiang Zhang. Traceable evidence enhanced visual grounded reasoning: Evaluation and method. In *International Conference on Learning Representations (ICLR)*, 2026a. arXiv:2507.07999.

Jiacong Wang, Zijian Kang, Haochen Wang, Xiao Liang, Ya Wang, Jiawen Li, Bohong Wu, Jiao Ran, Haiyong Jiang, Chao Feng, and Jun Xiao. VGR: Visual grounded reasoning. In *International Conference on Learning Representations*, 2026b. arXiv:2506.11991.

Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Ming Yan, Ji Zhang, and Jitao Sang. Amber: An llm-free multi-dimensional benchmark for mllms hallucination evaluation. *arXiv preprint arXiv:2311.07397*, 2023a.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025a.

Xintong Wang, Jingheng Pan, Liang Ding, and Chris Biemann. Mitigating hallucinations in large vision-language models with instruction contrastive decoding. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

Xiyao Wang, Zhengyuan Yang, Linjie Li, Hongjin Lu, Yuancheng Xu, Chung-Ching Lin, Kevin Lin, Furong Huang, and Lijuan Wang. Scaling inference-time search with vision value model for improved visual comprehension. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1173–1184, 2025b. arXiv:2412.03704.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023b. arXiv:2203.11171.

Xingyu Wu, Sheng-hao Wu, Jibin Wu, Liang Feng, and Kay Chen Tan. Evolutionary computation in the era of large language model: Survey and roadmap. *IEEE Transactions on Evolutionary Computation*, 2024a.

Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, et al. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*, 2024b.

xAI. RealWorldQA. Dataset, `https://huggingface.co/datasets/xai-org/RealworldQA`, 2024. Released with Grok-1.5V.

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. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Peng Gao, and Hongsheng Li. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision (ECCV)*, 2024a. arXiv:2403.14624.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *Transactions on Machine Learning Research (TMLR)*, 2024b. arXiv:2302.00923.

Linxi Zhao, Yihe Deng, Weitong Zhang, and Quanquan Gu. Mitigating object hallucination in large vision-language models via image-grounded guidance. *arXiv preprint arXiv:2402.08680*, 2024.

Hanlin Zhu, Shibo Hao, Zhiting Hu, Jiantao Jiao, Stuart J. Russell, and Yuandong Tian. Reasoning by superposition: A theoretical perspective on chain of continuous thought. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2670. NeurIPS 2025 Main Conference; arXiv:2505.12514.

A Additional Discussion

Limitations. We note several limitations of the current approach for further research. SAP is most effective and responds quickly only in large-scale or parallel inference settings, where its multi-route design can amortize computational costs. In addition, its performance depends on the quality of the prompt templates of the evolutionary system and the principle of generation, which requires careful design. Future work may explore meta-level principles for guiding principle generation itself, enabling more systematic control over diversity and randomness in the principle space.

Licenses and asset use. We use Qwen3-VL under Apache-2.0, InternVL3.5 under MIT, DeepSeek-VL2 code under MIT and its model weights under the DeepSeek Model License, Segment Anything under Apache-2.0, and VLMEvalKit under Apache-2.0. Benchmark data are accessed through their original distributions or VLMEvalKit and are not redistributed; each benchmark remains subject to the license and terms provided by its original authors.

Visual modality diminishes in textual autoregressive generation. As illustrated in Fig. 1, during long vision–language reasoning, the contribution of visual information tends to diminish as generation progresses. While early reasoning steps may actively rely on visual evidence, later steps increasingly focus on the textual space. This phenomenon reflects a fundamental discrepancy between modalities: textual representations are explicitly generated and updated at every decoding step, whereas visual inputs are typically incorporated only indirectly through fixed encodings or limited cross-modal interactions. Once visual information is compressed into an early textual summary, any omissions or misinterpretations become difficult to correct in subsequent reasoning steps. As inference-time computation is scaled by increasing generation length, such early grounding errors are therefore amplified rather than mitigated. Beyond this modality imbalance, inference-time optimization in VLMs faces two intrinsic challenges.

Challenge I: inconsistent and ordinal signals. First, supervision signals in multimodal reasoning are inherently inconsistent and ordinal. Evaluations of vision–language reasoning outputs—whether provided by humans or models—are often based on heterogeneous and implicit criteria, particularly regarding how visual evidence should be used. For the same reasoning outcome, different evaluators may emphasize different aspects, such as visual grounding, logical completeness, or conciseness. Unlike language-only settings, where such inconsistency primarily affects stylistic preferences, in vision–language reasoning it directly influences whether and how visual evidence is incorporated. As a result, supervision does not define a single well-calibrated scalar objective. Instead, what is naturally available is an *ordinal signal*: a partial ordering that indicates which reasoning behaviors are preferable to others, rather than forcing heterogeneous judgments into a single quantitative score.

Challenge II: sparse and discrete search space. Second, the search space of inference-time reasoning is discrete and sparse. Both token-level reasoning trajectories and higher-level reasoning principles are discrete objects drawn from a combinatorial space without a smooth local structure. This challenge is further amplified in vision–language reasoning, where different reasoning behaviors may rely on different subsets of visual evidence, making local comparisons between candidates ill-defined. Consequently, optimization strategies that rely on continuity assumptions or local gradient information are poorly suited for this setting.

Taken together, these properties suggest that effective inference-time optimization in VLMs should (i) operate directly on discrete candidates, (ii) rely on relative comparisons rather than absolute scalar supervision, and (iii) remain robust under noisy and inconsistent feedback. These requirements motivate a population-based, selection-driven optimization paradigm for principle-level reasoning, which we adopt in our approach.

B Additional Empirical Analyses

B.1 POPE-Recall and POPE-Overall: A Mathematical Note

POPE consists of balanced positive and negative yes/no queries about object presence. The POPE-recall metric reported in Tab. 2 measures hit rate on the target class, which is the principal axis along

which object-hallucination is evaluated: a higher recall on this class corresponds to better grounding fidelity for the agent's affirmative claims. With balanced classes the overall accuracy decomposes as

$$A \;=\; \tfrac{1}{2}\big(R + R'\big),$$

where R is the reported POPE-recall and R' is the recall on the complementary class. SAP's improvement on R from 83.9 to 89.9 is a $\sim 37\%$ relative reduction in error on the targeted axis. The corresponding move on the overall accuracy A is small because R' absorbs part of the calibration shift; SAP optimizes for the grounding-fidelity axis captured by R, which is the property the abstract claim "reduces object hallucination" refers to. We report POPE-recall in the main comparison for this reason; the full per-class breakdown is omitted from the main table only for space and follows the same trend.

B.2 Training-Free Property and the Role of External Tools

SAP is training-free: it performs no gradient updates, requires no labeled data, and fine-tunes none of its components. The grounding service (e.g., SAM) and the captioner used to label its outputs are invoked strictly as frozen inference-time tools.

Because the language model never absorbs information from these tools into its parameters, the relationship between SAP and the external grounding service is not a form of model distillation: distillation would require the target model to internalize external knowledge into its weights via training. SAP instead queries the tool at inference time to obtain grounding signals on a per-input basis, which makes the tool consultation pattern closer to retrieval-augmented generation. The advisory treatment of tool outputs (Appendix B.8) further ensures that the language model overrides erroneous tool outputs rather than absorbing them.

B.3 SAM-Augmented LongCoT Baseline

To isolate the contribution of evolutionary principle search from the contribution of the SAM-derived grounding summary itself, we evaluate a controlled baseline in which the identical grounding summary used by SAP is fed to the standard LongCoT pipeline. We additionally evaluate a version that further inherits the prompt skeleton used by SAP (advisory phrasing, evidence-hierarchy block) but without principle-level evolution or multi-route inference.

Table 4: SAM-augmented LongCoT, w/wo advisory prompt. Qwen3-VL-8B; POPE/OCRVQA/MME-R. Advisory prompt prevents catastrophic over-trust.

Method	POPE (%)	OCRVQA (%)	MME-R
Instruct	83.9	63.2	656.7
Thinking (LongCoT)	79.6	44.1	722.5
SAM-Aug. LongCoT	14.8	45.2	328.4
SAM-Aug. LongCoT w/ SAP prompt	54.8	62.2	623.1
SAP (ours)	**89.9**	**62.8**	**689.9**

Two findings follow. First, naively injecting SAM grounding into LongCoT without an advisory prompt is catastrophically unstable on POPE (14.8): the model over-trusts the textual summary and propagates a single grounding error throughout the entire reasoning chain—a single-point failure of tool use. Second, adding the advisory prompt partially recovers performance (POPE 54.8) but still leaves a substantial gap to SAP (89.9). The remaining gain is therefore directly attributable to evolutionary principle search and multi-route inference, not to the grounding signal alone.

B.4 Smooth Sensitivity of Fitness Weights

The four-component fitness in Eq. (3) encodes two natural tensions—*diversity vs. consensus* (routes should be diverse yet convergent) and *evidence vs. uncertainty* (ground in visual facts, suppress uninformed guesses). Each term blocks a distinct failure mode. Figure 4 reports extreme configurations in which entire components are zeroed out; the resulting drops therefore measure necessity, not local sensitivity.

A complementary smooth interpolation between consensus and diversity (holding evidence fixed at $w_e = 0.5$ and enforcing $\sum w = 1$) is reported as panel (c) of Tab. 3 in the main text. Performance degrades only at the two extremes (a single dominant component); intermediate configurations remain stable within $\sim$1.5 points. Throughout the paper SAP uses $w_c = w_d = w_e = w_u = 1$ across all 16 benchmarks—no per-dataset weight search is performed.

B.5 Principle Adherence under Increasing Reasoning Depth

A natural concern is whether the language model continues to adhere to the high-level principle as the chain-of-thought grows longer. We argue this failure mode is structurally absent in SAP and confirm the argument empirically.

Architectural argument. Each reasoning route in SAP is an independent forward call with the principle injected at position 0 of a fresh context, never competing with accumulating reasoning text. Each route never exceeds the length of standard LongCoT generation ($L = 4096$ tokens); under the default configuration ($\tau = 2$, $L/2 = 2048$ tokens per route), each route operates well within the typical LongCoT range while the *total* budget is identical. The dilution failure mode—in which a principle injected once is gradually overshadowed—is a property of single-chain generation under fixed budget; SAP scales through width, not depth, and therefore does not exhibit it.

Empirical argument: principle gain grows with depth. If principle adherence degraded with chain length, the absolute POPE score with principles should *converge* to the no-principle baseline as the chain length approaches the LongCoT budget. The marginal distribution from the (τ, L) interaction grid (Appendix B.7) is reported as panel (a) of Tab. 3 in the main text and shows the opposite: at $L = 4096$ (matching standard LongCoT generation length), the no-principle POPE collapses to 78.9 while the with-principle score holds at 87.9, the largest absolute gap across all four depths. Principle gains grow precisely because visual decay is most severe at full length, and principles compensate most effectively there. The plateau in absolute POPE beyond $\tau = 4$ also reflects the finite diversity of effective principles in practice rather than instruction-following failure. The full version with $\pm$ standard deviation across 3 seeds is shown below; main-text Tab. 3a (a) reports means only for space.

Table 5: Principle gain vs. chain length: absolute scores w/o vs. w/ principles, with $\pm$ standard deviation across 3 seeds (attached to the with-principle column; the no-principle baseline is deterministic across seeds).

		POPE		OCRVQA		MME-R	
Length	Tokens	w/o	w/	w/o	w/	w/o	w/
$L/8$	512	84.6	87.7±1.70	61.3	62.5±0.87	399.7	568.3±59.8
$L/4$	1024	84.6	88.4±2.85	62.0	62.6±1.78	533.6	628.5±55.4
$L/2$	2048	83.3	88.8±2.36	63.4	63.4±1.60	625.1	683.2±36.2
L	4096	78.9	**87.9±1.61**	62.9	63.7±0.74	643.3	674.5±16.7

B.6 Performance on Reasoning-Heavy Benchmarks

Consensus matching could in principle amplify systematic errors on hard problems where the model rarely produces the correct answer. We address this concern in two ways.

Joint operation of fitness signals. Consensus is one of four signals; Evidence Validity and Uncertainty Penalty operate jointly with it, so SAP is not equivalent to Self-Consistency. A principle that induces a perceptual error (e.g., misidentifying a circle as a square) is penalized by Evidence Validity and down-weighted independently of majority status, while the Uncertainty Penalty further suppresses low-confidence routes. If a VLM fundamentally cannot recognize a basic shape, no aggregation strategy can recover—but joint filtering on evidence consistency and confidence minimizes error *propagation*.

Reasoning-heavy benchmarks. Beyond MMMU (already in the main table), we evaluate on MathVista [Lu et al., 2024] and MathVerse [Zhang et al., 2024a].

Table 6: Qwen3-VL-8B on MathVista/MathVerse. SAP scales to reasoning-heavy tasks at longer routes.

Method	MathVista (%)	MathVerse (%)
Instruct	77.1	62.4
Thinking (LongCoT)	81.1	77.8
SAP, 8 routes, $1/8$ length each	78.4	70.5
SAP, 8 routes, $1/4$ length each	81.7	75.8
SAP, 8 routes, $1/2$ length each	**82.0**	**78.8**

SAP targets vision-sensitive reasoning (object hallucination), not deductive mathematical reasoning where visual inputs are typically sharp and the bottleneck is reasoning depth rather than perceptual grounding. The transferability shown above is therefore a positive but secondary result.

B.7 Disentangling Route Width and Chain Length

To rule out the possibility that SAP's gains arise from chain-shortening or route-count alone, we report the complete 15-configuration (τ, L) interaction grid below, with and without saliency-aware principles, under the same total compute envelope.

Table 7: (τ, L) grid, w/wo principles. Qwen3-VL-8B; POPE/OCRVQA/MME-R. Principles drive the gain at every cell.

τ	L	POPE w/o	POPE w/	Δ POPE	OCR w/o	OCR w/	Δ OCR	MME-R w/o	MME-R w/	Δ MME-R
1	$L/8$	85.1	85.8	$+0.7$	61.2	61.1	-0.1	398.2	482.3	$+84.1$
1	$L/4$	86.2	86.0	-0.2	62.2	60.4	-1.8	554.1	576.1	$+22.0$
1	$L/2$	83.9	86.1	$+2.2$	63.2	61.4	-1.8	656.7	668.2	$+11.5$
1	L	76.3	85.6	$+9.3$	62.6	62.5	-0.1	643.1	652.4	$+9.3$
2	$L/8$	83.9	88.2	$+4.3$	60.6	62.4	$+1.8$	389.2	598.3	$+209.1$
2	$L/4$	84.2	89.6	$+5.4$	61.1	62.1	$+1.0$	524.7	631.2	$+106.5$
2	$L/2$	83.2	89.9	$+6.7$	62.9	62.8	-0.1	589.5	689.2	$+99.7$
2	L	80.1	86.9	$+6.8$	63.1	63.8	$+0.7$	642.6	692.4	$+49.8$
4	$L/8$	84.8	89.1	$+4.3$	62.2	63.9	$+1.7$	411.8	624.4	$+212.6$
4	$L/4$	83.5	89.7	$+6.2$	62.7	65.2	$+2.5$	521.9	678.1	$+156.2$
4	$L/2$	82.8	90.4	$+7.6$	64.0	66.1	$+2.1$	629.2	692.3	$+63.1$
4	L	80.4	91.1	$+10.7$	63.1	64.8	$+1.7$	644.3	678.8	$+34.5$
8	$L/4$	84.3	86.8	$+2.5$	58.2	63.3	$+5.1$	568.4	631.7	$+63.3$
8	$L/2$	83.2	89.6	$+6.4$	62.7	65.9	$+3.2$	624.4	689.7	$+65.3$
8	L	80.2	90.6	$+10.4$	63.0	64.9	$+1.9$	642.8	682.6	$+39.8$

Principles contribute positively in 13/15 cells for POPE and in all 15 cells for MME-R, at every route count and every chain length. At fixed $\tau = 2$, ΔPOPE ranges from $+4.3$ to $+6.8$ across the full depth range—chain shortening alone cannot explain a gain that is consistent across depths. At fixed $L/2$, ΔMME-R ranges from $+11.5$ to $+63.1$—route count alone cannot explain a gain that only appears when principles are present. The two negative deltas occur exclusively at $\tau = 1$, where a single principle constrains visual attention without cross-route diversity; these vanish at $\tau \geq 2$ and represent an incomplete configuration rather than a method failure.

B.8 Grounding Robustness under Controlled Perturbations

To probe whether SAP collapses when the SAM grounding pipeline is perturbed, we run three controlled perturbations spanning the full pipeline: (i) input-level Gaussian noise on the raw image, (ii) feature-level token drop on the grounded representations, and (iii) output-level item drop on the grounding summary fed to the LLM.

Table 8: SAM input-image noise. Qwen3-VL-8B; POPE/OCRVQA/MME-R. SAP stays above Instruct up to σ=1.0.

Noise σ	POPE	OCRVQA	MME-R
0.0 (clean)	89.9	62.8	689.9
0.1	89.5	61.2	671.8
0.3	88.1	60.4	668.4
0.5	87.8	61.2	661.2
1.0	84.4	58.3	608.8
Instruct (no grounding)	83.9	63.2	656.7
SAM-Aug. LongCoT	14.8	45.2	328.4
SAM-Aug. + SAP prompt	54.8	62.2	623.1

Table 9: SAM feature-token drop. Qwen3-VL-8B; POPE/OCRVQA/MME-R. Robust to $\leq 70\%$ drop.

Drop ratio	POPE	OCRVQA	MME-R
0% (clean)	89.9	62.8	689.9
10%	89.5	61.9	683.3
30%	88.8	62.4	654.4
50%	86.2	61.1	621.4
70%	85.6	55.8	594.7
99.9%	81.4	48.2	568.4
Instruct (no grounding)	83.9	63.2	656.7
SAM-Aug. LongCoT	14.8	45.2	328.4
SAM-Aug. + SAP prompt	54.8	62.2	623.1

SAP remains above Instruct across all moderate perturbation levels (input $\sigma \leq 1.0$, feature drop $\leq 70\%$, output drop $\leq 100\%$). The single exception is feature noise at 99.9% (POPE 81.4), where almost-fully-randomised grounding occasionally produces locally plausible but misleading signals that partially bypass the advisory mechanism. The absence of grounding (output 100%, POPE 83.8) is therefore *safer* than fully randomised grounding (feature 99.9%, POPE 81.4). The SAM-Augmented LongCoT baseline (POPE 14.8) confirms that catastrophic failure is not caused by grounding quality—clean grounding without the advisory prompt is already catastrophic. SAP's robustness derives from the advisory design and multi-route cross-validation operating jointly, not from grounding signal quality alone.

B.9 Comparison with Inference-Time Search Baselines

Beyond Instruct/Thinking, we compare against representative inference-time search and visual-grounding baselines under identical token budgets and the same backbone (Qwen3-VL-8B). Two of the baselines (VGR, TreeVGR) involve additional training and are listed for context only.

SAP outperforms sampling-based and trajectory-level search baselines on perception-sensitive tasks, and is competitive with or better than the training-based visual-grounding baselines on POPE while remaining strictly training-free. This supports the claim that principle-level search with visual grounding consistency as the fitness signal addresses the modality imbalance more effectively than search procedures designed for general (text-only) reasoning.

B.10 Variance Reports across Multiple Seeds

All SAP-side reproduced numbers in Table 11 are accompanied by error bars from three independent runs. Third-party reported numbers do not provide error bars and therefore appear without uncertainty intervals. SAP's standard deviation on POPE is ± 2.4, on OCRVQA is ± 1.2, on MME-R is ± 10.8—well below the inter-method gaps reported in the main comparison.

Table 10: Grounding-summary item drop. Qwen3-VL-8B; POPE/OCRVQA/MME-R. Robust even at full drop.

Drop rate	POPE	OCRVQA	MME-R
0% (clean)	89.9	62.8	689.9
25%	90.2	61.7	669.4
50%	87.7	59.4	654.4
75%	87.7	56.3	634.2
100% (no grounding)	83.8	62.6	638.4
Instruct (no grounding)	83.9	63.2	656.7
SAM-Aug. LongCoT	14.8	45.2	328.4
SAM-Aug. + SAP prompt	54.8	62.2	623.1

Table 11: Inference-time search and visual-grounding baselines. Qwen3-VL-8B, 3 seeds; POPE/OCRVQA/MME-R. SAP leads while staying training-free.

Category	Method	POPE (%)	OCRVQA (%)	MME-R
Baseline	Instruct	83.9	63.2	656.7
Sequential scaling	Thinking (LongCoT)	79.6	44.1	722.5
Sampling	Self-Consistency [Wang et al., 2023b]	73.5 ± 3.4	57.8 ± 2.1	642.1 ± 32.1
Trajectory search	Tree of Thoughts [Yao et al., 2023]	–	62.2	618.2
Visual decomp.	Multimodal-CoT [Zhang et al., 2024b]	82.5	58.4	426.7
Vis. ground. (tr.)	VGR [Wang et al., 2026b]	87.4	61.2	710.2
Vis. ground. (tr.)	TreeVGR [Wang et al., 2026a]	87.3	63.2	620.1
Principle search (ours)	**SAP, training-free**	$\mathbf{89.9\pm2.4}$	$\mathbf{62.8\pm1.2}$	$\mathbf{689.9\pm10.8}$

B.11 Smooth Sweep of Fitness Weights

The four ordinal fitness components $\{c, d, e, u\}$ are combined linearly with non-negative weights. To check that the choice $w_c{=}w_d{=}w_e{=}w_u{=}1$ in the main paper is not on a sharp ridge, we sweep the trade-off between consensus weight w_c and within-principle diversity weight w_d along a 1D simplex (with w_e fixed at 0.5). The 16-benchmark average varies within $\sim$2 points across the interior, confirming that the operating point is robust and motivating the dataset-independent default.

Table 12: Smooth w_d/w_c sweep on Qwen3-VL-8B with $w_e{=}0.5$; 16-benchmark average. Interior values are flat.

w_d	0.0	0.1	0.2	0.3	0.4	0.5
w_c	0.5	0.4	0.3	0.2	0.1	0.0
avg.	73.0	73.6	74.8	75.4	74.6	**75.8**

B.12 Response Time on Parallel Hardware

The accompanying scatter plot reports wall-clock latency against task performance under varying numbers of parallel model instances. SAP decomposes inference into T rounds of short-route generation and principle updates ($T{=}2$ by default); all routes within a round are independent and execute in parallel. Beyond a single instance, this design avoids the long token-level dependencies and intermediate-context synchronization required by LongCoT, lowering wall-clock latency while keeping total token count and total compute identical.

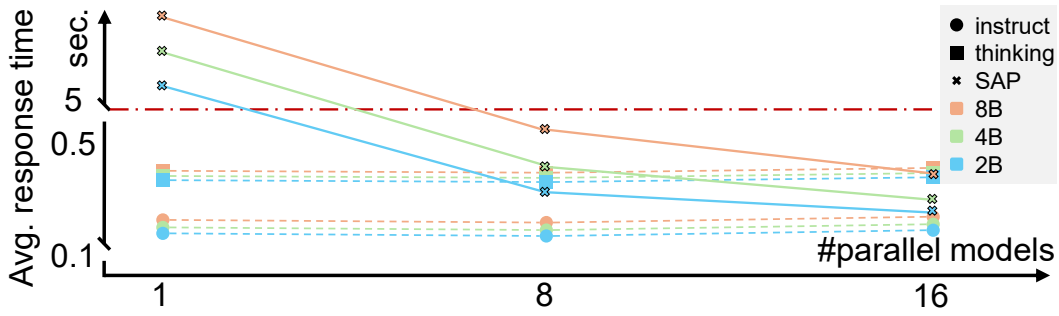


Figure 5: SAP supports inference-time speedup via parallelism.

C Implementation Details & Code Report: Saliency-Aware Principle Selection

SAP Overview. Given a question p and one or multiple images $\mathcal{I} = \{I_k\}_{k=1}^{K}$, SAP performs (i) grounding, (ii) principle generation and evolutionary selection, and (iii) final aggregation. Throughout the process, the LLM only observes the *original images* and a compact *textual grounding summary*; object lines and crops are used only for fitness evaluation and are never fed back to the LLM. All assessment signals are discretized into three levels: `low/medium/high`.

Step 1: Multi-image grounding. For each image I_k, SAP calls a grounding service (e.g., SAM-based AMG) to obtain a set of grounded objects $\mathcal{O}_k = \{o_{k,j}\}_{j=1}^{M_k}$, each with a label $\ell_{k,j}$ (obtained via an external captioner/labeler), and optional mask/bbox/score metadata. SAP converts $\{\mathcal{O}_k\}_{k=1}^{K}$ into a compact grounding summary string g that only describes counts and image sizes (optionally plus other lightweight statistics). SAP also builds a label map so that any object reference can be canonicalized as `img#k_obj#j(`$\ell_{k,j}$`)`.

Step 2: Initialize a principle population. SAP samples an initial population of principles $\{x_i\}_{i=1}^{\mu+\lambda}$ by a single LLM call conditioned on $(p, g, \mathcal{I})$. Each x_i is a short textual instruction that specifies a reasoning strategy (e.g., "verify with grounded objects before answering").

Step 3: One-call multi-route reasoning per principle. For each generation, SAP evaluates each principle x_i with exactly one LLM call. The LLM is provided with $(p, g, \mathcal{I}, x_i)$ and is asked to internally generate τ distinct reasoning routes under the same principle:

$$\{(r_{i,j}, s_{i,j})\}_{j=1}^{\tau}.$$

Each route produces a candidate `final_answer`, a short list of reasons, an uncertainty level $u_{i,j} \in \{\texttt{low}, \texttt{medium}, \texttt{high}\}$, and a list of evidence references. The LLM also outputs a within-principle diversity level $d_i \in \{\texttt{low}, \texttt{medium}, \texttt{high}\}$ summarizing how different the τ routes are.

Step 4: Representative answer and global consensus. SAP selects a representative route per principle (typically preferring lower uncertainty and non-empty answers) and obtains a representative answer $y_i^\star$. SAP then computes a global consensus answer $\bar{y}$ by majority vote over $\{y_i^\star\}$ and converts the winning fraction into a discrete consensus level $c \in \{\texttt{low}, \texttt{medium}, \texttt{high}\}$. The semantic-equivalence judgment $\sim_\varphi$ used below is performed by the same VLM backbone in zero-shot judge mode (Qwen3-VL-8B in our default configuration), consistent with treating the lightweight external judge of Eq. (3) as a separate inference-time call rather than a separately trained model. In addition, SAP assigns each principle a discrete match-to-consensus level

$$c_i = \begin{cases} \texttt{high}, & y_i^\star = \bar{y}, \\ \texttt{medium}, & y_i^* \sim_\varphi \bar{y} \\ \texttt{low}, & \text{otherwise.} \end{cases}$$

This design enforces that principles are rewarded only if their representatives' answers align with the population-level agreement. $y^* \sim_\varphi \bar{y}$ means y^* is judged as the same meaning of $\bar{y}$ by language model φ.

Step 5: Evidence validation and discretization. SAP canonicalizes each cited evidence item into the format `img#k_obj#j(`$\ell_{k,j}$`)` using the label map. It then evaluates whether cited object indices are valid grounding outputs (valid vs. invalid references). The ratio of valid references is discretized into an evidence level $e_i \in \{\texttt{low}, \texttt{medium}, \texttt{high}\}$, with empty evidence treated as `low`. This evidence level is used as a lightweight grounding-consistency signal; it never requires the LLM to see object crops.

Step 6: Discrete-level fitness and selection. For each principle x_i, SAP computes a scalar fitness score from four discrete signals: consensus match c_i, within-principle diversity d_i, representative uncertainty u_i (from the representative route), and evidence level e_i. Internally, SAP maps `low/medium/high` to ordinal scores and combines them with integer weights:

$$f_i \leftarrow w_c \,\mathrm{Score}(c_i) + w_d \,\mathrm{Score}(d_i) + w_e \,\mathrm{Score}(e_i) - w_u \,\mathrm{Pen}(u_i),$$

where $\mathrm{Score}(\texttt{low}, \texttt{medium}, \texttt{high}) = (0, 1, 2)$ and $\mathrm{Pen}(\texttt{low}, \texttt{medium}, \texttt{high}) = (0, 1, 2)$. SAP selects the top-$\mu$ principles as elites and (if more generations remain) asks the LLM to propose λ new principles conditioned on the elites and their fitness, forming the next $(\mu + \lambda)$ population.

Step 7: Final aggregation. After the last generation, SAP produces the final answer by aggregating the evaluated principles. Optionally, it issues one final LLM call conditioned on (p, g) and the list of representative summaries to synthesize a final answer with brief reasons and a discrete uncertainty level. If the aggregator fails, SAP falls back to the representative answer from the global-consensus group with conservative uncertainty.

C.1 Framework: Principle Evolutionary Reasoning

Building on the above formulation, SAP approximates the principle-level optimization objective through population-based evolutionary search. This procedure scales inference by iteratively improving a population of reasoning principles with respect to the utility function f.

The overall procedure consists of initialization and iterative evolution. As shown in Alg. 1, an initial population of principles $\{x_i\}_{i\in[\mu+\lambda]}$ is sampled conditioned on the prompt p. For each principle x_i, the language model generates multiple reasoning traces and summaries,

$$(r_{i,j}, s_{i,j}) \sim \varphi(r, s \mid x_i, p), \quad j \in [\tau],$$

which instantiate the reasoning route $\rho(x_i)$ induced by the principle.

Given these outputs, each principle is evaluated by a task-specific utility metric, producing a fitness score

$$f_i = f(\{r_{i,j}, s_{i,j}\}),$$

which estimates the effectiveness of the corresponding reasoning route. Based on these fitness scores, SAP performs population evolution as described in Alg. 2. Specifically, a $(\mu + \lambda)$ selection scheme is adopted: the top-μ principles

$$\{x_m^\mu\}_{m\in[\mu]} = \mathrm{Top}_\mu\big(\{(x_i, f_i)\}_{i\in[\mu+\lambda]}\big)$$

are retained as elites, while a new set of λ principles is generated by sampling conditioned on the elites,

$$x_n^\lambda \sim \varphi(x \mid \{x_m^\mu\}_{m\in[\mu]}, p), \quad n \in [\lambda].$$

The elites and newly generated principles are combined to form the updated population

$$\{x_i^+\}_{i\in[\mu+\lambda]} = \{x_m^\mu\}_{m\in[\mu]} \cup \{x_n^\lambda\}_{n\in[\lambda]}.$$

By iteratively applying population evolution, SAP provides a population-based approximation to the optimization problem in Eq. (2), enabling robust inference-time scaling under discrete generation and noisy feedback.

D Related Benchmarks

Benchmarks. We evaluate SAP on a diverse set of vision–language benchmarks spanning robustness, general reasoning, scientific understanding, OCR-centric reasoning, document analysis, and real-world generalization.

POPE Li et al. [2023b] evaluates object hallucination and existence verification by constructing balanced positive and negative samples. It directly measures whether models can correctly ground answers in visual evidence rather than prior bias.

MME Fu et al. [2025a] assesses multimodal robustness by separating perception and reasoning abilities. The reasoning subset emphasizes logical inference beyond direct visual recognition, making it suitable for evaluating inference-time reasoning strategies.

MMBench Liu et al. [2024a] is a large-scale multiple-choice benchmark for general multimodal understanding, covering object recognition, attribute reasoning, spatial relations, and commonsense inference. It serves as a standard indicator of overall VLM capability.

SEEDBench Li et al. [2023a] evaluates general visual reasoning and understanding across perception, logic, and commonsense tasks, emphasizing consistency across heterogeneous question types.

MMMU Yue et al. [2024] targets college-level multimodal reasoning in mathematics, physics, chemistry, and engineering. Many questions involve diagrams or multiple images, requiring structured reasoning and cross-image integration.

ScienceQA Lu et al. [2022] evaluates scientific question answering with visual contexts, combining diagrams with domain knowledge and multi-step reasoning.

TextVQA Singh et al. [2019] measures scene text understanding, where correct answers depend on reading and reasoning over text embedded in natural images.

OCRVQA Mishra et al. [2019] and **OCRBench** Liu et al. [2024b], Fu et al. [2025b] further stress OCR-centric reasoning under complex layouts and challenging visual conditions, focusing on accuracy and robustness of text grounding.

ChartQA Masry et al. [2022] evaluates reasoning over charts and plots, requiring numerical comparison and interpretation grounded in visualized data.

DocVQA Mathew et al. [2021] and **InfoVQA** Mathew et al. [2022] focus on document understanding, where models must reason over scanned documents, tables, and forms with dense textual content.

AI2D Kembhavi et al. [2016] tests diagram understanding in science education scenarios, emphasizing relational and structural reasoning over visual components.

BLINK Fu et al. [2024] evaluates multimodal entity linking, requiring alignment between visual entities and background knowledge.

RealWorldQA assesses open-world robustness using real-life images and unconstrained questions, testing generalization beyond curated benchmarks xAI [2024]. `https://huggingface.co/datasets/visheratin/realworldqa`

E Algorithm Analysis

E.1 Notation and Setup

Let $\mathcal{X}$ denote the (discrete) space of reasoning principles. The $(\mu + \lambda)$ evolutionary procedure maintains a population

$$P_t = \{x_i^{(t)}\}_{i=1}^{N}, \qquad N = \mu + \lambda,$$

at iteration $t \in \{0, 1, \ldots, T\}$. Each principle $x \in \mathcal{X}$ induces a set of reasoning outcomes (trajectories and summaries) via one model call, and is assigned a *fitness* value $f(x) \in \mathbb{R}$ through the discrete assessment and ordinal mapping described in Eq. (3). Define the best fitness in the population at iteration t as

$$F_t = \max_{x \in P_t} f(x).$$

Elites and offspring. At iteration t, the elite set $E_t \subseteq P_t$ consists of the top-μ principles in P_t ranked by fitness. The offspring multiset $O_t = \{X_n^{(t)}\}_{n=1}^{\lambda}$ is generated by sampling from a conditional distribution induced by the LLM-based proposer:

$$X_n^{(t)} \sim \varphi_x(\cdot \mid p, E_t), \qquad n \in [\lambda],$$

and the next population is formed as

$$P_{t+1} = E_t \cup O_t.$$

We emphasize that the discrete evaluation in SAP is used only to produce ordinal levels and an internal ranking/selection signal; no continuous confidence calibration is assumed.

E.2 Assumptions on Sampling and Dependence

Assumption E.1 (Conditional i.i.d. offspring sampling)**.** Conditioned on the history $\mathcal{H}_t$ up to iteration t (which includes p, all past populations, and thus E_t), the offspring samples $X_1^{(t)}, \ldots, X_\lambda^{(t)}$ are conditionally independent and identically distributed according to $\varphi_x(\cdot \mid p, E_t)$.

Assumption E.1 is satisfied when each offspring is generated by an independent call to the proposer (or independent decoding randomness) conditioned on the same (p, E_t). We use Assumption E.1 only to derive the search-space *coverage* bound (§below): once the elite set E_t has stabilized in the late iterations of the loop, successive offspring populations sample from a near-stationary conditional distribution $\varphi_x(\cdot \mid p, E_t)$, and the resulting offspring are approximately i.i.d. at the level of *semantic principle classes* (the granularity at which the coverage bound is stated), even though their token-level realizations are produced by autoregressive decoding. If implementation introduces stronger dependence, the probability bounds below can be extended under standard negative dependence conditions; we state the i.i.d. form for clarity.

E.3 Optimization Guarantees

E.3.1 Monotone best-fitness (no regression)

Theorem E.2 (Monotone best-fitness under $(\mu + \lambda)$ selection)**.** *For all iterations $t \geq 0$, the best fitness is non-decreasing:*

$$F_{t+1} \geq F_t.$$

Proof. By definition, E_t contains the top-μ elements of P_t ordered by $f(\cdot)$, and therefore contains at least one maximizer of f over P_t. Hence

$$\max_{x \in E_t} f(x) = \max_{x \in P_t} f(x) = F_t.$$

Since $P_{t+1} = E_t \cup O_t$ and $E_t \subseteq P_{t+1}$,

$$F_{t+1} = \max_{x \in P_{t+1}} f(x) \geq \max_{x \in E_t} f(x) = F_t.$$

□

E.3.2 One-step improvement probability

While Theorem E.2 shows stability (no regression), the *optimization speed* is governed by how likely the procedure produces strictly better principles.

Improvement set. Define the set of principles that strictly improve upon the current best:

$$\mathcal{A}_t = \{x \in \mathcal{X} : \ f(x) > F_t\}.$$

Assumption E.3 (Per-iteration improvement mass)**.** There exists $q_t \in [0, 1]$ such that conditioned on $\mathcal{H}_t$,

$$\Pr\Big(X_n^{(t)} \in \mathcal{A}_t \mid \mathcal{H}_t\Big) \geq q_t \quad \text{for each } n \in [\lambda].$$

Assumption E.3 does not require knowing $\mathcal{A}_t$ explicitly; it only postulates that the proposer distribution retains at least q_t probability mass on strictly improving principles.

Theorem E.4 (One-step improvement bound)**.** *Under Assumptions E.1 and E.3, the probability of a strict improvement at iteration t satisfies*

$$\Pr(F_{t+1} > F_t \mid \mathcal{H}_t) \geq 1 - (1 - q_t)^\lambda.$$

Consequently,

$$\Pr(F_{t+1} > F_t) \geq \mathbb{E}\big[\, 1 - (1 - q_t)^\lambda \,\big].$$

Proof. A strict improvement occurs if at least one offspring lies in $\mathcal{A}_t$:

$$\{F_{t+1} > F_t\} \supseteq \bigcup_{n=1}^{\lambda} \{X_n^{(t)} \in \mathcal{A}_t\}.$$

Equivalently, no improvement implies none of the offspring hits $\mathcal{A}_t$:

$$\{F_{t+1} = F_t\} \subseteq \bigcap_{n=1}^{\lambda} \{X_n^{(t)} \notin \mathcal{A}_t\}.$$

Conditioning on $\mathcal{H}_t$ and using conditional independence (Assumption E.1),

$$\begin{aligned}
\Pr(F_{t+1} = F_t \mid \mathcal{H}_t) &\leq \Pr\left(\bigcap_{n=1}^{\lambda} \{X_n^{(t)} \notin \mathcal{A}_t\} \,\middle|\, \mathcal{H}_t \right) \\
&= \prod_{n=1}^{\lambda} \Pr(X_n^{(t)} \notin \mathcal{A}_t \mid \mathcal{H}_t) \\
&\leq (1 - q_t)^\lambda,
\end{aligned}$$

where the last inequality follows from Assumption E.3. Taking complements gives

$$\Pr(F_{t+1} > F_t \mid \mathcal{H}_t) = 1 - \Pr(F_{t+1} = F_t \mid \mathcal{H}_t) \geq 1 - (1 - q_t)^\lambda.$$

Finally, removing the conditioning yields the stated unconditional bound by taking expectation over $\mathcal{H}_t$. □

Remark E.5 (Small-q_t regime and the role of λ). When q_t is small, the improvement probability satisfies

$$1 - (1 - q_t)^\lambda \approx \lambda q_t,$$

showing that increasing λ approximately linearly amplifies the chance of obtaining a strictly better principle in each iteration.

E.3.3 Multiple iterations: a (loose) bound on improvement within T steps

For completeness, we provide a standard union-style lower bound on obtaining at least one strict improvement over a horizon of T iterations.

Corollary E.6 (At least one improvement within T iterations (lower bound))**.** *Under Assumptions E.1 and E.3, and assuming $q_t \geq q_{\min}$ for $t = 0, \ldots, T-1$,*

$$\Pr(\exists\, t \in \{0, \ldots, T-1\} : F_{t+1} > F_t) \geq 1 - (1 - q_{\min})^{\lambda T}.$$

Proof. Condition on each iteration and apply Theorem E.4:

$$\Pr(F_{t+1} = F_t \mid \mathcal{H}_t) \leq (1 - q_t)^\lambda \leq (1 - q_{\min})^\lambda.$$

Assuming conditional independence across iterations is not necessary for an upper bound on the probability of *no improvements*: by iterated conditioning,

$$\Pr(\text{no improvements in } T \text{ iterations}) \leq \prod_{t=0}^{T-1} (1 - q_{\min})^\lambda = (1 - q_{\min})^{\lambda T}.$$

Taking complements yields the claim. □

The above corollary is intentionally conservative; it is included to illustrate scaling with λT and to align with coverage-style arguments below.

E.4 Generalization as Coverage in the Principle Space

E.4.1 Coverage model

We use a coverage-based notion of generalization: larger inference budgets explore more candidate principles, increasing the probability of encountering effective principles that induce reliable reasoning behaviors beyond the initial population.

Definition E.7 (Effective principle set)**.** Let $\mathcal{X}_{\text{good}} \subseteq \mathcal{X}$ denote the (unknown) set of effective principles under the task distribution of interest (e.g., principles that produce stable, grounded, and reliable outputs according to SAP's discrete assessments).

Assumption E.8 (Per-sample coverage lower bound)**.** There exists $q_{\text{good}} \in [0, 1]$ such that for every iteration t and every offspring sample $X_n^{(t)}$,

$$\Pr\Big(X_n^{(t)} \in \mathcal{X}_{\text{good}} \mid \mathcal{H}_t\Big) \geq q_{\text{good}}.$$

Assumption E.8 states that the proposer retains a nontrivial probability of sampling effective principles. This is the standard requirement for coverage guarantees in randomized search.

E.4.2 Discovery probability within T iterations

Proposition E.9 (Coverage increases with λT)**.** *Under Assumptions E.1 and E.8, the probability that at least one effective principle is sampled within T iterations is lower bounded by*

$$\Pr\Big(\exists\, t \in \{0, \ldots, T-1\},\ \exists\, n \in [\lambda] \text{ s.t. } X_n^{(t)} \in \mathcal{X}_{\text{good}}\Big) \geq 1 - (1 - q_{\text{good}})^{\lambda T}.$$

Proof. Define the event that no effective principle is sampled in iteration t:

$$\mathcal{E}_t := \bigcap_{n=1}^{\lambda} \{X_n^{(t)} \notin \mathcal{X}_{\text{good}}\}.$$

Conditioned on $\mathcal{H}_t$ and using conditional independence (Assumption E.1),

$$\Pr(\mathcal{E}_t \mid \mathcal{H}_t) = \prod_{n=1}^{\lambda} \Pr(X_n^{(t)} \notin \mathcal{X}_{\text{good}} \mid \mathcal{H}_t) \leq (1 - q_{\text{good}})^{\lambda},$$

where the inequality follows from Assumption E.8. Now consider the event of no discovery over T iterations: $\bigcap_{t=0}^{T-1} \mathcal{E}_t$. By iterated conditioning,

$$\Pr\left(\bigcap_{t=0}^{T-1} \mathcal{E}_t\right) = \mathbb{E}\left[\prod_{t=0}^{T-1} \Pr(\mathcal{E}_t \mid \mathcal{H}_t)\right] \leq \prod_{t=0}^{T-1} (1 - q_{\text{good}})^{\lambda} = (1 - q_{\text{good}})^{\lambda T}.$$

Taking complements yields

$$\Pr\left(\bigcup_{t=0}^{T-1} \mathcal{E}_t^c\right) \geq 1 - (1 - q_{\text{good}})^{\lambda T},$$

which is exactly the probability of sampling at least one effective principle within T iterations. □

Remark E.10 (Scaling in λ and T). The bound $1 - (1 - q_{\text{good}})^{\lambda T}$ is monotone increasing in both λ and T. For small q_{good}, it satisfies

$$1 - (1 - q_{\text{good}})^{\lambda T} \approx \lambda T q_{\text{good}},$$

making explicit that increasing the exploration width (λ) and/or the number of iterations (T) improves coverage of the principle space.

E.5 Connecting Optimization and Coverage to Inference-Time Scaling

The results above jointly support the inference-time scaling interpretation of SAP/PER: elite selection ensures optimization stability (Theorem E.2), while increasing inference budget via larger λ and/or T improves both the per-iteration chance of strict improvement (Theorem E.4) and the probability of discovering effective principles through coverage (Proposition E.9). These properties match the operational definition of scaling in our setting: allocating additional compute to explore and select among alternative reasoning principles and routes, rather than extending a single fixed trajectory.

F Template Design

System Prompt Template Design. A central objective of SAP is to prevent inference-time reasoning from drifting toward text-only heuristics and to ensure that reasoning remains grounded in visual evidence throughout long generation horizons. This objective is realized through a carefully designed family of user prompt templates that are shared across all stages of SAP, with stage-specific constraints. *Different VLMs should correspond to different templates.*

Motivation. We identify two recurring failure modes in long-horizon vision–language reasoning. First, visual information is typically provided only once at the beginning of generation, whereas textual reasoning states (e.g., early visual summaries) are updated autoregressively. As generation proceeds, reasoning therefore becomes increasingly text-dominated, allowing early visual grounding errors to compound over time. Second, auxiliary textual signals—such as grounding summaries or evaluator feedback—are often noisy, incomplete, or heuristic; yet, they are easily over-trusted by the model as factual evidence. Together, these effects cause models to rely excessively on intermediate text states instead of re-consulting the original image(s), leading to unstable inference-time scaling.

Design Principle. To address these issues, SAP enforces a strict evidence hierarchy in all user prompts. The original image(s) are explicitly designated as the primary and most reliable source of information. Textual grounding summaries are framed as optional references that may be missing, incomplete, or incorrect. Their role is limited to suggesting *where to look* in the image, rather than *what to believe*. Whenever textual cues conflict with visual observations, the model is instructed to re-examine the image(s) and respond conservatively if ambiguity remains. This principle is shared across all SAP stages to counteract text-dominated autoregressive drift.

Template Overview. SAP employs four user prompt templates corresponding to different inference-time stages: (*i*) principal initialization, (*ii*) principal evolution, (*iii*) principle-guided multi-route reasoning, and (*iv*) final answer aggregation. Despite their different roles, all templates explicitly reinforce visual dominance, uncertainty awareness, and conservative decision-making under noisy feedback.

(1) Principle Initialization Template. The goal of principle initialization is to generate a diverse set of high-level reasoning principles before producing concrete answers. The prompt instructs the model to focus on *how* to reason from visual evidence rather than encoding task-specific conclusions.

```
Generate several high-level reasoning principles for answering the visual question.
The image(s) are the primary source of truth.
The grounding summary is optional and may be incomplete or incorrect.
Each principle should describe a general way to reason from visual evidence,
not a specific answer.
Avoid assuming details that cannot be directly verified from the image(s).
```

This template encourages diversity in reasoning strategies while ensuring that all principles remain anchored to observable visual cues.

(2) Principle Evolution Template. During evolutionary refinement, principles are updated based on their observed utility. The evolution prompt presents prior principles and their relative performance while explicitly discouraging overfitting to noisy or short-term feedback.

```
Refine the reasoning principles for visual question answering.
The image(s) remain the most reliable evidence.
Fitness signals and summaries may be noisy and should be used cautiously.
Preserve strengths of effective principles, discard misleading ones,
```

```
and propose new principles that better encourage verification
against visual evidence.
Do not assume that high-scoring principles are always correct.
```

This design ensures that evolutionary updates favor principles that improve visual grounding rather than exploit accidental correlations in evaluator signals.

(3) Principle-Guided Multi-route Reasoning Template. Given a fixed principle, SAP requires the model to generate multiple distinct reasoning routes in parallel. Each route must independently justify its conclusion using visual evidence and explicitly report uncertainty when the evidence is weak.

```
Answer the visual question using the image(s)
and the single active reasoning principle below.
Generate multiple distinct reasoning routes under the same principle.
For each route:
- Base the reasoning on observable visual evidence.
- Cite visual cues when possible.
- Report uncertainty if evidence is weak or ambiguous.
The grounding summary is only a reference and may be incorrect.
If unsure, do not guess.
```

By construction, this template discourages blind reliance on early textual summaries and promotes repeated visual consultation across alternative reasoning routes.

(4) Final Aggregation Template. The final aggregation stage synthesizes candidate answers produced under different principles. The prompt emphasizes conservative selection based on visual support rather than textual agreement or majority voting.

```
Synthesize a final answer from multiple candidate answers.
Treat the image(s) as the authoritative evidence.
Candidate answers may be inconsistent or incorrect.
Prefer answers that are better supported by visual evidence.
Express uncertainty if no answer is clearly justified.
Do not introduce new assumptions beyond what can be verified
from the image(s).
```

Implicit Template Constraints. Beyond stage-specific instructions, SAP enforces several implicit constraints that are shared across all prompt templates. These constraints are implemented purely at the prompt level and shape inference-time behavior without introducing additional supervision.

Evidence Hierarchy Block. All SAP prompts explicitly prioritize visual evidence over any textual signals.

```
[Evidence Hierarchy]
- Image(s) are the most reliable evidence.
- Textual summaries and feedback are auxiliary and may be noisy.
- Text must never override visual observation.
- When uncertainty remains, respond conservatively.
```

Grounding-as-Reference Rule. Grounding summaries are treated strictly as weak references.

```
[Grounding Reference Rule]
- Grounding summaries may be incomplete or incorrect.
- Use them only to guide where to look in the image(s).
- Do not assume grounded attributes unless visually verified.
- If conflicts arise, trust the image(s).
```

Strict Output Rule. To support reliable evaluation and comparison, SAP enforces a strict structured-output constraint.

```
[Strict Output Rule]
- Follow the specified JSON schema exactly.
- Do not add extra fields or free-form explanations.
```

```
- Use predefined discrete levels for uncertainty and evaluation.
- If information is missing, return a valid placeholder.
```

Effect. Together, these prompt templates and shared constraints form a unified inference-time control mechanism that reallocates computation toward repeated validation against visual evidence. SAP introduces no new knowledge, external supervision, or task-specific heuristics. Instead, it reshapes how existing model capacity is used during inference, enabling stable and effective inference-time scaling under discrete generation and noisy multimodal feedback.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 list three concrete contributions — characterising text-dominated drift in long VLM reasoning, proposing the Saliency-Aware Principle Selection (SAP) framework, and demonstrating parallelizable inference-time scaling — all of which are directly supported by Section 4 (16-benchmark comparison) and Appendix B (interaction grid, robustness, baselines).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are explicitly discussed in Appendix A (More Discussion): SAP's amortized efficiency relies on parallel inference settings, performance depends on prompt-template quality for principle generation, and reasoning-intensive tasks (*e.g.*, MathVerse) require longer per-route budgets to fully match LongCoT (Appendix B.6). The grounding-tool dependency and its failure modes are quantified under controlled perturbation in Appendix B.8.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The single theoretical result (Theorem 3.1, on optimization stability and coverage of the $(\mu+\lambda)$ evolutionary procedure) is stated in Section 3 (Framework Analysis). All assumptions (Assumption E.1, E.3, E.8) are stated formally in Appendix E, and complete proofs of the monotone-fitness, one-step improvement, and coverage bounds appear in the same appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: SAP is training-free; all reproducibility information is therefore in the inference recipe. The full pipeline (grounding, principle initialization, multi-route inference, fitness, evolution, aggregation) is described step-by-step in Appendix C (Implementation), with the four prompt templates given verbatim in the Template Design appendix. Hyperparameters ($\mu = \lambda = 2$, $\tau = 2$, $T = 2$, w_c=w_d=w_e=w_u=1) and decoding settings are listed in the Experiment Setups subsection of Section 4. The 16 benchmarks are evaluated through the public VLMEvalKit harness with no per-dataset tuning.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the core logic of SAP (principle initialization, multi-route inference, fitness scoring, $(\mu + \lambda)$ evolution, aggregation) in the camera-ready supplementary material. Benchmark harness wrappers and dataset loaders are provided by the public VLMEvalKit package rather than duplicated. The implementation is intentionally lightweight: the algorithmic core is given step-by-step in Appendix C, the four prompt templates are reproduced verbatim in the Template Design appendix, and all hyperparameters in Section 4 are dataset-independent, so SAP can be applied to any frozen VLM through its standard inference API. Evaluation uses the public VLMEvalKit harness with no per-dataset tuning.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: SAP is training-free, so there are no optimizer/learning-rate choices to report. Test-time configuration — backbones (Qwen3-VL-8B and additional 2B/4B/30B/InternVL3.5-8B/DeepSeek-VL2 for ablations), evaluation splits via VLMEvalKit, decoding budget, and all SAP hyperparameters — is given in the Experiment Setups subsection of Section 5 and Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: SAP-side reproduced numbers in the inference-time-search comparison table (Appendix B.9) carry $\pm$ standard deviation across three independent runs (seeds 2025, 2026, 42). The marginal-distribution analyses in Appendix B.5 and Appendix B.7 also report $\pm$ values. Third-party reported numbers are listed without error bars because the original references do not provide them.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: SAP is training-free; all GPU work runs on a single NVIDIA A100. CPU resources come from a cloud-hosted virtual machine and the underlying physical CPU specifications are not exposed by the provider. Software environment versions (Python, PyTorch, VLMEvalKit, etc.) are pinned in the `requirements` file shipped with the supplementary material, which together with the A100 GPU is sufficient to reproduce the runs. The response-time analysis in Section 4.3 reports relative wall-clock latency between SAP and LongCoT under this hardware; further wall-clock numbers (per-route generation time and parallel-instance scaling) are reported in Appendix B.12.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors have read the NeurIPS Code of Ethics and confirm full conformance. The work introduces no new datasets, no human-subject data, and no high-risk model release; SAP is a training-free inference procedure layered on publicly available pre-trained VLMs.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: An Impact Statement is included before the acknowledgments. It notes that improved visual grounding may make VLM outputs more reliable, while SAP's multi-route inference increases total model computation relative to a single short response and may therefore increase energy use and its associated environmental footprint. Parallel execution reduces wall-clock latency but does not remove this additional computation.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper releases no new pre-trained models, no scraped datasets, and no image generators. SAP is a training-free inference procedure operating on top of existing publicly available VLMs (Qwen3-VL, InternVL3.5, DeepSeek-VL2) and a frozen grounding tool (SAM); responsible release is governed by the upstream providers of these assets.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All upstream assets are credited by citation: backbones (Qwen3-VL [Bai et al., 2025], InternVL3.5 [Wang et al., 2025a], DeepSeek-VL2 [Wu et al., 2024b]), the grounding tool (SAM [Kirillov et al., 2023]), the evaluation harness (VLMEvalKit [Duan et al., 2024]), and every benchmark in the 16-benchmark suite (Appendix D). Appendix A states the applicable licenses for the models, model code, grounding tool, and evaluation harness; benchmark data are used under the terms of their original distributions and are not redistributed.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper introduces no new datasets or pre-trained models. The only artefacts produced by SAP at run time are reasoning principles and routes, which are ephemeral test-time outputs rather than released assets.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects; all evaluations use established public benchmarks.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects, so IRB review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: VLMs (Qwen3-VL-8B as the primary backbone, plus Qwen3-VL-2B/4B/30B, InternVL3.5-8B and DeepSeek-VL2 in ablations) are the core inference engines on which SAP operates and are explicitly described in Section 5 and Appendix B. SAP uses them only via standard API-style inference calls; no fine-tuning is performed. LLM usage for paper writing/editing was limited to surface-level proofreading and is not a methodological component.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.