PROVIC: PROVENANCE-AWARE CREDIT ASSIGNMENT FOR VISUAL TOOL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Group-based reinforcement learning (RL) trains visual-tool agents from final-answer rewards, but typically assigns the same trajectory-level credit to tool calls with different roles in evidence acquisition. Local group-based methods distinguish these calls by comparing returns under matched intermediate conditions. For visual-tool calls, however, neither ordered histories nor returned values reliably identify equivalent evidence. In particular, different acquisition orders can yield equivalent prior evidence, while identical values can arise from different image regions or calculations. To address this matching problem, we introduce ProVIC (Provenance-aware Visual Interaction Credit), which converts final-answer rewards into call-specific learning signals for visual-tool segments. By using a persistent evidence graph, ProVIC matches equivalent prior evidence across acquisition orders while preserving observation provenance. It then uses matched final-answer rewards to give each tool segment a call-specific advantage. Experiments with Qwen3-VL-8B-Instruct on five visual reasoning benchmarks show that ProVIC achieves superior performance over all the compared methods.

1 INTRODUCTION

Vision-language models (VLMs) answer questions by reasoning over visual inputs (Bai et al., 2025). When relevant evidence is difficult to isolate in the original image, visual-tool frameworks let the model inspect and transform that image during reasoning (Gupta & Kembhavi, 2023; Surís et al., 2023; Hu et al., 2024). The observations returned by these tools guide subsequent reasoning and tool use (Jiang et al., 2025). The model therefore makes a sequence of decisions about what evidence to acquire and how to use it to answer the question.

Outcome-based reinforcement learning (RL) trains VLMs to make these decisions by using final-answer feedback. Methods including Pixel Reasoner (Su et al., 2025a), VLM-R^3 (Jiang et al., 2025), and VTool-R1 (Wu et al., 2026) use Group Relative Policy Optimization (GRPO) (Shao et al., 2024b) or related objectives for visual-tool training. For each input, GRPO assigns each sampled rollout a trajectory-level advantage based on its final-answer reward and applies that advantage to every generated token in the rollout. Consequently, all tool calls in a rollout receive the same credit, even when some provide useful evidence and others are irrelevant or misleading.

To address this limitation, local group-based methods distinguish individual decisions by comparing returns under matching intermediate conditions. GiGPO (Feng et al., 2025) groups decisions reached at the same environment state, whereas HGPO (He et al., 2026) further conditions each group on state histories. The grouping rule therefore determines which returns are compared to assign local credit. For visual-tool calls, neither ordered histories nor returned values reliably identify equivalent evidence. Concretely, different acquisition orders can yield the same prior evidence state, while identical values can arise from different regions or calculations. Credit assignment must therefore match prior evidence across acquisition orders and distinguish new evidence by provenance.

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

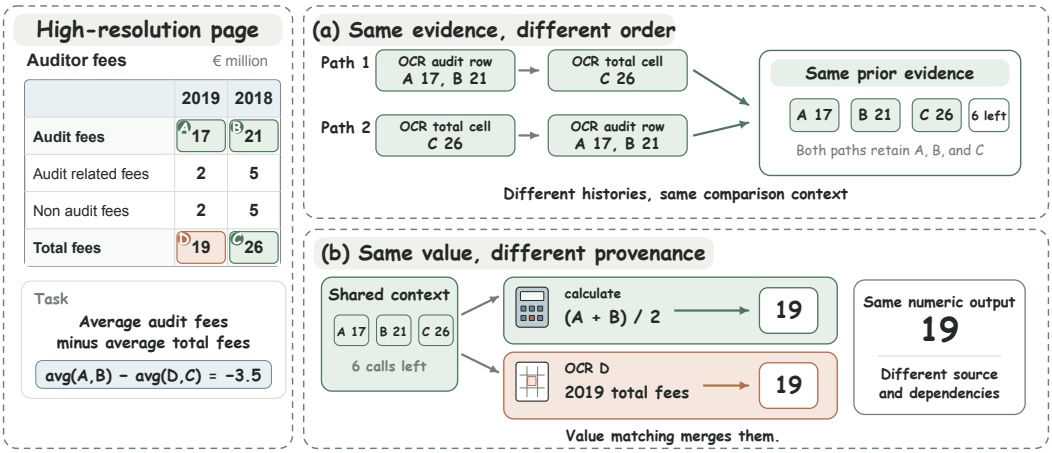

Figure 1: **Why visual-tool calls require provenance-aware matching.** A page from a high-resolution report illustrates how HGPO-style history matching and GiGPO-style state matching can fail. (a) The two histories apply the same two localized OCR calls in different orders and have six calls remaining. Accordingly, the prior evidence and remaining call budget are equivalent. (b) Calculating the audit-fee average and applying OCR to the 2019 total-fee cell both add an observation with value 19. Although both calls return 19, the calculation depends on two earlier observations, whereas the OCR result comes from a different page region.

Figure 1 shows how GiGPO-style state matching and HGPO-style history matching fail to satisfy these requirements in different ways. In panel (a), exact history matching separates calls that share the same acquired evidence and remaining budget. Specifically, the two histories apply the same OCR calls in different orders, yielding equivalent prior evidence states despite their different histories. In panel (b), state matching based only on the returned observation instead merges two calls that should remain distinct. Although the audit-fee calculation and OCR lookup both return 19, the calculation depends on two earlier observations, whereas the lookup reads another table cell. As a result, the two observations have different provenance.

To meet these requirements, we introduce ProVIC, a provenance-aware method that converts final-answer rewards into tool-segment advantages. At a high level, ProVIC matches tool calls across rollouts according to their prior evidence and outcomes. It then compares the corresponding final-answer returns to assign credit to each call. First, ProVIC represents each observation by its content and provenance in a persistent evidence graph. By using this graph, ProVIC forms a context group of calls with equivalent prior evidence and the same remaining budget. It then restricts this group to calls with matching outcomes, forming a nested transition group. ProVIC defines both groups at source, region, and content levels and selects the finest supported level. Second, rewards in the context group define a matched return baseline, while the target reward and rewards in the transition group estimate the return after the matched evidence transition. The normalized difference between these estimates gives the target call its tool-segment advantage.

We evaluate ProVIC on five visual reasoning benchmarks with Qwen3-VL-8B-Instruct. Across these benchmarks, ProVIC achieves superior performance over the compared methods. Ablations and fixed-history replay further support its grouping and credit-estimation design. Our contributions are summarized below.

- We introduce ProVIC, which uses a provenance-aware evidence representation to match equivalent prior evidence across acquisition orders while distinguishing observations with different origins.
- Building on this representation, we estimate call-specific advantages from matched context and transition returns, adapting the matching level to the support available in each rollout batch.
- ProVIC outperforms three matched Qwen3-VL-8B RL baselines across five visual reasoning benchmarks, demonstrating its effectiveness across diverse visual reasoning tasks.

2 Related Work

2.1 Visual Reasoning with Tools

Visual-tool systems extend VLMs with executable operations for inspecting and transforming images (Gupta & Kembhavi, 2023; Surís et al., 2023; Zhao et al., 2025; Zhang et al., 2025; Guo et al., 2025). Because each returned observation informs later decisions, these systems interleave evidence acquisition with reasoning. One training approach supervises the intermediate decisions directly by using stepwise preferences over tool actions (Li et al., 2025) or point-grounded tool traces (Ni et al., 2025). Alternatively, outcome-based RL trains the tool-use policy from task-level feedback, both for computational tools (Feng et al., 2026a) and for visual perception or image manipulation (Su et al., 2025a; Jiang et al., 2025; Wu et al., 2026; Lu et al., 2025; Su et al., 2025b; Zheng et al., 2026). Among outcome-based approaches, AdaptVision (Lin et al., 2026) separates the optimization of evidence acquisition from answer generation. ToolVision (Mao et al., 2026) instead rewards successful tool use according to its estimated benefit for each question. Within outcome-based visual-tool training, ProVIC focuses on matching individual calls to assign credit from final-answer rewards.

2.2 Credit Assignment for Tool Use

To distinguish the contributions of individual tool calls, one approach adds supervision beyond the final outcome. ToolRL (Qian et al., 2025) follows this approach by scoring calls against annotated tool-use decisions. A second approach derives intermediate signals by evaluating answers or verifying supporting evidence. For example, TACO (Feng et al., 2026b) compares answers decoded before and after a tool call. TRACE (Tao et al., 2026) measures the change in gold-answer predictability under a frozen reference model. TRACER (Yu et al., 2026) verifies links between answer claims and tool observations. These methods obtain local credit from annotations, auxiliary answer evaluations, or explicit evidence verification.

Without such local supervision, group-based methods must infer credit from final outcomes alone. GRPO (Shao et al., 2024b) assigns the same trajectory-level advantage to every decision in a rollout. To obtain finer-grained credit, GRPO-λ (Parthasarathi et al., 2025) introduces temporal weighting, GiGPO (Feng et al., 2025) compares decisions at shared states, and HGPO (He et al., 2026) further conditions these comparisons on nested state histories. For visual-tool interactions, PERIA (Li et al., 2026a) relaxes exact state matching, while TAPO (Dong et al., 2026) transfers credit through tool-argument similarity. BiPACE (Wang et al., 2026) uses learned representations to define groups for estimating state and action-conditioned returns. ProVIC instead uses provenance-aware matching of prior evidence and call outcomes to estimate call-specific credit from final-answer rewards.

3 Method

ProVIC first constructs provenance-aware comparison groups (Section 3.2). It then computes tool-segment advantages from their final-answer rewards (Section 3.3) and uses these advantages in policy optimization (Section 3.4).

3.1 Preliminaries

Given an input q consisting of images and a question, a VLM policy π_θ alternates language generation and tool execution. At each turn, the policy generates an assistant response conditioned on its ordered interaction history. We treat each response as one generated segment. If the segment contains a tool request, the executor runs the call and appends the executor output to the history before generation resumes. For credit assignment, we define the reasoning text and request for call t as its tool segment. A rollout is the complete sequence of generated segments and executor outputs.

To train this policy from final-answer feedback, we start from group-based RL with GRPO (Shao et al., 2024b). For each input and policy update, the sampling policy π_{old} generates $N \geq 2$ rollouts. After sampling, a final-answer verifier assigns each rollout i a reward $R_i \in [0, 1]$.

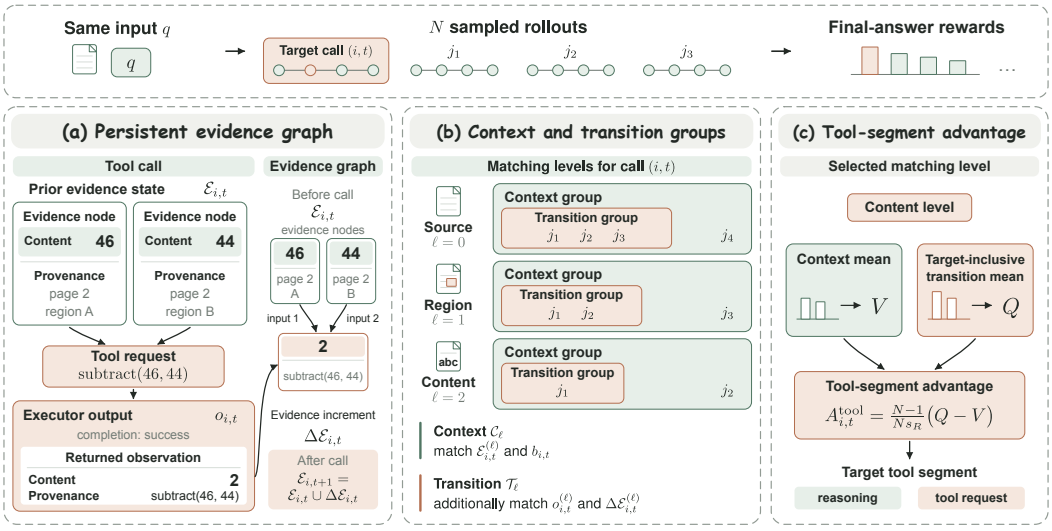

Figure 2: **Overview of the proposed ProVIC.** For each input q, the policy samples N rollouts and receives their final-answer rewards. (a) Executor output $o_{i,t}$ records the executed operation, completion status, and any returned observations. Each observation is represented by an evidence node with content and provenance. Here, the result 2 is new, so its node enters $\Delta\mathcal{E}_{i,t}$ and updates $\mathcal{E}_{i,t}$ to $\mathcal{E}_{i,t+1}$ for group matching. (b) Context groups contain other rollouts whose corresponding calls match the prior evidence state and remaining call budget. Transition groups further require the executor output and evidence increment to match those of the target call. Both groups use source, region, and content matching levels. (c) At the selected level, rewards in the context group determine V. The target reward and rewards in the transition group determine Q. Their normalized difference is the tool-segment advantage $A_{i,t}^{\text{tool}}$ assigned to the target tool segment.

The resulting trajectory-level advantage is

$$A_i^{\text{traj}} = \frac{R_i - \bar{R}}{s_R}. \tag{1}$$

Here $\bar{R}$ and s_R denote their mean and standard deviation. GRPO then applies this trajectory-level advantage to every generated token in rollout i, leaving individual tool calls indistinguishable.

3.2 Provenance-Aware Comparison Groups

Persistent evidence graph. ProVIC stores the evidence used for comparison groups in a persistent evidence graph, as shown in Figure 2(a). Each returned observation is represented by a graph node that records its content and provenance. The content is the returned value, and the provenance identifies where or how that value was obtained. For crop, zoom, OCR, and extraction results, the provenance records the source and original-image region. For calculated results, it instead records the executed operation and ordered inputs. The nodes accumulated before each call form its evidence state, which ProVIC matches across rollouts to construct the comparison groups.

Evidence state before and after a call. The evidence state $\mathcal{E}_{i,t}$ is the set of graph nodes available before call t in rollout i, independent of their acquisition order. ProVIC uses this state to form comparison groups, while the policy continues to condition on its ordered interaction history. Initially, the state contains the input images and text. After call t executes, the executor output $o_{i,t}$ records the executed operation, completion status, and any returned observations. Let $\Delta\mathcal{E}_{i,t}$ denote the new evidence nodes created for returned observations not yet represented in $\mathcal{E}_{i,t}$. As shown in Figure 2(a), the evidence state then updates as

$$\mathcal{E}_{i,t+1} = \mathcal{E}_{i,t} \cup \Delta\mathcal{E}_{i,t}. \tag{2}$$

Accordingly, $o_{i,t}$ describes the complete call outcome, whereas $\Delta\mathcal{E}_{i,t}$ contains only the evidence newly added to the graph.

Context and transition groups. For a target call t in rollout i, denoted by (i,t), the context group contains other rollouts whose corresponding calls have equivalent prior evidence and the same remaining call budget. The transition group further restricts these rollouts to those whose corresponding calls have executor outputs and evidence increments matching $o_{i,t}$ and $\Delta\mathcal{E}_{i,t}$, respectively. Together, the two groups define a matched return comparison for the target call. Rewards in the context group estimate a baseline under the same prior evidence and remaining budget, whereas the target reward and rewards in the transition group estimate the return after the matched evidence transition. Their difference yields call-specific credit for the target call.

Nested matching levels for both groups. As shown in Figure 2(b), ProVIC constructs both groups at three nested matching levels. Across levels, only the equivalence test changes, while the roles of the two groups remain unchanged. Each finer level adds detail while preserving the coarser conditions, allowing ProVIC to back off when exact matches are sparse. The source level ($\ell = 0$) matches the originating image or document page and the type of returned value. The region level ($\ell = 1$) additionally matches the original-image bounding box, and the content level ($\ell = 2$) further matches the exact returned text, number, image, or structured result. At level ℓ, we denote the prior evidence state, executor output, and evidence increment by $\mathcal{E}_{i,t}^{(\ell)}$, $o_{i,t}^{(\ell)}$, and $\Delta\mathcal{E}_{i,t}^{(\ell)}$, respectively. The remaining call budget before call t in rollout i is $b_{i,t}$. With the target call (i,t) fixed, we index other rollouts by j and define the two groups as

$$\begin{aligned}\mathcal{C}_\ell &= \{j \neq i : b_{j,t} = b_{i,t}, \quad \mathcal{E}_{j,t}^{(\ell)} = \mathcal{E}_{i,t}^{(\ell)}\},\\ \mathcal{T}_\ell &= \{j \in \mathcal{C}_\ell : o_{j,t}^{(\ell)} = o_{i,t}^{(\ell)}, \quad \Delta\mathcal{E}_{j,t}^{(\ell)} = \Delta\mathcal{E}_{i,t}^{(\ell)}\}.\end{aligned} \tag{3}$$

By construction, $\mathcal{T}_\ell \subseteq \mathcal{C}_\ell$, and the groups are nested as $\mathcal{C}_2 \subseteq \mathcal{C}_1 \subseteq \mathcal{C}_0$ and $\mathcal{T}_2 \subseteq \mathcal{T}_1 \subseteq \mathcal{T}_0$.

3.3 Tool-Segment Advantage from Matched Rewards

Selecting the matching level. To balance matching specificity against the cross-rollout support needed for credit estimation, ProVIC adapts the matching level to the rollout batch. When transition matches are available, ProVIC selects the finest level with a nonempty transition group. Otherwise, if at least one context group is nonempty, it selects the finest level with a nonempty context group. In either case, let ℓ denote the selected level and abbreviate its groups as $\mathcal{C} = \mathcal{C}_\ell$ and $\mathcal{T} = \mathcal{T}_\ell$. If every context group is empty, ProVIC sets $\mathcal{C} = \{j : j \neq i\}$ and $\mathcal{T} = \varnothing$, recovering the trajectory-level GRPO advantage.

Computing the tool-segment advantage. Given $\mathcal{C}$ and $\mathcal{T}$, the context mean V averages the final-answer rewards over $\mathcal{C}$ and provides a baseline under the same prior evidence state. The target-inclusive transition mean Q averages R_i with the rewards over $\mathcal{T}$, and it reduces to R_i when $\mathcal{T}$ is empty. $Q - V$ therefore measures the reward difference between the target-inclusive transition mean and the context mean. By using these definitions, ProVIC computes the following matched means and tool-segment advantage.

$$V = \frac{\sum_{j\in\mathcal{C}} R_j}{|\mathcal{C}|}, \quad Q = \frac{R_i + \sum_{j\in\mathcal{T}} R_j}{1 + |\mathcal{T}|}, \quad A_{i,t}^{\text{tool}} = \frac{N-1}{N s_R}\big(Q - V\big). \tag{4}$$

The factor $(N-1)/(Ns_R)$ places this difference on the GRPO advantage scale. When $s_R \leq 10^{-6}$, we set both trajectory and tool-segment advantages to zero to avoid dividing by a negligible reward scale. Because the groups are specific to (i,t), the resulting $A_{i,t}^{\text{tool}}$ is assigned to the target tool segment. Positive values reinforce calls whose target-inclusive transition mean exceeds the matched context baseline. Conversely, negative values suppress calls whose target-inclusive transition mean falls below that baseline.

3.4 ProVIC Policy Optimization

For rollout i, let $\mathcal{S}_i$ denote its generated assistant segments and $\mathcal{K}_{i,s}$ the rollout-level token positions belonging to segment s. A segment ending with tool call t receives $A_{i,s} = A_{i,t}^{\text{tool}}$. All other generated segments, including the final-answer segment, receive $A_{i,s} = A_i^{\text{traj}}$. Every token in a tool segment therefore shares its call-specific advantage, while the remaining tokens retain trajectory-level credit.

For the token $y_{i,k}$ at position k, let $h_{i,k}$ denote its prefix history. We define its importance ratio as $\rho_{i,k}(\theta) = \pi_\theta(y_{i,k} \mid h_{i,k})/\pi_{\text{old}}(y_{i,k} \mid h_{i,k})$. ProVIC maximizes the clipped policy objective

$$\mathcal{J}_{\text{ProVIC}}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathcal{S}_i|}\sum_{s\in\mathcal{S}_i}\frac{1}{|\mathcal{K}_{i,s}|}\sum_{k\in\mathcal{K}_{i,s}}\Big[\min\big(\rho_{i,k}(\theta)A_{i,s}, \text{clip}(\rho_{i,k}(\theta), 1-\epsilon, 1+\epsilon)A_{i,s}\big) - \beta_{\text{KL}}D_{i,k}(\theta)\Big]. \tag{5}$$

Here ϵ sets the clipping interval, and $D_{i,k}$ is the sampled KL penalty to the fixed reference policy π_{ref} obtained by supervised fine-tuning (SFT). The segment advantages are computed from the π_{old} rollout batch and held fixed during optimization. To compute these advantages, ProVIC indexes the counts and reward sums for each context and transition group. Each target call then obtains V and Q from these statistics without rescanning the peer rollouts.

4 Experiments

4.1 Experimental Setup

Benchmarks and metrics. We evaluate ProVIC on five benchmarks. Together, they cover visual search, information extraction, and quantitative reasoning. VisFinEval measures financial image understanding (Liu et al., 2025), while FinMME covers financial chart understanding and calculation (Luo et al., 2025). ChartQAPro evaluates chart and dashboard reasoning (Masry et al., 2025), and O3-Bench tests high-resolution visual search (Li et al., 2026b). We use the full MathVision test set for visual mathematics (Wang et al., 2024). Each benchmark contributes one primary metric. These are weighted accuracy for VisFinEval, average accuracy for FinMME, and overall accuracy for the other three benchmarks. Their unweighted mean gives the aggregate score.

Training setup. Our base policy is Qwen3-VL-8B-Instruct (Bai et al., 2025), equipped with page selection, crop, zoom, OCR, structured extraction, chart operations, and calculation tools. Before RL, every method is initialized from the same SFT checkpoint trained on 6,000 tool-use trajectories. Each method then trains on 12,000 RL prompts. The SFT and RL pools contain disjoint source documents and images. For each RL prompt, we sample $N = 16$ rollouts with at most eight attempted tool calls and 2,048 generated tokens per assistant turn. We use the same five training seeds for all methods and report their mean. At evaluation, we use greedy decoding with the same call and per-turn token limits.

Compared methods. We compare ProVIC with GRPO (Shao et al., 2024b), HGPO (He et al., 2026), and BiPACE (Wang et al., 2026). These baselines cover trajectory-level credit, history-based grouping, and action-conditioned return estimation. GRPO assigns one advantage to the full rollout. In our visual-tool setting, HGPO groups calls by the current image-text observation and progressively longer histories of tool requests and responses. BiPACE clusters the VLM's representations of the pre-call multimodal histories and compares returns for the same parsed tool request within each cluster. For a controlled comparison, all four methods share the SFT initialization, final-answer verifier, optimizer, loss aggregation, training budget, tool executor, and tool-segment definition.

Evaluation setup. All methods evaluated in our setup use Qwen3-VL-8B-Instruct, and the four matched RL methods in Table 1 share the same evaluation budget. ReAct follows the prompting strategy of Yao et al. (2023). By contrast, the published general-purpose VLM rows retain their source evaluation settings and reported scores (Luo et al., 2025; Masry et al., 2025), including chain-of-thought prompting for GPT-4o on ChartQAPro. We provide additional settings, protocols, and results in Appendices D–G.

Diagnostic sampling. For both diagnostics in Figure 3, we sample 200 held-out inputs with 16 rollouts each, forming 3,200 shared credit rollouts. Each input contributes at most one target call per evaluation set, capped at 200 calls overall and 100 each for different-order and value-collision subsets. The equal-peer control retains collision targets with nonempty context and transition groups under both observation identities.

Table 1: **Performance on five visual reasoning benchmarks.** Scores are on a 0–100 scale. Mean equally weights the five benchmark metrics. Published rows follow their source evaluation settings. The four matched RL rows average five training seeds, and −− denotes an unreported score.

Model / method	VisFinEval Weighted Acc.	FinMME Avg. Acc.	ChartQAPro Acc.	O3-Bench Acc.	MathVision Acc.	Mean
Published general-purpose VLMs						
GPT-4o	68.5	46.56	41.68	28.0	30.39	–
Gemini-2.5-Pro	64.7	–	–	65.4	–	–
Qwen2.5-VL-72B	71.0	52.54	–	–	25.9	–
InternVL2.5-8B	–	41.90	35.67	–	–	–
Qwen3-VL-32B-Instruct	–	–	–	60.4	63.4	–
Local Qwen3-VL-8B reference systems						
Qwen3-VL-8B-Instruct	66.8	40.4	45.8	43.4	53.7	50.0
ReAct	65.9	39.5	44.9	42.2	52.9	49.1
SFT	70.8	48.1	50.7	46.0	55.8	54.3
Matched Qwen3-VL-8B RL comparisons						
GRPO	73.4	51.6	52.5	47.3	57.4	56.4
HGPO	74.0	52.1	53.0	47.8	57.9	57.0
BiPACE	74.3	52.3	53.2	47.9	58.0	57.1
ProVIC	75.9	54.0	54.7	49.2	58.4	58.4

4.2 Main Results

Table 1 shows that ProVIC outperforms GRPO, HGPO, and BiPACE on every benchmark, achieving the highest five-benchmark mean of 58.4 among the four RL methods. The improvement spans financial understanding, chart reasoning, visual search, and mathematics, showing that ProVIC benefits several forms of visual-tool reasoning.

4.3 Ablation Study

Ablation setup. To identify which parts of ProVIC contribute to these gains, we report trained-policy performance for every ablation. For the component controls, we additionally apply fixed-history decision replay. Policy performance is the mean on VisFinEval, ChartQAPro, and O3-Bench. Decision replay instead tests whether a credit rule ranks pairs of candidate calls in the same order as independent continuation returns. We initially sample 200 successful call positions in SFT rollouts and retain their preceding histories. At each history, we sample a shared batch of 16 SFT rollouts and apply every credit rule to the same candidate calls. We then execute each candidate from that history and sample 16 fresh continuations. Replay agreement averages these pairwise comparisons within each history and then across histories.

Representation and estimator. Table 2(a) separates grouping from credit estimation by pairing History and Evidence grouping with the HGPO and ProVIC estimators. History matches the current observation with zero, one, or two preceding tool exchanges in order. Evidence matches accumulated observations at source, region, and content levels. All variants match the remaining call budget and use the same undiscounted terminal rewards. The policy always conditions on its ordered history, and both Evidence variants receive identical provenance information. Evidence grouping improves performance with either estimator. Similarly, switching from the HGPO estimator to ProVIC improves performance under both grouping rules. Together, Evidence grouping and the ProVIC estimator yield the best performance on all three benchmarks.

Evidence and transition information. Both restricted evidence retention and removal of transition information reduce policy performance in Table 2(b). The recent-evidence control keeps the

Table 2: **Ablations of representation and credit estimation.** Panel (a) pairs three-level History and Evidence representations with the HGPO and ProVIC credit estimators. Panel (b) ablates evidence retention and matched-reward computation. Policy mean equally averages VisFinEval (VF), ChartQAPro (CQP), and O3-Bench. Replay agreement compares credit rankings with rankings from independent continuation returns. Recent evidence retains the initial observations, evidence from the last three tool attempts, and their operand dependencies.

(a) Representation × estimator

Representation	Estimator	VF	CQP	O3	Mean
History	HGPO	73.8	52.9	47.7	58.1
Evidence	HGPO	74.1	53.1	47.9	58.4
History	ProVIC	74.4	53.3	48.0	58.6
Evidence	**ProVIC**	75.9	54.7	49.2	59.9

(b) Evidence retention and matched-reward ablations

Component	Variant	Policy mean	Replay agreement
Evidence retention	Recent three-call evidence	58.5	71.2%
Transition information	Context baseline only	58.2	68.5%
Matching-level selection	Source level only	58.6	72.4%
Matching-level selection	Content level only	58.4	70.8%
Target inclusion	Excluded from transition mean	58.7	73.6%
Full method	**ProVIC**	59.9	76.4%

initial observations, evidence from the last three tool attempts, and any earlier operands needed by calculations in that window. Failed attempts count toward the window. Its lower score favors retaining earlier evidence for credit assignment. The context-only control keeps ProVIC's selected level and context baseline but sets the transition group to empty, leaving credit proportional to $R_i - V$. Its lower score supports pooling matched transition rewards for credit estimation.

Matching level and target reward. The remaining controls favor adaptive matching over fixing either the coarsest or finest level. The source-only and content-only controls use $\mathcal{C}_0, \mathcal{T}_0$ and $\mathcal{C}_2, \mathcal{T}_2$, respectively, falling back to GRPO when the corresponding context group is empty. Both underperform full ProVIC, supporting selection of the matching level according to the available comparison groups. Target inclusion also matters. The exclude-target control omits R_i from the transition mean whenever the transition group is nonempty and otherwise uses R_i. The lower score supports retaining the target reward in the transition mean.

4.4 Analysis

Fixed-history decision replay. We next test whether the credit rules favor calls with higher continuation returns under the common SFT policy. In Table 2(b), ProVIC agrees more closely with independent continuation-return rankings than the context-only and content-only controls, reaching 76.4% replay agreement. This provides a direct check of credit rankings alongside the trained-policy evaluation. The shared prior history holds the pre-call evidence fixed, focusing this diagnostic on outcome matching and credit estimation. The following two diagnostics extend this comparison across histories and isolate observation provenance.

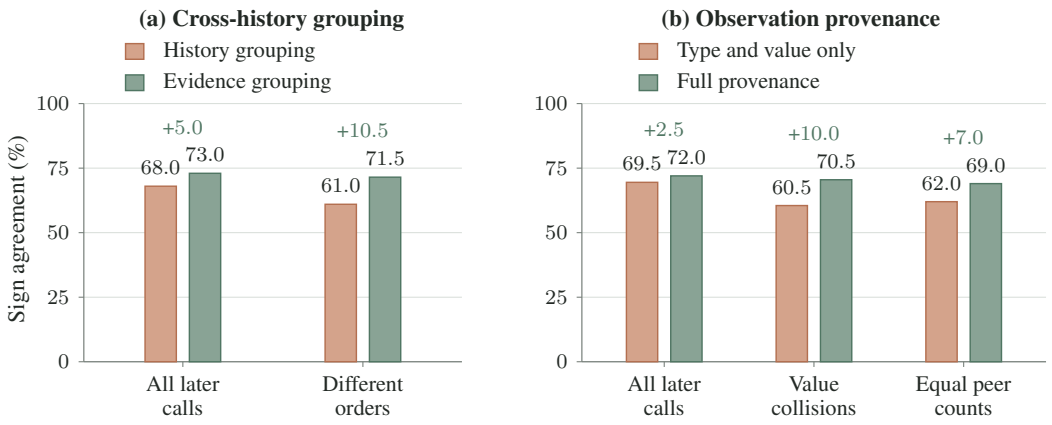

Figure 3: **Mechanism diagnostics.** Sign agreement (%) for successful calls at attempt indices 2–8. (a) History versus Evidence grouping. (b) Type-and-value versus full-provenance identities under fixed exact matching. The equal-peer control matches context and transition group sizes on the supported collision subset. Bars show point estimates, and labels above pairs show differences in percentage points.

Cross-history grouping. We compare History and Evidence grouping on the same batches of 16 SFT rollouts with the ProVIC estimator fixed. For each target, both rules use independent estimates of the return after executing its tool segment and the baseline return from letting the policy continue freely from the same history. Sign agreement measures whether the assigned credit and this return difference have the same sign. Figure 3(a) favors Evidence grouping, especially when equivalent evidence is acquired in different orders, where agreement rises from 61.0% to 71.5%. This supports credit comparison across acquisition histories.

Observation provenance. We next compare full identities, which retain content and provenance, with type-and-value identities under fixed exact matching. Both conditions share the rollout batches and independent return comparisons. Figure 3(b) shows the largest difference when equal returned values have different origins, with full provenance reaching 70.5% agreement compared with 60.5% for type-and-value identities. To control group size, we subsample value-only peers to match the full-provenance context and transition counts on the same supported collision targets. The advantage persists under this control, supporting the role of provenance in selecting informative comparisons.

Evidence requirements. Finally, we group a 2,000-question analysis subset from the three ablation benchmarks by the evidence required to answer each question. We assign these categories from source material and reference answers before examining model outputs. ProVIC's advantage over GRPO is smallest when the supplied view is sufficient and largest when answering requires multiple sources or dependent calculations. This pattern is consistent with ProVIC's focus on accumulated evidence, since a call's contribution depends on how its output relates to observations already available.

5 Conclusion

We introduce ProVIC, a provenance-aware credit-assignment method for visual-tool RL. ProVIC stores returned observations in a persistent evidence graph, forms matched context and transition groups at multiple levels, and assigns each tool call a tool-segment advantage from final-answer rewards. Experiments on five visual reasoning benchmarks show improved performance over GRPO, HGPO, and BiPACE. A limitation is that ProVIC relies on executor-supplied provenance. When region- or content-level matches are sparse, it uses coarser groups that may merge observations with different fine-grained provenance. Future work will make matching robust to missing provenance and replace discrete backoff with smoother weighting across levels.

AI use statement. Generative AI tools were used to assist with language editing and improving the readability of the manuscript. All AI-assisted content was reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

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, 2025. URL https://arxiv.org/abs/2511.21631.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. FinQA: A dataset of numerical reasoning over financial data. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3697–3711, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.300. URL https://aclanthology.org/2021.emnlp-main.300/.

Chengqi Dong, Chuhuai Yue, Hang He, yandong liu, Fenghe Tang, S Kevin Zhou, Xiaohan Wang, Jiajun Chai, and Guojun Yin. Tapo: Tool-aware policy optimization via credit transfer for multimodal search agents, 2026. URL https://arxiv.org/abs/2606.05784.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 37909–37926, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/4038c9208dfc22644c60ad39c24e5c53-Paper-Conference.pdf.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 46375–46408. Curran Associates, Inc., 2025. doi: 10.52202/085713-1544. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/420c9f777c0b4f78d515e53cf74d58b2-Paper-Conference.pdf.

Mingkuan Feng, Jinyang Wu, Hao Gu, Fangrui Lv, Ruihan Jin, Chuyuan Zhang, Zhengqi Wen, and Jianhua Tao. Taco: Tool-augmented credit optimization for agentic tool use, 2026b. URL https://arxiv.org/abs/2606.30251.

Zirun Guo, Minjie Hong, Feng Zhang, Kai Jia, and Tao Jin. Thinking with programming vision: Towards a unified view for thinking with images, 2025. URL https://arxiv.org/abs/2512.03746.

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14953–14962, June 2023.

Shuo He, Lang Feng, qi wei, Xin Cheng, Lei Feng, and Bo An. Hierarchy-of-groups policy optimization for long-horizon agentic tasks. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 27572–27593, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/2e7eb932f93d5db3b61c7a001e72fd0f-Paper-Conference.pdf.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 139348–139379. Curran Associates, Inc., 2024. doi: 10.52202/079017-4423. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/fb82011040977c7712409fbdb5456647-Paper-Conference.pdf.

Chaoya Jiang, Yongrui Heng, Wei Ye, Han Yang, Haiyang Xu, Ming Yan, Ji Zhang, Fei Huang, and Shikun Zhang. Vlm-r^3: Region recognition, reasoning, and refinement for enhanced multimodal chain-of-thought, 2025. URL https://arxiv.org/abs/2505.16192.

Changye Li, Meng Lu, Yi Wu, and Ligeng Zhu. Perceive, interact, reason: Building tool-augmented visual agents for spatial reasoning, 2026a. URL https://arxiv.org/abs/2606.12830.

Kaican Li, Lewei Yao, Jiannan Wu, Tiezheng YU, Jierun Chen, Haoli Bai, LU HOU, Lanqing HONG, Wei Zhang, and Nevin Zhang. Insight-o3: Empowering multimodal foundation models with generalized visual search. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 128950–129002, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/d194e6b365e2cca61170445f25412122-Paper-Conference.pdf.

Pengxiang Li, Zhi Gao, Bofei Zhang, Yapeng Mi, Xiaojian (Shawn) Ma, Chenrui Shi, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, and Qing Li. Iterative tool usage exploration for multimodal agents via step-wise preference tuning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 59496–59528. Curran Associates, Inc., 2025. doi: 10.52202/085713-1991. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/55d16334943f8728073e17139e5baa3d-Paper-Conference.pdf.

Zichuan Lin, Yicheng Liu, Yang Yang, Lvfang Tao, and Deheng Ye. Adaptvision: Efficient vision-language models via adaptive visual acquisition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11923–11932, June 2026.

Zhaowei Liu, Xin Guo, Haotian Xia, Lingfeng Zeng, Fangqi Lou, Jinyi Niu, Mengping Li, Qi Qi, Jiahuan Li, Wei Zhang, Yinglong Wang, Weige Cai, Weining Shen, and Liwen Zhang. VisFinEval: A scenario-driven Chinese multimodal benchmark for holistic financial understanding. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 24088–24146, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1229. URL https://aclanthology.org/2025.emnlp-main.1229/.

Meng Lu, Ran Xu, Yi Fang, Wenxuan Zhang, Yue Yu, Gaurav Srivastava, Yuchen Zhuang, Mohamed Elhoseiny, Charles Fleming, Carl Yang, Zhengzhong Tu, Yang Xie, Guanghua Xiao, Hanrui Wang, Di Jin, Wenqi Shi, and Xuan Wang. Scaling agentic reinforcement learning for tool-integrated reasoning in vlms, 2025. URL https://arxiv.org/abs/2511.19773.

Junyu Luo, Zhizhuo Kou, Liming Yang, Xiao Luo, Jinsheng Huang, Zhiping Xiao, Jingshu Peng, Chengzhong Liu, Jiaming Ji, Xuanzhe Liu, Sirui Han, Ming Zhang, and Yike Guo. FinMME: Benchmark dataset for financial multi-modal reasoning evaluation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29465–29489, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1426. URL https://aclanthology.org/2025.acl-long.1426/.

Delin Mao, Chenghao Sun, Jingwei Song, Chishui Chen, and Linfeng Zhang. Toolvision: Learning when and how to use visual tools with capability-aligned supervision, 2026. URL https://arxiv.org/abs/2608.08907.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.177. URL https://aclanthology.org/2022.findings-acl.177/.

Ahmed Masry, Mohammed Saidul Islam, Mahir Ahmed, Aayush Bajaj, Firoz Kabir, Aaryaman Kartha, Md Tahmid Rahman Laskar, Mizanur Rahman, Shadikur Rahman, Mehrad Shahmohammadi, Megh Thakkar, Md Rizwan Parvez, Enamul Hoque, and Shafiq Joty. ChartQAPro: A more diverse and challenging benchmark for chart question answering. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19123–19151, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.978. URL https://aclanthology.org/2025.findings-acl.978/.

Minheng Ni, Zhengyuan Yang, Linjie Li, Chung-Ching Lin, Kevin Lin, Wangmeng Zuo, and Lijuan Wang. Point-rft: Improving multimodal reasoning with visually grounded reinforcement finetuning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 20538–20559. Curran Associates, Inc., 2025. doi: 10.52202/085713-0692. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/1d83ad88759cef8192451543e5d59bf6-Paper-Conference.pdf.

Prasanna Parthasarathi, Mathieu Reymond, Boxing Chen, Yufei Cui, and Sarath Chandar. Grpo-λ: Credit assignment improves llm reasoning, 2025. URL https://arxiv.org/abs/2510.00194.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru WANG, Xiusi Chen, Dilek Hakkani-Tur, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 105523–105553. Curran Associates, Inc., 2025. doi: 10.52202/085713-3524. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/97c5b2707228e7e3fb67e4ecc2e0e607-Paper-Conference.pdf.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 8612–8642. Curran Associates, Inc., 2024a. doi: 10.52202/079017-0275. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/0ff38d72a2e0aa6dbe42de83a17b2223-Paper-Datasets_and_Benchmarks_Track.pdf.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024b. URL https://arxiv.org/abs/2402.03300.

Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel space reasoning via curiosity-driven reinforcement learning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 8222–8251. Curran Associates, Inc., 2025a. doi: 10.52202/085713-0280. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/0c38f54740062529aa4117a04b583f3c-Paper-Conference.pdf.

Zhaochen Su, Linjie Li, Mingyang Song, Yunzhuo Hao, Zhengyuan Yang, Jun Zhang, Guanjie Chen, Jiawei Gu, Juntao Li, Xiaoye Qu, and Yu Cheng. Openthinkimg: Learning to think with images via visual tool reinforcement learning, 2025b. URL https://arxiv.org/abs/2505.08617.

Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11888–11898, October 2023.

Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. Slidevqa: A dataset for document visual question answering on multiple images. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):13636–13645, June 2023. ISSN 2159-5399. doi: 10.1609/aaai.v37i11.26598. URL http://dx.doi.org/10.1609/aaai.v37i11.26598.

Leitian Tao, Baolin Peng, Wenlin Yao, Tao Ge, Hao Cheng, Mike Hang Wang, Jianfeng Gao, and Sharon Li. Trace: Turn-level reward assignment via credit estimation for long-horizon agents, 2026. URL https://arxiv.org/abs/2607.13988.

Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. Hierarchical multimodal transformers for multipage docvqa. *Pattern Recognition*, 144:109834, December 2023. ISSN 0031-3203. doi: 10.1016/j.patcog.2023.109834. URL http://dx.doi.org/10.1016/j.patcog.2023.109834.

Jordy Van Landeghem, Rubèn Tito, Łukasz Borchmann, Michał Pietruszka, Pawel Joziak, Rafal Powalski, Dawid Jurkiewicz, Mickael Coustaty, Bertrand Anckaert, Ernest Valveny, Matthew Blaschko, Sien Moens, and Tomasz Stanislawek. Document understanding dataset and evaluation (dude). In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19528–19540, October 2023.

Hanyang Wang, Weijieying Ren, Yuxiang Zhang, Ding Cao, Zhizhao Zeng, Ke Zeng, and Tianxiang Zhao. Bipace: Bisimulation-guided policy optimization with action counterfactual estimation for llm agents, 2026. URL https://arxiv.org/abs/2606.25556.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 95095–95169. Curran Associates, Inc., 2024. doi: 10.52202/079017-3014. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ad0edc7d5fa1a783f063646968b7315b-Paper-Datasets_and_Benchmarks_Track.pdf.

Mingyuan Wu, Jingcheng Yang, Jize Jiang, Meitang Li, Kaizhuo Yan, Hanchao Yu, Minjia Zhang, ChengXiang Zhai, and Klara Nahrstedt. Vtool-r1: Vlms learn to think with images via reinforcement learning on multimodal tool use. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 78298–78319, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/7e6f445a74cdb71931aac64f1e3f49c9-Paper-Conference.pdf.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Bihui Yu, Caijun Jia, Jing Chi, Xiaohan Liu, Yining Wang, He Bai, Yuchen Liu, Jingxuan Wei, and Junnan Zhu. Tracer: Verifiable generative provenance for multimodal tool-using agents, 2026. URL https://arxiv.org/abs/2605.09934.

Yi-Fan Zhang, Xingyu Lu, Shukang Yin, Chaoyou Fu, Wei Chen, Xiao Hu, Bin Wen, Kaiyu Jiang, Changyi Liu, Tianke Zhang, Haonan Fan, Kaibing Chen, Jiankang Chen, Haojie Ding, Kaiyu Tang, Zhang Zhang, Liang Wang, Fan Yang, Tingting Gao, and Guorui Zhou. Thyme: Think beyond images, 2025. URL https://arxiv.org/abs/2508.11630.

Shitian Zhao, Haoquan Zhang, Shaoheng Lin, Ming Li, Qilong Wu, Kaipeng Zhang, and Chen Wei. Pyvision: Agentic vision with dynamic tooling, 2025. URL https://arxiv.org/abs/2507.07998.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing "thinking with images" via reinforcement learning, 2026. URL https://arxiv.org/abs/2505.14362.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3277–3287, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.254. URL https://aclanthology.org/2021.acl-long.254/.

A Training Algorithm and Notation

Algorithm 1 summarizes one training loop. Comparison groups are formed separately for each input, and their statistics are fixed during the policy update. Table 3 collects the notation used below.

Algorithm 1 ProVIC training

Input SFT policy π_θ, fixed reference π_{ref}, prompt batches, $N \geq 2$
for each prompt batch **do**
  Set the sampling policy $\pi_{\mathrm{old}} \leftarrow \pi_\theta$
  for each input q in the batch **do**
    Sample N rollouts and obtain final-answer rewards $R_1, \ldots, R_N$
    Record $(\mathcal{E}_{i,t}, o_{i,t}, \Delta\mathcal{E}_{i,t}, b_{i,t})$ for every tool attempt
    Compute $\bar{R}$ and the population standard deviation s_R
    Index context and transition identities at levels $\ell = 0, 1, 2$ by Eq. 8
    for each rollout i **do**
      Set A_i^{traj} by Eq. 1, or to zero if $s_R \leq 10^{-6}$
      for each recorded tool call (i, t) **do**
        Query $\mathcal{C}_\ell$ and $\mathcal{T}_\ell$ at all levels, excluding rollout i
        if any transition group is nonempty **then**
          Select the finest such level and its context and transition groups
        else if any context group is nonempty **then**
          Select the finest such level and set $\mathcal{T} \leftarrow \varnothing$
        else
          Set $\mathcal{C} \leftarrow \{j : j \neq i\}$ and $\mathcal{T} \leftarrow \varnothing$
        end if
        Compute V, Q, and $A_{i,t}^{\mathrm{tool}}$ by Eq. 4, or set the advantage to zero if $s_R \leq 10^{-6}$
      end for
      Assign tool advantages to tool segments and A_i^{traj} to other segments
    end for
  end for
  Update θ with the batch mean of Eq. 5, masking nongenerated tokens
end for

Table 3: **Core notation.** A target call (i, t) and its input q are implicit in the group and return-estimate symbols.

Symbol	Meaning
q, N, i, t	Input, number of rollouts, rollout index, and attempted-call index
R_i, $\bar{R}$, s_R	Final-answer reward, batch mean, and population standard deviation for q
$o_{i,t}$	Executor output containing the operation, status, and returned observations
$\mathcal{E}_{i,t}$, $\Delta\mathcal{E}_{i,t}$	Evidence available before the call and new evidence added by it
$b_{i,t}$	Remaining call budget before the call
$\ell = 0, 1, 2$	Source, region, and content matching levels
$\mathcal{C}_\ell$, $\mathcal{T}_\ell$	Context and transition groups, both excluding the target rollout
V, Q	Context mean reward and target-inclusive transition mean reward
$A_{i,t}^{\mathrm{tool}}$, A_i^{traj}	Tool-segment and trajectory-level advantages

B Evidence Representation

B.1 Executor Records and Observation Identities

An executor output $o_{i,t}$ records the operation, completion status, and any returned observations. Each observation has content and provenance and is represented by a canonical graph node. Thus a single output may return several observations, including observations already present in the evidence state. Table 4 distinguishes the fields used to identify observations from those used to compare call

outcomes. Reference answers and correctness labels are used by the verifier and are excluded from the evidence graph.

Table 4: **Executor records used by ProVIC.** Observation identities determine the evidence state, while the call signature also retains the executed operation and completion status.

Field	Role in the representation
`source_id`	Identifies the original image or document page.
`root_support`	Locates the observation in original-image coordinates.
`typed_payload`	Stores returned text, a number, an image, or a structured value.
`operand_refs`	Links a calculated observation to its ordered input observations.
`operation`	Identifies the executed operation and the producing operator of a calculation.
`status`	Distinguishes successful and failed call outcomes in the transition signature.
`parent_artifact`	Resolves a temporary view to its source coordinates before matching.

Sources and image regions. A source is identified by its document/page or image identifier within the input and a digest of the original image. Multi-page inputs retain their page order. A region is a half-open bounding box (x_0, y_0, x_1, y_1) in original-image pixels. The executor composes the crop and resize transforms along the parent chain, maps the local box to the original image, and rounds outward to integer boundaries. Sets of boxes are sorted and deduplicated. Whole-image observations use the full image extent, while nonspatial constants use a designated null region. The region records where the tool looked, while the content records what it returned.

Direct observations. For source s, region B, observation type k, and content y, define the observation identity

$$u = \big(\mathrm{obs}, s, B, k, \mathrm{canon}(y)\big). \tag{6}$$

The function canon normalizes content according to its type. Text uses Unicode NFC, normalized line endings, and collapsed whitespace, while case, punctuation, and token order are preserved. Scalars use exact decimals and explicit units, with unit conversion only when specified by the executor. Structured objects sort fields and preserve array order. Images retain their dimensions, color mode, and a digest of the returned pixels. Two observations match when their tuples are equal. Hashes index these tuples.

This construction matches observations that differ only in serialization or temporary view coordinates. Different image resolutions, OCR strings, and bounding boxes remain distinct. Grouping compares the complete evidence state rather than only the current observation. For example, a path that inspects an extra view has a different context even if its final extraction matches that of another path.

Calculated observations. A calculation records its executed operator f, operands $u_1, \ldots, u_m$, and typed result y in the following form.

$$u = \big(\mathrm{derive}, f, [(r_1, \mathrm{id}(u_1)), \ldots, (r_m, \mathrm{id}(u_m))], \mathrm{canon}(y)\big). \tag{7}$$

Here $\mathrm{id}(u_k)$ is the canonical identity of operand u_k, and r_k gives its role or position. Executor references determine these dependencies and preserve operand order and repeated operands. Operators retain their executed meaning. We apply no general symbolic simplification or commutativity rule. Typed literal records distinguish model-supplied constants from observed values. Tracing the operands to their source observations determines the calculation's sources and image regions.

The observation records form the graph's nodes, and operand references form its dependency edges. The initial record set contains the images and text supplied to the policy. Each successful calculation adds its result with links to its operands. The record set grows throughout the rollout and retains one copy of each observation identity. Failed or invalid calls consume a budget unit and add no observation record.

Writing $\mathrm{evidence}(o_{i,t})$ for the observations and provenance extracted from executor output $o_{i,t}$, the update is

$$\mathcal{E}_{i,t+1} = \mathcal{E}_{i,t} \cup \mathrm{evidence}(o_{i,t}), \qquad \Delta\mathcal{E}_{i,t} = \mathcal{E}_{i,t+1} \setminus \mathcal{E}_{i,t}.$$

Thus $\Delta\mathcal{E}_{i,t}$ contains only newly added observation identities, giving the equivalent update in Eq. 2.

B.2 Evidence Levels and Matching

New evidence is identified from complete observation identities before any coarsening. The projections ψ_ℓ, $\ell = 0, 1, 2$, then retain progressively more detail from each graph node. At the source level, ψ_0 keeps source identities and observation types. At the region level, ψ_1 also keeps image regions. At the content level, ψ_2 keeps the complete observation identity, including the operator and canonical operands of a calculation. Each projection produces a sorted set of node identities. Removing fields and deduplicating recovers a coarser projection from a finer one, so the resulting groups are nested. For a transition, we project $\Delta\mathcal{E}$ directly. We do not subtract the projected contexts because doing so could hide a new observation that shares a coarse type with existing evidence.

The call signature $\omega_\ell(o)$ records the normalized operation, completion status, and returned observations at level ℓ. Crop, zoom, OCR, and extraction arguments refer to original-image regions, while calculation arguments refer to the executed operator and canonical operands. The full signature retains parameters that affect the operation's meaning. Errors use stable executor categories, with volatile tracebacks and addresses removed. A successful empty extraction therefore differs from a failed extraction, even though neither adds a new observation. The level-specific representations in Eq. 3 are $\mathcal{E}_{i,t}^{(\ell)} = \psi_\ell(\mathcal{E}_{i,t})$, $\Delta\mathcal{E}_{i,t}^{(\ell)} = \psi_\ell(\Delta\mathcal{E}_{i,t})$, and $o_{i,t}^{(\ell)} = \omega_\ell(o_{i,t})$.

Let $b_{i,t}$ be the remaining call budget before execution. At each level, the context identity combines this budget with prior evidence, while the transition identity also includes the call signature and new evidence. The resulting identities are

$$\begin{aligned} C_{i,t}^{\ell} &= \big(b_{i,t}, \psi_\ell(\mathcal{E}_{i,t})\big), \\ T_{i,t}^{\ell} &= \big(C_{i,t}^{\ell}, \omega_\ell(o_{i,t}), \psi_\ell(\Delta\mathcal{E}_{i,t})\big). \end{aligned} \tag{8}$$

Thus both identities include the budget before the call. Each attempt decreases this budget, so a rollout reaches a given value at most once. Repeating an observation gives $\Delta\mathcal{E} = \varnothing$, while the call signature still records the operation and returned value. The repeated call's credit is computed from the return comparisons defined above, with no additional repeat penalty.

Matching these identities across the other rollouts for the same input gives the groups in Eq. 3. Each peer rollout contributes at most one call at a given remaining budget.

B.3 Worked Example

Figure 1 uses a high-resolution report-page example. For 2019, the table gives audit fees of 17 million and total fees of 19 million. For 2018, the corresponding values are 21 and 26 million. The displayed task evaluates $\frac{17+21}{2} - \frac{19+26}{2} = -3.5$ million. The figure labels the 2019 and 2018 audit fees as A and B, the 2018 total fee as C, and the 2019 total fee as D.

The two histories apply localized OCR to the audit-fee row (A, B) and the 2018 total-fee cell C in different orders. Both retain the same three values and leave six calls in the eight-call budget. From this shared context, one next call calculates $\mathrm{mean}(A, B) = 19$, while the other applies OCR to obtain $D = 19$. The calculation creates a record with references to A and B, and the OCR call creates a record tied to a different page region. Thus, both calls add new evidence and return the same numeric payload, but their sources and dependencies define distinct transitions. If a later call reads D again with the same region and content, its executor output still contains 19, but its evidence increment is empty. The three cases therefore distinguish equal returned values, distinct observation identities, and repeated acquisition of an existing observation.

C Credit Derivations and Policy Implementation

C.1 Group Statistics

After level selection or fallback in Section 3.3, the selected groups have counts and reward sums

$$n_C = |\mathcal{C}|, \quad S_C = \sum_{j \in \mathcal{C}} R_j, \qquad n_T = |\mathcal{T}|, \quad S_T = \sum_{j \in \mathcal{T}} R_j. \tag{9}$$

These statistics give $V = S_C/n_C$ and $Q = (R_i + S_T)/(1 + n_T)$. The context count is positive because the fallback uses all $N - 1$ peer rollouts and $N \geq 2$. Appendix C.4 describes how the counts and sums are shared across target calls.

C.2 Missing Matches and Normalization

The reward mean and variance over the N sampled rollouts are

$$\bar{R} = \frac{1}{N}\sum_{i=1}^{N} R_i, \qquad s_R^2 = \frac{1}{N}\sum_{i=1}^{N}(R_i - \bar{R})^2.$$

If no transition group contains a match, $n_T = 0$ and $Q = R_i$. If no context group contains a match, the selected context mean is the leave-one-out mean $\bar{R}_{-i}$. Then

$$\frac{N-1}{N s_R}(R_i - \bar{R}_{-i}) = \frac{R_i - \bar{R}}{s_R}. \tag{10}$$

The normalized tool-segment advantage therefore equals the trajectory advantage. We use $N \geq 2$ and set all ProVIC advantages for an input to zero when $s_R \leq 10^{-6}$.

C.3 Contribution of the Target Reward

When $n_T > 0$, write $\bar{R}_T = S_T/n_T$ for the mean reward of the transition peers. Then the target-inclusive estimate can be expressed as

$$Q = \bar{R}_T + \frac{R_i - \bar{R}_T}{1 + n_T}. \tag{11}$$

The target's weight is therefore $1/(1 + n_T)$ and decreases as more transition matches become available. When no transition peer is available, $Q = R_i$, so the call can still be compared with a matched context baseline.

If every context peer also matches the transition, then $\mathcal{T} = \mathcal{C}$ and $\bar{R}_T = V$. For $s_R > 10^{-6}$, this gives

$$A_{i,t}^{\text{tool}} = \frac{N-1}{N s_R}\frac{R_i - V}{1 + n_C}. \tag{12}$$

Thus identical peer groups do not necessarily yield zero credit, because Q also contains the target reward while V excludes it. Both identities follow directly from Eq. 4 and do not require assumptions about the distribution of rewards.

C.4 Efficient Group Indexing

For each input and matching level, two hash tables index the context and transition identities in Eq. 8. Each entry stores the number of calls with that identity and the sum of their rollout rewards. Because a rollout reaches each remaining-budget value at most once, it contributes at most one call to an entry. The statistics for a target call are obtained by subtracting one from the entry count and R_i from its reward sum. This excludes the target from both peer groups, after which Q adds its reward back exactly once.

For M recorded calls and $L = 3$ levels, building and querying the tables takes expected $O(LM)$ work with cached keys and hash-table lookup. At most $2LM$ entries are needed before identical keys are merged. Evidence canonicalization and key construction are additional costs. This implementation shares group statistics across calls instead of scanning all other rollouts for every target.

C.5 Segment Policy Objective

An assistant segment starts when the policy begins its initial response or resumes generation after an executor observation. It ends when the policy issues a tool request, produces the final answer, or otherwise stops generating. The segment includes the reasoning generated before that endpoint. The executor's response is added to the next policy history. Executor observations, system text, and

padding are masked from the action loss. An invalid parsed request consumes a call and returns an error response. A recorded tool attempt therefore includes calls that return an executor error, even though they add no evidence node. A response that ends without a tool call receives the trajectory advantage.

Each rollout permits at most eight attempted calls followed by a final response. Each assistant turn has a separate 2,048-token generation cap. If a turn reaches the cap without a complete request or answer, the rollout ends with the verifier's failure reward. Equal remaining call budgets therefore imply equal remaining numbers of generation opportunities.

The objective in Eq. 5 averages over generated tokens within each segment, segments within each rollout, and rollouts. For the fixed SFT reference π_{ref}, the sampled KL penalty is $D_{i,k} = r_{i,k}^{\mathrm{ref}} - \log r_{i,k}^{\mathrm{ref}} - 1$, where $r_{i,k}^{\mathrm{ref}} = \pi_{\mathrm{ref}}(y_{i,k} \mid h_{i,k})/\pi_\theta(y_{i,k} \mid h_{i,k})$. The comparison protocol uses this averaging rule and KL convention for all RL methods. Empty generated segments are omitted. Zero segment advantages suppress the surrogate term while leaving the reference-policy penalty in the objective. Groups, return estimates, and advantages are computed from the old-policy batch and held fixed during optimization.

D Experimental Details

D.1 Training Data

Training prompts come from public training partitions of SlideVQA, MP-DocVQA, DUDE, ChartQA, Visual-CoT, FinQA, and TAT-QA (Tanaka et al., 2023; Tito et al., 2023; Van Landeghem et al., 2023; Masry et al., 2022; Shao et al., 2024a; Chen et al., 2021; Zhu et al., 2021). We partition source documents, slide decks, and images into separate SFT and RL pools before constructing prompts. Table 5 defines four data strata by the evidence needed to answer a question. Each stratum contributes 3,000 RL prompts and 1,000 SFT prompts. Items meeting multiple criteria are assigned to the Mixed stratum before the single-criterion strata are sampled. The resulting data pools contain distinct prompts from disjoint source pools.

Table 5: **Training data by evidence stratum.** Each row specifies its public sources, admission rule, and RL sampling quota after source partitioning and filtering. Mixed items satisfy at least two rules and are assigned before sampling the single-rule strata.

Stratum	Public training sources	Admission rule	RL
Source	SlideVQA, MP-DocVQA, DUDE	A released page or slide annotation identifies the answer's source.	3,000
Region	ChartQA, Visual-CoT	A released box or chart-entity annotation identifies localized evidence.	3,000
Calculation	FinQA, TAT-QA, and numerical SlideVQA	A released program or expression identifies the answer's operands.	3,000
Mixed	Sources above	At least two of source, region, and calculation are required in one item.	3,000

Rendering and supervision. Multi-page examples retain the released page order and distractor pages. Spatial annotations are expressed in original-image pixels. For FinQA and TAT-QA, we render tables and associated text as page images and record the boxes of cells and text blocks. Executing the released programs gives calculation routes with verified answers. These annotations support SFT construction and terminal verification. During RL, matching uses only observations returned by the executor.

For 500 SFT prompts in each stratum, we construct two valid tool-use trajectories, and the remaining 500 prompts receive one trajectory each. This gives $4(500 \times 2 + 500) = 6{,}000$ trajectories. Each contains tool requests, returned observations, and a final answer. Where possible, the two routes use different pages, crops, or extraction paths. We retain a trajectory if all calls execute successfully and its answer passes the released task verifier.

Source separation. We exclude records from the primary benchmarks before sampling, including the MME-RealWorld records used by O3-Bench (Li et al., 2026b). Filtering across pools removes duplicate source identifiers, image digests, and normalized question–answer records. Near-duplicate image and OCR matches are reviewed during corpus construction. For each source group, the manifest records the dataset release, license, and assigned split.

Terminal verification. Exact and multiple-choice answers use normalized matching. Numerical answers follow the source task's tolerance and scale. When a reference program is available, its executed result supplies the answer for verification. The task verifier supplies the only reward. Call count, dependency depth, and evidence-group labels add no reward terms.

D.2 Tools and Interaction Protocol

Table 6 specifies the seven tool interfaces. All listed arguments are required, and a null region selects the whole view. The accompanying protocol files contain the complete JSON schemas. A view identifier names an image returned to the policy, while an observation identifier refers to a previously returned evidence item. The accompanying executor design uses Pillow for image loading, cropping, and resizing, Python decimal arithmetic for calculation, and a shared frozen Qwen3-VL-8B-Instruct backend for OCR, structured extraction, and chart reading. The perception backend receives the requested image region and tool arguments and uses greedy decoding.

Table 6: **Tool interfaces.** Boxes use half-open pixel coordinates in the named view. The executor maps them to original-image coordinates for matching. Observation references in calculations preserve operand order.

Tool	Arguments	Returned evidence
`select_page`	`source_id`	One source-page image view
`crop`	`view_id`, `box`	A cropped image view
`zoom`	`view_id`, `scale`	A resized view, with $1 < \texttt{scale} \leq 4$
`ocr`	`view_id`, `box`	Text blocks and recognized numbers
`extract`	`view_id`, `box`, `fields`	Typed fields with supporting regions
`read_chart`	`view_id`, `box`, `query`	Structured chart evidence
`calculate`	`operator`, ordered `operands`	A numerical result linked to its inputs

An extraction field specifies its name, type, and description. Calculations use ordered references to returned numeric observations and typed constants, preserving the dependencies needed for provenance.

The policy-visible response contains the operation, completion status, returned observations, and remaining call budget. Image observations are delivered with their image blocks. Source coordinates, view transforms, and operand links are retained in a separate executor record stream for ProVIC matching. Thus the policy observes the same interface under every credit method. Every attempted request consumes one call, including a malformed or failed request, while failed calls add no evidence. A repeated observation remains visible but adds no duplicate graph node.

D.3 Prompt Template and Interaction Example

The backbone chat template combines the system prompt in Table 7 with the tool definitions in Appendix D.2. The initial user message supplies the question, ordered source catalog, image blocks, and remaining call budget. Each view in the catalog includes its identifier and pixel dimensions.

Illustrative interaction. Table 8 completes the numerical example in Figure 1. It is an author-constructed protocol example, rather than a sampled policy trajectory. The supplied page and question form the initial evidence state. Letters abbreviate the observation identifiers, and the accompanying JSON example gives the requests, responses, and provenance records.

Table 7: **System prompt.** The same prompt and tool definitions are used for all RL methods.

You answer questions about the supplied images and documents. Use the visible evidence and the available tools when needed. At each turn, briefly state the next useful step, then either request one tool or give the final answer. Do not issue parallel tool calls.
Use the supplied source and view identifiers. Image coordinates are half-open pixel boxes [x0, y0, x1, y1] in the specified view. Read a smaller region if a tool reports that its output is too large. For calculations, refer to the observation identifiers returned by tools. Use literal operands only for constants in the question or formula. Preserve operand order and the units of observed values. Do not invent observations or observation identifiers. An attempted tool call consumes one unit of the displayed remaining budget, including a failed call. Use the error message to revise an invalid request. When no calls remain, provide your best final answer from the available evidence without requesting another tool.
Write a tool request as one JSON object with name and arguments, enclosed in <tool_call> and </tool_call>. Otherwise, write the final answer inside <answer> and </answer>, retaining any required units.

Table 8: **A complete illustrative tool interaction.** The page contains audit fees $A = 17$, $B = 21$ and total fees $D = 19$, $C = 26$. The request column abbreviates image regions by these cell labels. The budget is measured before each call.

Call	Budget	Request	Returned observations
1	8	OCR of the audit-fee row	$A = 17, B = 21$
2	7	OCR of cell C	$C = 26$
3	6	$\mathrm{mean}(A, B)$	$E = 19$
4	5	OCR of cell D	$D = 19$
5	4	$\mathrm{mean}(D, C)$	$F = 22.5$
6	3	$\mathrm{subtract}(E, F)$	$G = -3.5$

The final response is -3.5 million, with two calls unused. Although E and D both contain 19, E depends on A and B, whereas D comes from another image region. They therefore enter the graph as distinct observations.

D.4 Training and Decoding Settings

Table 9 gives the common configuration. All RL methods initialize from the same SFT checkpoint and use it as the fixed reference policy. Seeds $\{11, 23, 37, 53, 71\}$ are paired across methods for prompt order, adapter initialization, and rollout random-number streams. The resulting trajectories may diverge as the policies change.

All methods share image preprocessing and the limits in Table 9. To reserve space for the next assistant turn, we remove the oldest complete assistant–tool exchanges as needed, retaining the system prompt, tool definitions, initial input, and most recent exchange. If this retained context exceeds the limit, the rollout is counted as a failure. Credit computation retains the complete evidence graph after truncation. Tool responses exceeding the output cap return an error and add no evidence.

D.5 Baseline Adaptation

We apply local credit to assistant segments ending in a tool request. Other segments use the trajectory advantage in Eq. 1. HGPO and BiPACE form comparison pools within the same input and remaining call budget, with the target record included in their group statistics. All methods share the policy-visible image-text history, executor, verifier, reference policy, and segment averaging.

HGPO observation and history. The state before a call is the current observation delivered to the policy, consisting of the initial input at the first call or the latest executor response thereafter. Its representation preserves the ordered text and image blocks, including their boundaries. Text is serialized deterministically, and each image retains its dimensions, color mode, and a digest of its decoded pixels. Temporary image handles resolve to these visible image identities. Observation handles resolve to their visible typed contents, removing rollout-local names without introducing hidden provenance. The depth-k representation adds the preceding k tool request–response exchanges in order, including failed attempts and repeated observations. This representation uses no

Table 9: **Training and decoding configuration.** The common settings apply to all matched RL comparisons. The table also reports the hardware and training framework.

Item	Setting
Primary policy	Qwen3-VL-8B-Instruct
Updated parameters	Language LoRA rank / scale / dropout $64/128/0.05$ (vision frozen)
SFT	6,000 trajectories, 2 epochs, sequence cap 4,096, global batch 128
RL prompts / prompt batch	12,000 / 64, one pass with 188 updates (32 prompts in the final batch)
Rollouts per prompt	16 (1,024 trajectories in a full update batch)
Interaction budget	At most 8 attempted tool calls, with 2,048 generated tokens per assistant turn
Image pixels	65,536–1,310,720 per image, with aspect ratio preserved
RL / evaluation context	32,768 tokens including text, visual tokens, and the next assistant turn
Tool response cap	1,024 non-image tokens and at most one image per call
Training decoding	Temperature 0.7, top-p 0.8
RL evaluation decoding	Greedy, with the same per-turn and call budgets as RL training
SFT / RL optimizer	AdamW with learning rates 10^{-5} / 5×10^{-7}
AdamW settings	$(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.1
Schedule / gradient clipping	Cosine with 3% warmup, global gradient norm 1.0
Policy epochs per rollout batch	1
PPO clipping / reference KL	$\epsilon = 0.2$ / $\beta_{\mathrm{KL}} = 0.01$
Precision	bfloat16
GPU model / count	NVIDIA H100 / 8
Training framework	verl

accumulated evidence graph or hidden provenance fields. We use a maximum history depth of four for the complete baseline and two for the three-level representation ablation.

HGPO computes a return-relative advantage at each available history depth, by using the group mean and population standard deviation plus 10^{-6}. We use the paper's linear depth weights, proportional to $k + 1$, and renormalize them over levels with nonzero local advantage (He et al., 2026). The tool segment receives this weighted advantage, with zero assigned if all levels give zero. No additional trajectory term is added to HGPO's tool credit.

BiPACE state representation. We use BiPACE's Actor-Hidden representation and Q-style estimator (Wang et al., 2026). The sampling VLM processes the actual multimodal history before the current assistant segment, including the input images and returned crops under the common preprocessing. We extract the last non-padding prompt token from `hidden_states[-8]` and normalize the vector to unit Euclidean norm in float32. This position precedes the current segment's reasoning and tool request.

For each run, the first training rollout batch calibrates the cosine-distance threshold by using only the state representations. Following BiPACE's geometric calibration, we search the interval $[0.02, 0.40]$ by at most six bisection probes, stopping when the median cluster size is between four and eight. If no probe reaches this range, we select the probed threshold whose median is closest to six, breaking ties toward the smaller threshold. A batch with no tool calls uses the published threshold of 0.10. The threshold then remains fixed for that run. Clustering follows rollout order and uses the published nearest-centroid assignment and online centroid update.

BiPACE tool actions and credit. An action identity contains the parsed tool name and all typed arguments. Object fields are sorted, while positional, array, and operand order are preserved. Image and view references resolve to the visible asset identities used above. Observation references similarly resolve to their visible typed contents. Unparseable requests use their raw request text as the action identity. The action identity excludes the current executor output, completion status, and

evidence graph. Within a state cluster, the Q-style estimate subtracts the full-cluster mean return from the same-action mean, including the target in both means, and divides by the cluster population standard deviation plus 10^{-6}. Singleton and uniform-action clusters give zero local advantage. BiPACE adds this local advantage to the trajectory advantage with weight one.

The complete HGPO and BiPACE baselines retain the published discount factor of 0.95 for local returns. We index generated segments in chronological order and place the verifier reward at the last segment, so segment s has local return $0.95^{|\mathcal{S}_i|-s}R_i$. BiPACE's trajectory term uses the undiscounted terminal reward. The terminal-reward variance threshold in Table 9 does not suppress either baseline's local advantage when discounted returns differ. The common settings in Table 9 apply to both adapters. Representation extraction and threshold calibration are included in training cost.

E Ablation and Analysis Protocols

E.1 Representation and Estimator Comparison

The representation comparison uses three nested levels. All four configurations estimate credit from the same undiscounted terminal rewards R_i. Thus the crossed comparison holds the return definition fixed while varying the grouping representation and credit estimator. History uses the observation and request serialization in Appendix D.5. For History grouping, level 0 matches the current environment observation as serialized in the policy-visible messages. Level 1 additionally matches the immediately preceding tool exchange, and level 2 also matches the tool exchange before it. Each exchange contains the assistant's tool request and the serialized executor response. Evidence uses the three context identities $C^0_{i,t}, C^1_{i,t}, C^2_{i,t}$ from Eq. 8. Both representations include the remaining call budget.

For HGPO, the selected representation defines the context groups. The two representation rows use the same aggregation rule, so the group representation is the only factor varied between them. For ProVIC, the representation defines the same context groups. History transitions append the executed request and returned observation. Evidence transitions use the normalized executor output and newly added evidence. Thus both Evidence configurations receive identical provenance records. All four configurations use the same optimizer and three analysis benchmarks. The three-level History–HGPO variant isolates the grouping comparison, while the complete HGPO baseline uses the four-exchange history defined in Appendix D.5.

E.2 Component Ablations

Recent evidence. This control retains the initial observations and evidence from the last three recorded tool attempts, including their traced operand dependencies. Failed attempts count toward this three-attempt window. An observation outside the retained set can count as new when read again, while older records retained as operand dependencies remain available. All other context and transition fields and the return estimator follow ProVIC. This variant compares complete evidence retention with a dependency-preserving three-call window.

Context baseline only. This control keeps ProVIC's selected context baseline and sets $\mathcal{T} = \varnothing$ for every tool segment. The advantage becomes $(N-1)(R_i - V)/(Ns_R)$. This compares the observed reward with the context baseline without using rewards from the transition group.

Source level only. When $\mathcal{C}_0$ is nonempty, this control fixes $\mathcal{C} = \mathcal{C}_0$ and $\mathcal{T} = \mathcal{T}_0$. If $\mathcal{C}_0$ is empty, it uses the same GRPO fallback as ProVIC. This removes the region and content refinements while retaining the same matched-return calculation.

Content level only. When $\mathcal{C}_2$ is nonempty, this control fixes $\mathcal{C} = \mathcal{C}_2$ and $\mathcal{T} = \mathcal{T}_2$. If $\mathcal{C}_2$ is empty, it uses the GRPO fallback rather than backing off to region or source matches. This tests the value of the finest-available backoff rule.

Excluding the target reward. When $\mathcal{T}$ is nonempty, this control uses $Q = |\mathcal{T}|^{-1}\sum_{j\in\mathcal{T}} R_j$, excluding the target reward R_i. When $\mathcal{T}$ is empty, it retains $Q = R_i$. This tests whether including the target's observed outcome improves the transition comparison.

E.3 Fixed-History Decision Replay

We select 200 successful call positions from rollouts generated by the common SFT policy and retain the policy history immediately before each selected call. The analysis prompts are source-disjoint and balanced across the four data strata. Histories are selected before credit is computed. From each retained history h, we sample $N = 16$ complete continuations to form a shared batch for all credit rules. We keep up to four distinct candidate calls. When a request appears more than once, we average its assigned credit.

For each candidate call a, we restore history h, execute the request, and sample 16 fresh continuations from the same policy. Let $R_k^{\text{replay}}(h, a)$ denote the terminal reward of continuation k. Their mean return, $\widehat{U}(h, a) = 16^{-1} \sum_{k=1}^{16} R_k^{\text{replay}}(h, a)$, estimates the outcome of taking that call and continuing with the remaining budget. These continuations are independent of the batch used to compute credit. Histories with fewer than two distinct candidates contribute no comparison pair.

Replay agreement compares the credit ranking of a pair (a, a') with the sign of $\widehat{U}(h, a) - \widehat{U}(h, a')$. Pairs with equal replay returns are omitted. For an untied pair, a correct credit ranking scores 1, an incorrect ranking scores 0, and equal credit scores $1/2$. We average pair scores within each history and then across histories with at least one retained pair. The retained history and pair counts are 188 and 924. All candidate comparisons therefore share the full policy history and remaining budget. Higher agreement indicates closer alignment with the sampled continuation rankings under the common SFT policy. Appendix H discusses the scope of this diagnostic.

E.4 Required-Evidence Groups

We use a 2,000-question analysis subset from VisFinEval, ChartQAPro, and O3-Bench, grouped by evidence requirements. Before inspecting model trajectories, an annotator uses the source image and reference evidence to identify a sufficient route to the answer. The four groups require (1) no observation beyond the supplied view, (2) one localized observation, (3) multiple regions or one calculation on observed values, or (4) multiple source images or dependent calculations. The highest applicable group takes precedence. A second annotator resolves ambiguous cases before model outputs are scored. The labels describe the task's requirements independently of the policy's chosen calls.

Within each group, we compute the official primary metric for each represented benchmark and average these scores equally. Paired method differences use the same questions and weights. We report the group size alongside its scores. The resulting breakdown describes performance across different evidence requirements. Reference routes and group labels are used only for analysis. The group counts show how many questions support each comparison.

E.5 Statistics and Cost

For each RL method, we average each benchmark score over five training seeds. The primary mean first averages benchmark scores within each seed, then averages the five seed means. We report the sample standard deviation across the five seed means.

Training cost includes rollout generation, tool execution, optimizer updates, and method-specific auxiliary computation. The shared policy and reference-policy computations remain part of RL training cost. Auxiliary baseline work, including BiPACE's representation extraction, is included under the same accounting rule. Independent continuations for decision replay are additional diagnostic work, distinct from online credit estimation. ProVIC's evidence indexing and matched-return computation are included in online iteration time.

Table 13 reports inference cost as the average number of tool calls and generated tokens over all evaluated questions, including failed rollouts. Although credit estimation is absent at inference, a learned policy can change both quantities. These usage counts describe policy behavior, while inference latency would also depend on the tools and serving system.

F Additional Results

F.1 Results Across Training Seeds

Table 10 reports the primary mean for every training seed. Each column pairs runs from the same seed, and the last column summarizes the five seed means.

Table 10: **Primary mean across training seeds.** Each seed column gives the equally weighted five-benchmark mean of one 8B policy. The final column reports their mean and sample standard deviation. All rows use the matched training and evaluation protocol.

Method	11	23	37	53	71	Mean $\pm$ SD
GRPO	56.2	56.3	56.4	56.5	56.6	56.4 ± 0.2
HGPO	56.7	56.9	57.0	57.1	57.3	57.0 ± 0.2
BiPACE	56.8	56.9	57.1	57.2	57.3	57.1 ± 0.2
ProVIC	58.0	58.3	58.9	59.1	57.9	58.4 ± 0.5

F.2 Supplementary FinMME Metrics

Table 11 reports supplementary FinMME metrics. FinScore averages question scores within each financial domain and weights domains equally before applying a penalty for incorrectly selected answer options (Luo et al., 2025). We report FinScore separately and use average accuracy in the five-benchmark mean.

Table 11: **Supplementary FinMME metrics.** FinMME average accuracy and supplementary FinScore for Qwen3-VL-8B-Instruct policies.

Method	FinMME Acc.	FinScore
GRPO	51.6	20.5
HGPO	52.1	21.1
BiPACE	52.3	21.3
ProVIC	54.0	22.0

F.3 Performance by Required Visual Evidence

Because ProVIC forms comparison groups from accumulated evidence, we test whether its gain over GRPO varies with the evidence required by a question. We group a 2,000-question analysis subset from VisFinEval, ChartQAPro, and O3-Bench into four evidence-requirement categories, ranging from no observation beyond the supplied view to multiple sources or dependent calculations. By using the source images and evidence associated with the reference answers, we assign each question its highest applicable category before examining model outputs. As shown in Table 12, ProVIC's advantage over GRPO grows across these categories and is largest for questions requiring multiple sources or dependent calculations. Because the categories contain different questions, this comparison is descriptive.

Table 12: **Performance by required visual evidence.** The 2,000-question analysis subset from VisFinEval, ChartQAPro, and O3-Bench is grouped by the highest applicable evidence requirement. The groups are assigned before model outputs are examined. Within each group, we compute each benchmark's official metric by using its item weights and then average the represented benchmarks equally. n is the total question count across benchmarks in each category. Δ is the paired difference on the same questions, computed as ProVIC minus GRPO. Positive values favor ProVIC.

Required evidence	n	GRPO	HGPO	**ProVIC**	Δ
No additional observation	620	58.7	58.9	59.1	+0.4
One localized region	580	56.9	57.4	58.0	+1.1
Multiple regions / one calculation	510	51.8	52.6	53.6	+1.8
Multiple sources / dependent calculations	290	43.7	45.0	46.5	+2.8

F.4 Training and Inference Cost

To assess the cost of these gains, we report the training overhead of ProVIC's online evidence indexing and matched-reward computation in Table 13. All four methods train on eight NVIDIA H100 GPUs. Relative to GRPO, ProVIC adds 60 seconds per iteration and 25 GPU-hours over the full run. Because credit estimation is not used at inference, we separately compare the behavior of the learned policies. The ProVIC policy uses 0.31 fewer tool calls and 76 fewer generated tokens per question than GRPO. Together with Table 1, these measurements characterize the performance–cost tradeoff.

Table 13: **Training and inference cost by RL method.** Iteration time averages online RL iterations, while total GPU-hours include auxiliary computation. Calls and generated tokens are averaged over all questions in the primary benchmarks, including failures.

Method	Iteration (s)	Total GPU-h	Calls / question	Tokens / question
GRPO	2,840	1,187	5.02	1,180
HGPO	2,880	1,203	4.88	1,146
BiPACE	2,890	1,207	4.80	1,129
ProVIC	2,900	1,212	4.71	1,104

G Mechanism Diagnostic Protocols

This section describes the two diagnostics in Figure 3.

G.1 Comparisons Across Acquisition Histories

Both grouping variants share 3,200 complete SFT-policy rollouts sampled from 200 held-out inputs, with $N = 16$ rollouts per input. History uses the three-level representation in Appendix E. Both variants use the ProVIC return estimator and level-selection rule.

Independent return comparison. For each sampled successful call after the first attempt, let h be its original history before the tool segment and z the complete segment, including its reasoning and request. We estimate the return after forcing z and the return from freely continuing from h, each with 16 fresh continuations under the same policy and remaining budget. Their difference is

$$\widehat{d}(h, z) = \frac{1}{16}\sum_{k=1}^{16} R_k(h, z) - \frac{1}{16}\sum_{k=1}^{16} R_k(h). \tag{13}$$

Both continuation batches are independent of the credit batch and of each other.

Agreement and acquisition order. The reported metric is sign agreement between the assigned credit and $\widehat{d}$. Matching signs score 1, opposite signs score 0, and zero credit scores $1/2$ when $\widehat{d} \neq 0$. Calls with $\widehat{d} = 0$ are omitted from agreement. The different-order subset requires a peer with the same full evidence state and budget but a different ordering of the same canonical acquisition events. Membership is determined before inspecting rewards. Both methods are scored on the same targets, including their fallbacks.

G.2 Contribution of Observation Provenance

The provenance diagnostic shares the same rollout batches, targets, independent returns, and ProVIC estimator. Both conditions accumulate all observations independently of acquisition order and use fixed exact matching with the same GRPO fallback. The full-identity condition uses content-level-only matching.

Removing provenance consistently. The value-only condition omits source, region, producing operation, and operand identities from grouping. It reconstructs evidence sets and evidence increments by using these value-only identities, so duplicate detection follows the same rule. Its call signature retains completion status and typed returned values without recovering provenance through tool arguments or handles. The policy-visible history and actual tool execution remain identical for both conditions. Value-collision cases are identified when value-only transition matching merges a target and a peer that return the same typed values from different origins.

Controlling the number of peers. On collision cases with nonempty full-identity context and transition groups, the value-only groups are also subsampled to the corresponding full-identity counts. We sample transition peers first and then fill the context group, preserving $\mathcal{T} \subseteq \mathcal{C}$ and excluding the target. Sampling is independent of rewards and repeated 100 times. The full-identity and subsampled value-only conditions are evaluated on the same retained calls.

Aggregation. Per-input agreement averages its scored calls, followed by an equal average over inputs. Peer-subsampling repetitions are averaged within each call and are not counted as additional independent observations. Figure 3 reports point estimates of agreement.

H Discussion and Limitations

Provenance and group support. ProVIC relies on executor-supplied source, region, and operand information. Canonicalization aligns temporary views and equivalent serializations, but cannot recover missing provenance or correct an erroneous observation. Coarser matching increases support by retaining fewer distinctions, while rare states may still require the GRPO fallback. This fallback defines an advantage without providing additional evidence about the target call.

Interpretation and evaluation scope. Matching evidence and budget leaves possible differences in the policy's ordered history and earlier reasoning. The return comparison therefore measures an association under the sampled policy, rather than identifying a causal action effect. Fixed-history replay controls the prior history, but its finite continuations remain noisy and are sampled under the SFT policy. The training experiments cover one 8B backbone, five benchmarks, and a fixed interaction budget. Transfer to larger models, longer tool-use horizons, or executors with incomplete provenance remains open.