Anchored Reasoning on Tables: Bridging the Verifiability Gap in Multimodal Reinforcement Learning

Anonymous ACL submission

Abstract

The Verifiability Gap in table reasoning arises when multimodal LLMs produce correct answers while grounding intermediate steps in unsupported visual evidence. Intermediate visual claims are often not exposed in a symbolically addressable form, making standard RL with verifiable rewards unable to reliably inspect whether a reasoning step is actually supported by the image. We mitigate this gap for table reasoning with Anchored Reasoning, which turns each reasoning step into a checkable (coordinate, value, structural role) claim over the table grid. We instantiate this paradigm as V-tableR1: a policy VLM generates anchor-laden visual chains-of-thought, while a learned critic VLM estimates anchor fidelity against the source image. The policy is trained with PGPO, a GRPO-based process-guided formulation that integrates critic-gated trajectory rewards with stabilization mechanisms for long anchor-laden rollouts. V-tableR1 (4B) achieves the best open-source average accuracy across seven tabular reasoning benchmarks, obtains state-of-the-art results on several individual datasets, surpasses open-source models up to $18\times$ its size in aggregate, and improves over its SFT baseline by up to 10.66% absolute accuracy.

1 Introduction

Vision-language models (VLMs) fail systematically on tables: they can read individual cells accurately, yet often collapse when those readings must be composed into multi-step derivations (Favero et al., 2024; Rawte et al., 2025; Zhao et al., 2025; Bai et al., 2025a). This failure persists even in frontier models: correct final answers can coexist with demonstrably wrong intermediate trajectories (Basu and Chakraborty, 2026; Wu et al., 2025; Jin et al., 2022; Sun et al., 2020), showing that scale alone cannot replace process-level supervision.

We identify the root cause as the *Verifiability Gap*: the inability of outcome-level or text-only verifiers to reliably confirm whether an intermediate reasoning step is genuinely grounded in the image region it purports to reference. In symbolic domains such as mathematics and code, each reasoning step is a verifiable object—checkable against an identity, a unit test, or a logical predicate—a property that underlies the success of Reinforcement Learning with Verifiable Rewards (RLVR) (Shao et al., 2024; Yu et al., 2025) and Process Reward Models (PRMs) (Cui et al., 2025). In the visual domain, this property becomes much weaker: even when OCR metadata confirms that a value exists in the table, an outcome-level verifier cannot reliably determine whether the model's reasoning *actually derived* that value from the claimed structural region. Consequently, naïve transfers of RLVR to vision reduce to outcome-only optimization (Su et al., 2025; Mroueh, 2025), while chain-of-thought distillation merely instills the *surface syntax* of reasoning, leaving models free to hallucinate intermediate steps provided the terminal answer is correct (Geirhos et al., 2020). The Verifiability Gap *precedes* the problem PRMs address: visual evidence must first be exposed through a sufficiently structured interface before step-level verification techniques can be applied reliably.

Tables provide a natural setting for reducing this gap: every cell carries a coordinate, every value belongs to a header, every aggregation operates over a definite operand set (Zhao et al., 2024; Laurençon et al., 2024). Unlike natural scenes, the deterministic grid geometry of tables converts visual evidence into symbolically auditable objects, making tables a structurally controlled testbed for verifiable multimodal reasoning.

We exploit this structure to propose Anchored Reasoning: a paradigm requiring every reasoning step to emit an auditable (coordinate, value, structural role) anchor that binds the step to a checkable region of the source image. We focus on tables as the first instantiation of this paradigm; extensions to other structured visuals

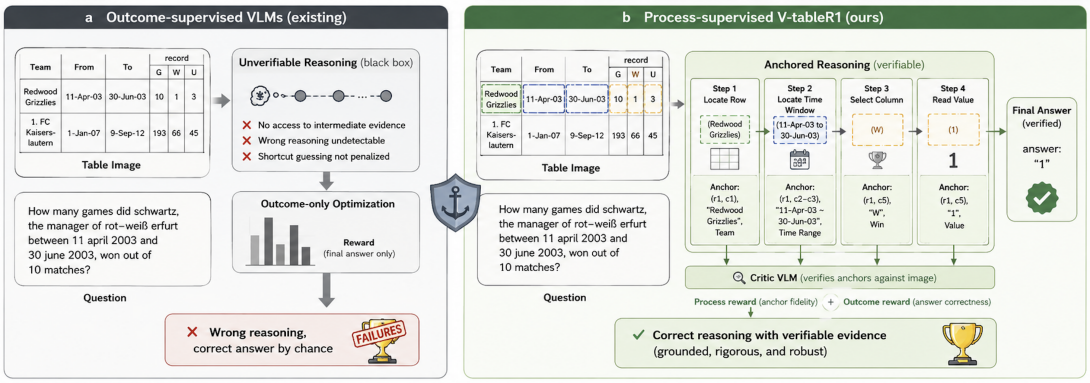

Figure 1: Outcome supervision vs. Anchored Reasoning. **(a)** Existing VLMs reward only the final answer, leaving intermediate reasoning indistinguishable from shortcut guessing. **(b)** V-tableR1 binds every step to a verifiable anchor (coordinate, value, role), scored by a critic VLM to produce an anchor-aware process reward under PGPO.

are sketched in Appendix E and left to future validation. We realize this paradigm as **V-tableR1**, coupling a policy VLM that generates anchor-laden Visual Chain-of-Thought (V-CoT) with a critic VLM that scores anchor fidelity against the source image to gate outcome rewards. Figure 1 contrasts this process-supervised design with conventional outcome-supervised VLMs. The policy is trained with **Process-Guided Policy Optimization (PGPO)**, a GRPO-based formulation that integrates critic-gated trajectory rewards with decoupled clipping (Yu et al., 2025) and length-aware sampling (Chen et al., 2025), targeting the three failure modes our framework identifies: faithful reasoning, visual hallucination, and shortcut guessing.

V-tableR1 (4B) achieves the strongest open-source average performance across seven tabular benchmarks, outperforming open-source models up to 18× its size in aggregate, matching closed-source frontier systems on fact verification, and improving over its SFT baseline by up to **10.66%** absolute accuracy. Our contributions are:

- **Verifiability Gap & Anchored Reasoning.** We formally identify the Verifiability Gap—the missing symbolic interface between visual evidence and reasoning steps—and propose Anchored Reasoning to narrow it via explicit, verifiable visual anchors.
- **V-tableR1 framework.** A process-guided RL framework coupling a policy and critic VLM through anchor-aware trajectory rewards, enabling group-relative optimization to penalize hallucination and shortcut guessing.
- **Parameter-efficient inference.** V-tableR1 (4B) surpasses the evaluated 72B-parameter open-source inference-time policies in average TFV/TQA performance and achieves the strongest open-source result on several individual benchmarks, suggesting that process supervision can complement parameter scaling for structured visual reasoning.

2 Related Work

2.1 Structured Visual Reasoning

Multimodal models have been extended to tables via SFT on large image–text corpora (Zheng et al., 2024; Yang et al., 2022), chain-of-thought distillation (Chen et al., 2024b), and table-as-program representations (Wang et al., 2024b). Despite their differences, these approaches share a single training signal: the likelihood of the terminal answer, leaving intermediate derivations unconstrained. As a result, models learn to associate question patterns with answer tokens without exposing which structural region supplies their evidence—a regime under which hallucinated rows, misattributed headers, and shortcut aggregation are indistinguishable from genuine reasoning (Ji et al., 2024). V-tableR1 departs fundamentally: rather than decorating outcome supervision with chain-of-thought syntax, it elevates the *structural provenance* of every reasoning step to a first-class supervised quantity via Anchored Reasoning.

2.2 Reinforcement Learning with Verifiable Rewards (RLVR)

RLVR has delivered step-change gains in language model reasoning by replacing dense reward models with symbolic, rule-based verifiers (Shao et al., 2024; Vojnovic and Yun, 2025; Yue et al., 2025),

with GRPO and DAPO further stabilizing long-horizon optimization via group-relative advantages and decoupled clipping (Yu et al., 2025; Chen et al., 2025). The precondition underlying all these successes—that each intermediate step is a *symbolically inspectable object*— silently fails in the visual domain. A claim such as "the value in row 2 is 4.2" is not self-verifiable: OCR metadata may confirm the value exists, yet no verifier can determine whether the model genuinely derived it from the claimed structural region. Existing multimodal RLVR therefore defaults to outcome-only rewards (Su et al., 2025; Mroueh, 2025), inheriting precisely the shortcut-learning pathologies it was designed to mitigate. PGPO treats the *construction* of this missing structural reference—not its exploitation—as the central design problem.

2.3 Process Supervision and Faithful Reasoning

Text-domain PRMs have established that step-level credit assignment yields substantially more robust reasoning than outcome-only optimization (Lightman et al., 2024; Wang et al., 2024a; Cui et al., 2025), closely linked to the broader question of chain-of-thought faithfulness (Lyu et al., 2023; Turpin et al., 2023). Extending this machinery to vision, however, confronts a categorical obstacle: these methods presuppose that the verification target already exists as a symbolic object. Recent multimodal attempts either retreat to synthetic environments with hand-crafted rule checkers (Zhang et al., 2025; Jiao et al., 2024), or evaluate visual reasoning at the response level rather than the step level (Wang et al., 2025)—both of which circumvent rather than resolve the absence of a step-level verification target. Anchored Reasoning addresses this prerequisite directly: by requiring each reasoning step to emit a verifiable visual anchor expressed in the table's intrinsic grid coordinates, it *constructs* the reference target that step-level visual verification has always assumed but never possessed—thereby making the dense-feedback machinery of text PRMs applicable, more directly, in the multimodal regime.

3 Method

We instantiate Anchored Reasoning as **V-tableR1**, a process-guided RL framework in which every reasoning step is bound, by construction, to a verifiable structural reference in the source document. The framework consists of three coupled components: an anchored reasoning protocol that constrains the policy VLM π_θ to emit step-level visual anchors (Sec. 3.2), a critic VLM C_ϕ that scores anchor fidelity to produce an anchor-aware process signal (Sec. 3.3), and **Process-Guided Policy Optimization (PGPO)**, a GRPO-based optimization formulation that turns this signal into a stable policy-learning objective (Sec. 3.4). An overview is shown in Figure 2.

3.1 Problem Formulation

Let a policy π produce a reasoning trajectory $y = (s_1, \ldots, s_K, \hat{y})$ on visual input x with query q, and let $\Phi : (s_k, x) \mapsto \{1, 0, \bot\}$ be an idealized step-level verifier that returns 1 when s_k is grounded in x, 0 when contradicted, and $\bot$ when s_k exposes *no symbolically inspectable reference* to x. We define the *Verifiability Gap* of π on a distribution $\mathcal{D}$ as

$$\mathrm{VG}(\pi; \mathcal{D}) \;=\; \mathbb{E}_{(x,q)\sim\mathcal{D},\, y\sim\pi}\left[\frac{1}{K}\sum_{k=1}^{K} \nVdash[\Phi(s_k, x) = \bot]\right]. \tag{1}$$

Outcome-only RLVR optimizes $\mathbb{E}[r(\hat{y})]$ while leaving VG unconstrained, rendering faithful (P1) and shortcut-correct (P3) trajectories observationally identical. **Anchored Reasoning** drives $\mathrm{VG} \to 0$ by construction: it requires each step to emit an auditable anchor $a_k = (\mathcal{C}_k, \mathcal{V}_k, \rho_k)$ such that $\Phi(s_k, x) \neq \bot$, recasting step-level verification as a checkable predicate over the source image.

3.2 Anchored Reasoning Protocol

Anchored trajectories. Given a table image x and a natural-language question q, the policy π_θ autoregressively emits a trajectory

$$y = (s_1, a_1, s_2, a_2, \ldots, s_K, a_K, \hat{y}), \tag{2}$$

where s_k is the k-th logical step (a natural-language proposition), a_k is its *visual anchor*, and $\hat{y}$ is the final answer. Each anchor is a structured triple

$$a_k = (\mathcal{C}_k,\ \mathcal{V}_k,\ \rho_k), \tag{3}$$

where $\mathcal{C}_k = \{(r_{k,m}, j_{k,m})\}_{m=1}^{M_k}$ denotes the referenced cell(s) or region(s), $\mathcal{V}_k$ is the associated value set, and $\rho_k \in \mathcal{R}$ is the assigned structural role, with $\mathcal{R}$ including roles such as `header`, `data`, `operand`, and other document-specific roles. This unified anchor covers single-cell lookup, comparison, filtering, and aggregation, while preserving structural coordinates for resolution-invariant verification across rendering styles. We provide the

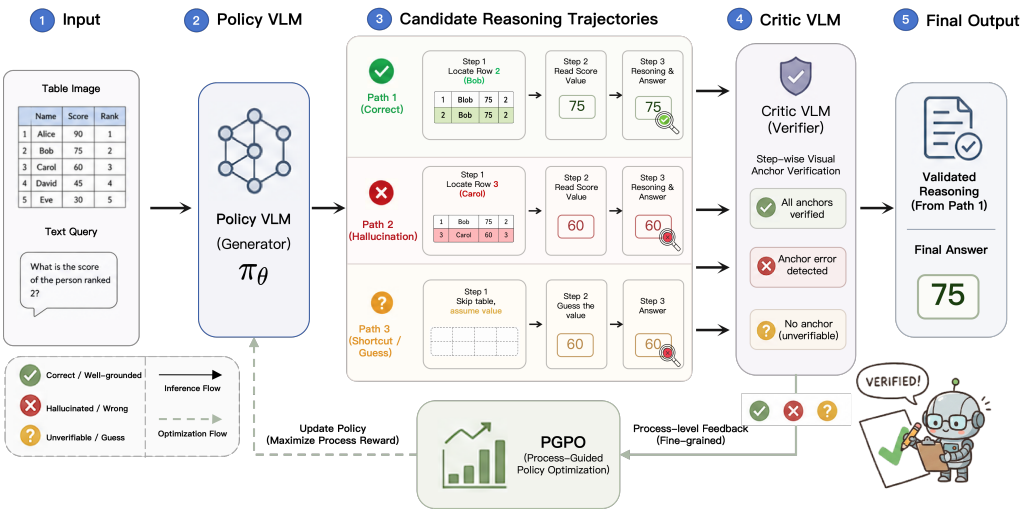

Figure 2: V-tableR1 training pipeline. Given a table image and query, the policy VLM generates anchor-laden visual reasoning trajectories. The critic VLM scores each visual anchor to separate faithful inference, hallucinated grounding, and shortcut guessing, then converts the scored trajectory into process feedback for PGPO. Only reasoning paths whose anchors receive high critic scores are treated as faithful evidence for the final answer.

full role ontology and the rationale behind each role design in Appendix E.

Behavioral decomposition. The anchor triple $(\mathcal{C}_k, \mathcal{V}_k, \rho_k)$ separates three behaviors that outcome-only RLVR conflates (Fig. 2, right): **(P1)** *faithful*: anchors are valid, values match the image, and the answer is correct; **(P2)** *hallucinated*: an anchor cites unsupported coordinates or values, causing the answer to follow corrupted evidence; **(P3)** *shortcut*: anchors are missing or invalid, yet the answer is correct through priors rather than the declared chain. Under terminal supervision, P1 and P3 receive identical reward despite opposite evidential status.

3.3 Critic VLM as a Step-Level verifier

Construction of process-score supervision. A key challenge in training a reliable process verifier is that directly scoring erroneous model-generated reasoning trajectories is often noisy: such trajectories may contain intertwined errors in perception, anchoring, arithmetic, and language generation, making it difficult to assign a faithful process score. We therefore construct supervision through *controlled perturbation* of verified anchored trajectories. Starting from high-quality reasoning chains whose anchors have been checked against the source table, we inject localized errors into the anchored process while keeping the overall trajectory format unchanged. This allows us to obtain paired data between a reasoning trajectory and a calibrated process-quality score, with explicit knowledge of which step has been corrupted and how severe the corruption is.

Specifically, for each verified trajectory $y = (s_1, a_1, \ldots, s_K, a_K, \hat{y})$, we initialize $r^\star_{\text{proc}} = 1$ and perturb a subset of steps with three atomic operations: (i) *coordinate corruption*, shifting $\mathcal{C}_k$ to an incorrect row, column, or region; (ii) *value corruption*, replacing $\mathcal{V}_k$ with values inconsistent with the referenced evidence; and (iii) *reasoning deletion*, removing a necessary step s_k and its anchor a_k. To avoid treating all errors equally, each corrupted step receives a severity-weighted penalty:

$$\Delta_k = w(\rho_k)\left(\lambda_{\text{coord}} e_k^{\text{coord}} + \lambda_{\text{val}} e_k^{\text{val}} + \lambda_{\text{del}} e_k^{\text{del}}\right), \quad (4)$$

where $e_k^{\text{coord}}, e_k^{\text{val}}, e_k^{\text{del}} \in \{0, 1\}$ indicate the applied perturbations, $\lambda_\cdot$ controls error-type severity, and $w(\rho_k)$ weights the importance of the structural role. The process score is then

$$r^\star_{\text{proc}} = \max\left(0,\ 1 - \sum_{k \in \mathcal{E}} \Delta_k\right), \quad (5)$$

where $\mathcal{E}$ is the set of perturbed steps. This assigns larger penalties to errors that break the evidence chain, alter operands, or remove necessary reasoning steps, yielding graded supervision beyond error counting. We provide a detailed validation of the critic's reliability, including human agreement and hard-negative analysis, in Appendix B.

Critic training. The critic C_ϕ is initialized from Qwen3-VL-32B and trained on pairs $(x, q, y, r^\star_{\text{proc}})$

with a regression objective

$$\mathcal{L}_\phi = \mathbb{E}_{(x,q,y)} \left[\left(C_\phi(x, q, y) - r^\star_{\text{proc}} \right)^2 \right], \quad (6)$$

where the output head produces a scalar in $[0, 1]$ via a sigmoid activation. Crucially, the critic conditions on the full input (x, q, y), so its judgement of a_k depends jointly on the image, the question, and the surrounding logical context—making it sensitive to whether an anchor is *used* faithfully, not merely whether it is *stated* consistently.

3.4 Process-Guided Policy Optimization

PGPO should be understood as a process-aware adaptation of GRPO rather than a new optimizer family. Although the critic outputs a trajectory-level scalar, the score is conditioned on step-level visual anchors and their consistency with the source table; thus the reward is process-aware without requiring token-level PRM supervision.

Interval-gated composite reward. Let $r_{\text{fmt}} \in \{0, 1\}$ indicate whether the trajectory follows the anchored format, $r_{\text{acc}} \in \{0, 1\}$ indicate final-answer correctness, and $r_{\text{proc}} = C_\phi(x, q, y) \in [0, 1]$ denote the critic's process score. We always reward format compliance, but gate answer credit by process quality only after correctness is established, separating valid anchoring, answer correctness, and visually grounded reasoning. We define

$$R(y) = r_{\text{fmt}} + \begin{cases} 0, & r_{\text{acc}} = 0, \\ 1 + \alpha, & r_{\text{acc}} = 1,\ r_{\text{proc}} \geq \tau_h, \\ \beta, & r_{\text{acc}} = 1,\ r_{\text{proc}} \leq \tau_\ell, \\ 1, & r_{\text{acc}} = 1,\ \tau_\ell < r_{\text{proc}} < \tau_h. \end{cases} \quad (7)$$

Here τ_h and τ_ℓ are high- and low-confidence process thresholds; by default, $\tau_h = 0.8$, $\tau_\ell = 0.5$, $\alpha = 0.5$, and $\beta = 0.2$.

This interval gate rewards correct answers supported by reliable anchors, preserves standard credit for acceptable reasoning, and discounts shortcut correctness. Compared with directly injecting the continuous critic score, thresholding reduces sensitivity to calibration noise and yields more robust policy optimization. Appendix F ablates thresholds and coefficients.

From DAPO to PGPO. Directly substituting $R(y)$ into GRPO is unstable in our setting: critic-gated rewards increase group-level variance, while long anchored trajectories make rare but important anchor tokens sensitive to clipping. PGPO addresses this by making the policy update itself process-aware. For a group of G rollouts $\{y_i\}_{i=1}^G$ from $\pi_{\theta_{\text{old}}}$, we first compute the group-normalized advantage

$$\hat{A}_i = \frac{R(y_i) - \text{mean}_j R(y_j)}{\text{std}_j R(y_j) + \varepsilon}. \quad (8)$$

The key difference from directly applying DAPO is that PGPO uses the critic score not only to shape $R(y_i)$, but also to modulate the clipping range of each trajectory-level update.

Score-adaptive clipping for anchor-laden rollouts. Anchored trajectories are token-imbalanced: anchor tokens are sparse but disproportionately informative, while explanatory tokens dominate length. A fixed upper clipping bound treats faithful and poorly grounded trajectories identically. We therefore replace the fixed DAPO upper bound with a score-adaptive bound:

$$\epsilon_h^{(i)} = \epsilon_h^{\text{base}} + \gamma\, \mathbb{I}[r_{\text{acc}}^{(i)} = 1]\, \max(0, r_{\text{proc}}^{(i)} - \tau_h). \quad (9)$$

Only trajectories that are both answer-correct and highly supported by the critic receive a relaxed upper clip; incorrect or weakly grounded trajectories use the base bound ϵ_h^{base}. The full SAC-PGPO objective is given in Appendix C.

Length-aware active set. After score-adaptive clipping fixes the update magnitude of each trajectory, PGPO further filters rollouts by length to retain high-value training signals. In anchored visual reasoning, short rollouts often reveal decisive anchoring decisions, while moderately long rollouts capture evidence aggregation, comparison, or self-correction. Routine middle-length samples and overlong tails, in contrast, tend to provide weaker signal or redundant exploration. Motivated by the link between rollout length and policy improvement (Chen et al., 2025), PGPO constructs an active set from each group by retaining the shortest 25% and the 60–85% length band:

$$\mathcal{G}_{\text{act}} = \{i \mid |y_i| \leq Q_{25} \ \text{ or } \ Q_{60} \leq |y_i| \leq Q_{85}\}, \quad (10)$$

where $Q_p = Q_p(\{|y_j|\}_{j=1}^G)$ is the p-th percentile of rollout lengths within the group.

Length is not used as a proxy for correctness, but as a stratification signal that emphasizes trajectories likely to expose either compact anchoring behavior or non-trivial multi-step reasoning. Updating only on $\mathcal{G}_{\text{act}}$ reduces gradient noise from low-information rollouts while preserving regimes

most useful for learning faithful anchored reasoning. Appendix G ablates alternative partitions and shows that the 0–25% and 60–85% split gives the best trade-off among stability, anchor fidelity, and answer accuracy.

4 Experiments

To evaluate the effectiveness of V-tableR1, we construct a comprehensive training and evaluation pipeline across multiple tabular reasoning benchmarks. Our evaluation encompasses two primary tasks: Table Fact Verification (TFV) and Table Question Answering (TQA).

4.1 Datasets

We evaluate V-tableR1 on seven standard tabular reasoning datasets, covering two task families: Table Fact Verification (TFV) and Table Question Answering (TQA). These datasets span diverse table layouts, hierarchical headers, numerical reasoning, multi-hop lookup, and semantic entailment. Table 1 summarizes the training/evaluation splits and test-set image statistics.

Table Fact Verification (TFV). We use TabFact (Chen et al., 2019) and InfoTabs (Gupta et al., 2020). TabFact requires verifying natural-language statements against Wikipedia tables, while InfoTabs extends table-based inference to semi-structured infoboxes with entailment, contradiction, and neutral labels.

Table Question Answering (TQA). We evaluate on FinQA (Chen et al., 2021), HiTab (Cheng et al., 2022), TAT-QA (Zhu et al., 2021), TabMWP (Lu et al., 2022), and WikiTableQuestions (WTQ) (Pasupat and Liang, 2015). These datasets require models to answer questions over tables involving financial arithmetic, hierarchical header grounding, hybrid table-text evidence, math problems, aggregation, superlatives, and multi-hop lookup.

Notably, InfoTabs and TAT-QA are evaluated without any training data (Table 1), making their results a direct measure of out-of-domain zero-shot transferability under distribution shift.

4.2 Main Results

Table 2 compares V-tableR1 with closed-source APIs, large open-weight VLMs, and table-specialized baselines. Overall, V-tableR1 achieves the strongest average performance among the evaluated open-source models across both Table Fact Verification (TFV) and Table Question Answering (TQA), providing evidence that explicit supervision over auditable reasoning trajectories improves structured visual reasoning.

Table 1: Statistics of the tabular reasoning datasets used in this study. The datasets cover two task types: Table Fact Verification and Table Question Answering. Split sizes are reported as SFT Train / RL Train / Test, and image statistics are computed from the test splits.

Dataset	Splits	Image Stats
Table Fact Verification (TFV)		
TabFact	5.25K / 3.65K / 3.78K	1360×409, 103.2KB
InfoTabs	– / – / 1.20K	1425×392, 60.4KB
Table Question Answering (TQA)		
FinQA	2.57K / 0.58K / 1.13K	1838×308, 82.4KB
HiTab	3.20K / 1.55K / 1.58K	1242×458, 77.6KB
TabMWP	4.41K / 3.79K / 2.69K	269×192, 20.7KB
WTQ	5.74K / 5.89K / 2.34K	1140×520, 106.9KB
TAT-QA	– / – / 0.74K	1506×358, 77.6KB

Note: Splits are reported as SFT Train / RL Train / Test.

Complex TQA and process-guided optimization. TQA benchmarks require cell localization, multi-hop lookup, aggregation, and arithmetic reasoning. V-tableR1 achieves the best open-source average TQA performance and sets new open-source results on FinQA, HiTab, and TabMWP with substantially fewer parameters. For example, V-tableR1 (4B) reaches 28.98% on FinQA and 83.38% on TabMWP, outperforming InternVL-2.5-14B (24.27% and 74.34%) and remaining competitive with QVQ-72B (23.12% and 82.20%). The gains from SFT checkpoints to RL-tuned V-tableR1 models further show the importance of process-guided optimization: V-tableR1 (2B) improves over Qwen3-VL-2B (SFT) by +9.19% on FinQA and +9.00% on HiTab, while V-tableR1 (4B) improves over its SFT counterpart by +10.10% on FinQA and +10.66% on HiTab. These results suggest that SFT mainly teaches the format of anchored reasoning, whereas PGPO further optimizes whether the generated anchors faithfully support the reasoning process.

Fact verification and parameter efficiency. Although V-tableR1 is designed for complex multi-step table reasoning, it also performs strongly on TFV. V-tableR1 (4B) obtains 87.95% on TabFact and 88.94% on InfoTabs, matching or surpassing several much larger open-source models and approaching strong proprietary systems. The improvement over its SFT baseline on TabFact (83.88% $\rightarrow$ 87.95%) indicates that anchor-level

Table 2: Performance comparison on Table Fact Verification (TFV) and Table Question Answering (TQA) benchmarks. † denotes zero-shot evaluation sets with no SFT/RL training data. Overall Avg. is the macro-average over all seven datasets. Best and second-best open-source results are shown in **bold** and underlined, respectively.

Model	Table Fact Verification (TFV)			Table Question Answering (TQA)						Overall Avg.
	TabFact	InfoTabs†	Avg.	FinQA	HiTab	TAT-QA†	TabMWP	WTQ	Avg.	
Closed-Source VLMs										
GPT-4o	85.16	88.75	86.96	13.42	27.59	30.98	71.88	57.80	40.33	53.65
GPT-4.1	87.32	92.43	89.88	23.44	32.39	47.63	74.06	63.45	48.19	60.10
Gemini-2.5-Flash	88.16	91.43	89.80	40.48	55.43	70.27	88.37	78.18	66.55	73.19
Open-Source VLMs										
InternVL-2.5-8B (Chen et al., 2024a)	72.92	74.67	73.80	19.42	32.26	41.44	72.80	57.80	44.74	53.04
InternVL-2.5-14B (Chen et al., 2024a)	75.98	77.48	76.73	24.27	33.43	46.28	74.34	61.89	48.04	56.24
LLaVA-1.5-7B (Liu et al., 2024)	23.25	32.34	27.80	1.50	3.66	8.29	5.92	12.32	6.34	12.47
LLaVA-1.5-13B (Liu et al., 2024)	28.37	37.34	32.86	1.24	5.81	10.05	6.14	15.63	7.77	14.94
LLaVA-NeXT-7B (Li et al., 2024)	34.18	41.23	37.71	2.03	4.67	12.64	19.16	8.36	9.37	17.47
Qwen2.5-VL-72B (Bai et al., 2025b)	73.45	72.82	73.14	19.34	44.04	52.23	81.95	65.70	52.65	58.50
Qwen3-VL-2B (Bai et al., 2025a)	70.40	74.83	72.62	10.50	29.80	32.07	68.60	46.04	37.40	47.46
Qwen3-VL-4B (Bai et al., 2025a)	73.28	77.33	75.31	14.74	27.75	35.19	71.92	54.72	40.86	50.70
Qwen3-VL-8B (Bai et al., 2025a)	79.44	71.33	75.39	17.12	43.56	50.00	77.88	63.04	50.32	57.48
Qwen3-VL-32B (Bai et al., 2025a)	82.43	78.25	80.34	26.32	46.20	**56.23**	80.37	65.49	54.92	62.18
QVQ-72B (Team, 2024)	77.68	74.64	76.16	23.12	45.70	55.48	82.20	**68.20**	54.94	61.00
Table-LLaVA-7B (Zheng et al., 2024)	59.85	65.26	62.56	5.29	10.09	12.82	57.78	18.43	20.88	32.79
Table-LLaVA-13B (Zheng et al., 2024)	65.00	66.91	65.96	7.71	10.85	15.67	59.77	20.41	22.88	35.19
Ours										
Qwen3-VL-2B (SFT)	79.28	85.17	82.23	15.30	36.45	43.64	74.56	47.48	43.49	54.55
Qwen3-VL-4B (SFT)	83.88	87.33	85.61	18.88	36.58	46.63	76.60	54.32	46.60	57.75
V-tableR1 (2B)	84.69	87.54	86.12	24.49	45.45	49.83	78.94	56.49	51.04	61.06
V-tableR1 (4B)	**87.95**	**88.94**	**88.45**	**28.98**	**47.24**	54.23	**83.38**	63.37	**55.44**	**64.87**

Table 3: Ablation study on the optimization components using the 2B model. We report average accuracy across TFV and TQA benchmarks.

Method	TFV-Avg.	TQA-Avg.
GRPO	81.34	46.43
DAPO	82.48	49.04
PGPO w/o PS	83.12	49.34
PGPO	**86.12**	**51.04**

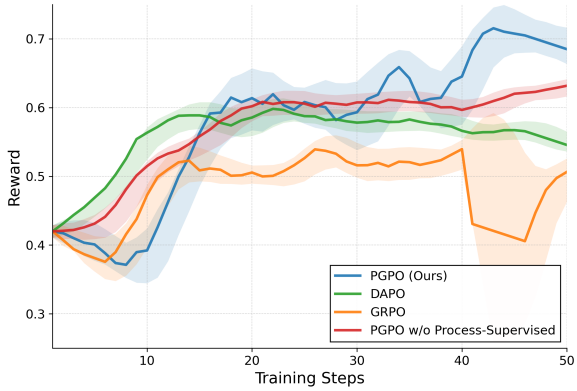

Figure 3: Training reward curves for different optimization strategies. PGPO achieves more stable convergence and a higher final reward than GRPO, DAPO, and PGPO without process supervision.

supervision also reduces unsupported table claims. More broadly, V-tableR1 delivers strong performance with only 2B–4B parameters: the 4B model outperforms many 72B-scale open-weight VLMs across most TFV and TQA benchmarks, while table-specialized SFT baselines such as Table-LLaVA remain weak on numerically intensive TQA tasks. These comparisons suggest that the key bottleneck is not merely table exposure or model scale, but the absence of step-level supervision over visually grounded reasoning.

Out-of-domain generalization. InfoTabs and TAT-QA are evaluated zero-shot, with no in-domain SFT or RL data used by V-tableR1. V-tableR1 (4B) leads the strongest evaluated open-source baseline by 10.69 points on InfoTabs, suggesting transfer beyond training table formats. On TAT-QA, it remains close to the strongest open-source result (54.23% vs. 56.23%), indicating that hybrid table–text evidence is still harder for the current table-centric anchor design.

4.3 Qualitative Analysis

Figure 4 illustrates how process supervision changes reasoning behavior on a multi-hop table grounding query. The baseline Qwen3-VL-2B identifies the question intent but grounds the answer to the wrong region, extracting the receiving-yard total instead of the rushing-yard total, a visual hallucination failure in which plausible reasoning is

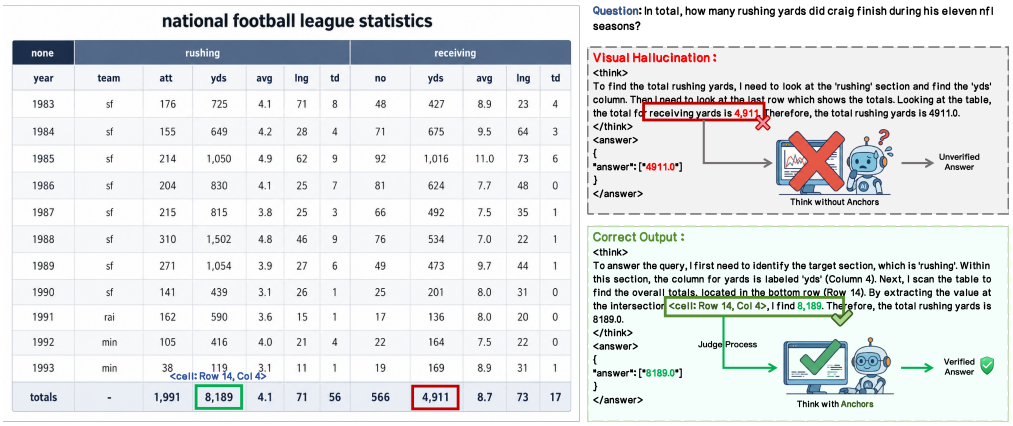

Figure 4: Qualitative comparison before and after PGPO. The baseline extracts the value from an incorrect table region, while V-tableR1 grounds the answer through an explicit visual anchor.

not supported by the cited evidence. After PGPO training, V-tableR1 localizes the relevant rushing section and anchors the required value to the correct row and column before answering. The anchor, `<cell: Row 14, Col 4>`, binds the extracted value to the source table, making the trajectory auditable and replacing shortcut inference with faithful visual evidence tracing.

4.4 Ablation Studies

We ablate the key components of V-tableR1 using Qwen3-VL-2B as the base model. Table 3 reports the average accuracy on TFV and TQA benchmarks. Starting from standard GRPO, DAPO improves TQA performance from 46.43% to 49.04%, suggesting that decoupled clipping stabilizes RL training for long visual reasoning trajectories. Adding length-aware sampling further improves performance to 49.34%, indicating that selecting informative rollouts provides modest but consistent gains. The largest improvement comes from introducing critic-gated process supervision: full PGPO reaches 86.12% on TFV and 51.04% on TQA, outperforming GRPO by +4.78% and +4.61%, respectively. These results show that optimization stability helps, but the dominant source of improvement is the process-level supervision that penalizes ungrounded or shortcut reasoning.

Training stability. Figure 3 compares reward curves across optimization strategies. GRPO shows unstable dynamics and a late-training collapse, while DAPO improves stability but reaches a lower plateau. PGPO without process supervision remains limited by outcome-only feedback. In contrast, full PGPO achieves the highest and most stable reward trajectory, suggesting that critic-gated process rewards suppress shortcut exploitation and guide the policy toward faithful anchored reasoning. To further substantiate process supervision, we use GPT-5 as an LLM-as-a-judge evaluator for a complementary assessment of generated reasoning trajectories; this analysis is reported alongside human and anchor-level audits in Appendix J.

5 Conclusion

The failure of outcome supervision on table reasoning is not only a reward-design problem but also a representational one: when intermediate visual evidence stays outside the symbol system used for rewards, reward shaping cannot reliably separate faithful derivation from confident hallucination. Anchored Reasoning addresses this by requiring each step to expose, in table-intrinsic coordinates, the structural region from which it draws evidence. We instantiate this as V-tableR1, where a policy VLM emits anchor-laden chains-of-thought scored by a learned critic VLM and is trained with PGPO, a GRPO-based formulation coupling critic-gated rewards, decoupled clipping, and length-aware sampling. Across seven tabular benchmarks, V-tableR1 (4B) achieves the strongest average accuracy among the evaluated open-source models, surpasses substantially larger open-source inference-time policies in aggregate, and improves over SFT by up to 10.66%. These results suggest that process supervision can complement parameter scaling: within a structurally controlled domain such as table reasoning, auditable intermediate structure improves the efficiency–scale trade-off for smaller inference-time policies.

Limitations

Our framework has several important limitations. First, V-tableR1 deliberately scopes Anchored Reasoning to the deterministic grid geometry of tables, where coordinates, headers, and operand sets admit unambiguous symbolic verification. Extending this paradigm to documents whose structural provenance is only approximately symbolic—charts with continuous axes, scientific diagrams with relational topology, or natural scenes—requires redefining what constitutes a verifiable anchor and remains beyond the scope of this work. Second, PGPO depends on a separately trained 32B critic VLM, which introduces non-trivial compute overhead during training and ties the upper bound of process supervision to the critic's own perceptual fidelity. While the critic is fixed and incurs no inference-time cost for the deployed policy, lightweight alternatives that preserve the dense step-level signal would broaden the applicability of our approach. Third, our data-construction and analysis pipeline relies on large or proprietary models for trajectory generation, verification, perturbation, and judging, which limits reproducibility before release and makes the reported parameter efficiency primarily an inference-time claim for the deployed policy.

Although we describe anchors as verifiable, PGPO currently operationalizes this verification with a learned critic rather than a deterministic symbolic checker; our claim is therefore that Anchored Reasoning narrows the observability gap by making intermediate visual evidence inspectable, not that it provides formal verification guarantees.

Ethical Considerations

This work uses publicly available tabular reasoning benchmarks (TabFact, InfoTabs, FinQA, HiTab, TAT-QA, TabMWP, WTQ) under their original licenses, and introduces no new data involving human subjects or personally identifiable information. The proposed Anchored Reasoning paradigm is, by design, oriented toward improving the transparency and auditability of multimodal reasoning: by requiring each step to emit a verifiable visual anchor, V-tableR1 makes intermediate evidence inspectable rather than opaque, which we view as a positive contribution to the trustworthiness of VLM-based systems in high-stakes domains such as financial and scientific document analysis. We nonetheless caution that anchor verification reduces but does not eliminate residual hallucinations, and outputs should not be treated as authoritative without human review in deployment-critical settings.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, and 1 others. 2025a. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025b. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Abhinaba Basu and Pavan Chakraborty. 2026. Measuring and curing reasoning rigidity: from decorative chain-of-thought to genuine faithfulness. *arXiv preprint arXiv:2603.22816*.

Weizhe Chen, Sven Koenig, and Bistra Dilkina. 2025. Lspo: Length-aware dynamic sampling for policy optimization in llm reasoning. *arXiv preprint arXiv:2510.01459*.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2019. Tabfact: A large-scale dataset for table-based fact verification. *arXiv preprint arXiv:1909.02164*.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, and 1 others. 2024a. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

Zhenfang Chen, Qinhong Zhou, Yikang Shen, Yining Hong, Zhiqing Sun, Dan Gutfreund, and Chuang Gan. 2024b. Visual chain-of-thought prompting for knowledge-based visual reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 1254–1262.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan R Routledge, and 1 others. 2021. Finqa: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711.

Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. 2022. Hitab: A hierarchical table dataset for question answering and natural language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1094–1110.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, and 1 others. 2025. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*.

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

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. 2020. Infotabs: Inference on tables as semi-structured data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2309–2324.

Deyi Ji, Lanyun Zhu, Siqi Gao, Peng Xu, Hongtao Lu, Jieping Ye, and Feng Zhao. 2024. Tree-of-table: Unleashing the power of llms for enhanced large-scale table understanding. *arXiv preprint arXiv:2411.08516*.

Fangkai Jiao, Chengwei Qin, Zhengyuan Liu, Nancy Chen, and Shafiq Joty. 2024. Learning planning-based reasoning by trajectories collection and process reward synthesizing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 334–350.

Nengzheng Jin, Joanna Siebert, Dongfang Li, and Qingcai Chen. 2022. A survey on table question answering: recent advances. In *China Conference on Knowledge Graph and Semantic Computing*, pages 174–186. Springer.

Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. 2024. What matters when building vision-language models? *Advances in Neural Information Processing Systems*, 37:87874–87907.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. 2024. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2022. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *arXiv preprint arXiv:2209.14610*.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 305–329.

Youssef Mroueh. 2025. Reinforcement learning with verifiable rewards: Grpo's effective loss, dynamics, and success amplification. *arXiv preprint arXiv:2503.06639*.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480.

Vipula Rawte, Aryan Mishra, Amit Sheth, and Amitava Das. 2025. Defining and quantifying visual hallucinations in vision-language models. In *Proceedings of the 5th Workshop on Trustworthy NLP (TrustNLP 2025)*, pages 501–510.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Yi Su, Dian Yu, Linfeng Song, Juntao Li, Haitao Mi, Zhaopeng Tu, Min Zhang, and Dong Yu. 2025. Crossing the reward bridge: Expanding rl with verifiable rewards across diverse domains. *arXiv preprint arXiv:2503.23829*.

Ningyuan Sun, Xuefeng Yang, and Yunfeng Liu. 2020. Tableqa: a large-scale chinese text-to-sql dataset for table-aware sql generation. *arXiv preprint arXiv:2006.06434*.

Qwen Team. 2024. Qvq: To see the world with wisdom.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965.

Milan Vojnovic and Se-Young Yun. 2025. What is the alignment objective of grpo? *arXiv preprint arXiv:2502.18548*.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024a. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439.

Weiyun Wang, Zhangwei Gao, Lianjie Chen, Zhe Chen, Jinguo Zhu, Xiangyu Zhao, Yangzhou Liu, Yue Cao, Shenglong Ye, Xizhou Zhu, and 1 others. 2025. Visualprm: An effective process reward model for multimodal reasoning. *arXiv preprint arXiv:2503.10291*.

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, and 1 others. 2024b. Chain-of-table: Evolving tables in the reasoning chain for table understanding. *arXiv preprint arXiv:2401.04398*.

Zhenhe Wu, Jian Yang, Jiaheng Liu, Xianjie Wu, Changzai Pan, Jie Zhang, Yu Zhao, Shuangyong Song, Yongxiang Li, and Zhoujun Li. 2025. Table-r1: Region-based reinforcement learning for table understanding. *arXiv preprint arXiv:2505.12415*.

Jingfeng Yang, Aditya Gupta, Shyam Upadhyay, Luheng He, Rahul Goel, and Shachi Paul. 2022. Tableformer: Robust transformer modeling for table-text encoding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 528–537.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, and 1 others. 2025. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. The lessons of developing process reward models in mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10495–10516.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, and 1 others. 2025. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1702–1713.

Weichao Zhao, Hao Feng, Qi Liu, Jingqun Tang, Binghong Wu, Lei Liao, Shu Wei, Yongjie Ye, Hao Liu, Wengang Zhou, and 1 others. 2024. Tabpedia: Towards comprehensive visual table understanding with concept synergy. *Advances in Neural Information Processing Systems*, 37:7185–7212.

Mingyu Zheng, Xinwei Feng, Qingyi Si, Qiaoqiao She, Zheng Lin, Wenbin Jiang, and Weiping Wang. 2024. Multimodal table understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9102–9124.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. In *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)*, pages 3277–3287.

A Experimental Settings

A.1 Training and Implementation Details

We provide the default training and implementation details for V-tableR1 in Table 4. Unless otherwise specified, all SFT and RL experiments follow the same preprocessing pipeline, anchored trajectory format, and evaluation protocol described in the main paper. All experiments use a fixed random seed, seed = 42, for data splitting, sampling, and model initialization.

Table 4: Default training and implementation hyperparameters for V-tableR1.

Hyperparameter	Value
Policy backbones	Qwen3-VL-2B / Qwen3-VL-4B
Critic backbone	Qwen3-VL-32B
Rollout group size G	16
Max response length	2048
Sampling temperature	1.0
Top-p	0.9
SFT learning rate	1×10^{-5}
RL learning rate	5×10^{-7}
Optimizer	AdamW
Batch size	128
ϵ_ℓ	0.2
ϵ_h	0.28
τ_ℓ	0.5
τ_h	0.8
α	0.5
β	0.2
Active set	0–25% and 60–85%
SFT hardware	32× H20 GPUs
Post-training hardware	64× H20 GPUs
4B / critic training hardware	32× H20 GPUs

For SFT, we train the policy model on verified anchored trajectories so that it learns the required output format and basic grounding behavior. For RL post-training, we sample $G = 16$ rollouts per query and optimize the policy with PGPO using the interval-gated composite reward. The 2B post-training experiments are conducted on $64\times$ H20 GPUs. The SFT stage, as well as 2/4B model training and critic-model training, is conducted on $32\times$ H20 GPUs. The critic model is trained separately before RL post-training and remains fixed during PGPO optimization.

A.2 Critic Training Details

We provide additional details on the construction and training of the critic VLM. The critic is designed to estimate the process quality of an anchored reasoning trajectory, rather than simply judge whether the final answer is correct.

Empirical coverage of perturbation types. To verify that the three atomic perturbations (coordinate corruption, value corruption, and reasoning deletion) faithfully cover the failure modes that arise in practice, rather than reflecting only our a priori expectations, we conduct an error audit on naturally occurring policy rollouts. Specifically, we sample 500 incorrect rollouts uniformly from available training or development splits, excluding the final test sets, and annotate each erroneous step under a unified taxonomy: (i) coordinate error—the referenced cell, row, or region is incorrect; (ii) value error—the extracted value is inconsistent with the cited region; (iii) reasoning omission—a necessary intermediate step or anchor is missing; (iv) arithmetic error—the numerical computation is wrong given the stated operands, regardless of whether the operands are fully correct; and (v) other—fluency artifacts, format violations, and uncategorized cases.

Table 5 reports the distribution of annotated error types. Coordinate, value, and omission errors jointly account for 89.4% of all observed failures, with coordinate and value errors dominating (41.6% and 34.2%, respectively) and reasoning omission accounting for 13.6%. The remaining 10.6% are split between arithmetic errors (6.8%) and miscellaneous artifacts (3.8%); many arithmetic errors co-occur with upstream anchoring failures, while a smaller subset remains as pure computation errors over otherwise correctly identified operands. The empirical distribution also justifies our sampling probabilities $(0.4, 0.4, 0.2)$ for the three perturbation types: they approximate the relative frequency of the corresponding natural errors, ensuring the critic is trained on a synthetic distribution aligned with the failure modes it will encounter during PGPO optimization.

Verified anchored trajectories. We construct verified anchored trajectories using GPT-4.1 as the trajectory-generation teacher and Qwen3-VL-32B as an auxiliary multimodal verifier. Given a table image and its question, the teacher generates an anchored reasoning trajectory with explicit cell coordinates, extracted values, structural roles, and the final answer. We then verify the generated anchors with Qwen3-VL-32B against the source table image and its parsed table structure. Cell coordinates are defined in the table's relative grid space rather than absolute pixel space, so that anchors remain stable across image resolutions and rendering

Table 5: Distribution of error types in 500 incorrect rollouts from the early-stage SFT policy. Coordinate, value, and omission errors jointly cover 89.4% of observed failures, supporting the design of our controlled perturbation taxonomy.

Error Type	Frequency (%)	Covered
Coordinate error	41.6	✓
Value error	34.2	✓
Reasoning omission	13.6	✓
Arithmetic error	6.8	Indirect[†]
Other	3.8	—
Total covered	**89.4**	

[†]Arithmetic errors often co-occur with upstream anchoring failures, although a subset remains as pure computation errors even when anchors are correct.

styles. For merged cells and hierarchical headers, we record the corresponding cell span, e.g., the row index together with the start and end column indices, and use this span-level representation during verification.

A trajectory is retained only if both the final answer and the intermediate anchored reasoning are verified as correct. Specifically, we first filter out samples whose final answer does not match the ground-truth label under the task evaluation rule. For the remaining samples, Qwen3-VL-32B and Gemini-2.5-Flash independently check whether the chain-of-thought contains unsupported cell references, incorrect value extraction, wrong structural-role assignments, or reasoning errors. We keep only trajectories that pass both verifiers, yielding a high-precision set of positive trajectories for SFT and critic supervision.

Controlled perturbation. Directly scoring naturally erroneous model trajectories is unreliable, since such trajectories often mix localization errors, value extraction errors, arithmetic mistakes, and language-generation artifacts. We therefore construct process-score supervision by applying controlled perturbations to verified trajectories. Specifically, we use Qwen3-32B as an auxiliary perturbation model to rewrite verified trajectories under predefined corruption rules, ensuring that the injected errors are localized and controllable while the resulting reasoning text remains fluent. We perturb a sampled subset of reasoning steps with three atomic operations: coordinate corruption, value corruption, and reasoning deletion. Unless otherwise specified, we sample perturbation types with probabilities 0.4, 0.4, and 0.2, respectively. These operations correspond to the dominant failure modes observed in preliminary rollouts: wrong localization, wrong value reading, and missing reasoning evidence.

Severity-weighted process labels. To avoid treating all errors equally, we reuse the severity-weighted penalty Δ_k defined in Eq. 4 of Sec. 3.3, instantiated here with concrete coefficients. The error-type weights are set to $\lambda_{\text{coord}} = 0.2$, $\lambda_{\text{val}} = 0.3$, $\lambda_{\text{del}} = 0.4$, reflecting the increasing severity of mislocalization, incorrect value extraction, and missing reasoning steps. The role weight $w(\rho_k)$ is set to 1.0 for auxiliary references, 1.2 for headers or conditions, and 1.5 for operands or aggregation sets. Under this parameterization, the final process label $r^{\star}_{\text{proc}}$ follows directly from Eq. 5, yielding graded supervision that reflects both the type of error and its importance in the reasoning chain.

Critic training and validation. The critic C_ϕ is initialized from a strong VLM backbone and trained to regress the process score $r^{\star}_{\text{proc}}$ from the input tuple (x, q, y). We use a mean-squared error objective and reserve a held-out validation split to monitor calibration. During validation, we evaluate whether the critic assigns higher scores to verified trajectories than to corrupted ones, and whether its predicted scores are monotonic with perturbation severity. This validation ensures that the critic provides a meaningful process-quality signal before it is used for PGPO training.

Perturbation prompt. We use Qwen3-32B as an auxiliary perturbation model to rewrite verified trajectories under predefined corruption rules. The prompt used for controlled perturbation is shown in Figure 5.

Data hygiene and teacher–evaluator separation. Because our trajectory-generation and verification pipeline uses GPT-4.1, Gemini-2.5-Flash, and Qwen3-VL-32B—some of which also appear in Table 2 as evaluation baselines—we explicitly separate the roles these models play to avoid teacher–evaluator contamination. First, GPT-4.1 and Gemini-2.5-Flash are used *only* on training-split tables and questions for trajectory generation and verification; no test instance from any of the seven benchmarks is exposed to these models during data construction. Second, the resulting verified trajectories supervise SFT and the critic, never the test set. Third, when GPT-4.1 and Gemini-2.5-Flash appear in Table 2, they are evaluated zero-shot on test splits under the same protocol applied to all other open- and closed-source baselines, with

Prompt for Controlled Trajectory Perturbation

System Instruction:
You are an expert assistant for constructing controlled negative samples for table reasoning. Your task is to perturb a verified anchored reasoning trajectory according to a specified corruption type, while keeping the overall reasoning format fluent and natural.

Input:
You are given:

- a natural-language question;
- a verified anchored reasoning trajectory;
- the target perturbation type;
- the step index to perturb.

Allowed Perturbation Types:

- **Coordinate Corruption:** replace the referenced coordinate or region with an incorrect row, column, or neighboring region.
- **Value Corruption:** replace the claimed value with a value inconsistent with the referenced evidence.
- **Reasoning Deletion:** remove a necessary intermediate reasoning step and its associated anchor.

Constraints:

- Do not change the question.
- Do not change unperturbed reasoning steps.
- Do not introduce additional errors beyond the specified perturbation.
- Preserve the original output style and anchored reasoning format.
- Ensure the corrupted trajectory remains fluent and plausible.

Output Format:
Return the corrupted trajectory and a short perturbation record: {
  "corrupted_trajectory": "...",
  "perturbed_step": step index,
  "perturbation_type":
"coordinate"/"value"/"deletion",
  "changed_content": "brief description"
}

Figure 5: Prompt used by Qwen3-32B for controlled perturbation of verified anchored reasoning trajectories. The auxiliary model rewrites trajectories under predefined corruption rules while preserving fluency and format consistency.

no access to V-tableR1's training artifacts. Under this separation, the closed-source models serve both as upstream data-construction tools on the training side and as independent reference points on the evaluation side, but never both for the same instance.

B Critic Validation

The reliability of the critic $\mathcal{C}_\phi$ is the load-bearing assumption of our framework: every claim about process supervision collapses if the critic itself cannot distinguish faithful reasoning from its corrupted counterpart. We therefore audit $\mathcal{C}_\phi$ along five complementary dimensions before deploying it as the reward signal for PGPO. Our central finding is that the critic is well-calibrated on synthetic supervision, transfers cleanly to naturally occurring failure modes, and—critically—its score tracks anchor-level visual grounding rather than surface fluency. Throughout this section we compare two configurations, Critic-4B and Critic-32B, to establish both the absolute reliability of our deployed critic and the sensitivity of process supervision to verifier capacity.

Protocol summary. All evaluations are conducted on the held-out validation split, disjoint from both the policy training data and the critic training data. Synthetic perturbations follow the taxonomy of Section 3.3; natural failure modes are mined from the early-stage PGPO policy. Unless otherwise stated, the decision thresholds $\tau_\ell = 0.5$ and $\tau_h = 0.8$ inherit directly from the main experiments, so the validation results below are not post-hoc tuned.

Alignment with Human Judgement. We first establish that the critic agrees with humans on what constitutes a faithful trajectory. Two annotators independently score 200 trajectories (100 verified, 100 perturbed) on a continuous $[0, 1]$ scale using the rubric that produced the critic's training labels. Inter-annotator agreement reaches Cohen's $\kappa = 0.81$, confirming the rubric is operationally well-defined. We then measure mean absolute error (MAE) and Pearson correlation against the averaged human score (Table 6). Critic-32B attains $r = 0.912$ with MAE$= 0.073$, indicating that its predictions are not merely correlated with human judgement but quantitatively calibrated to the same $[0, 1]$ scale. Critic-4B is consistent but less precise ($r = 0.821$); a per-error-type breakdown reveals that its residual error concentrates on coordinate-shift cases, where fine-grained spatial verification

dominates.

Table 6: Alignment with human judgement on 200 held-out trajectories. Human scores are averaged across two annotators ($\kappa = 0.81$).

	MAE ↓	Pearson r ↑
Critic-4B	0.114	0.821
Critic-32B	**0.073**	**0.912**

Discriminative Reliability. Process rewards operate by separating faithful trajectories from corrupted ones; the relevant question is therefore not absolute score accuracy but *ranking* accuracy. We construct 1,200 verified–corrupted pairs from the validation split and report two ranking-based metrics: ROC–AUC and pairwise accuracy, defined as the fraction of pairs in which the critic assigns a strictly higher score to the verified trajectory. As shown in Table 7, Critic-32B reaches AUC $= 0.943$ and 91.7% pairwise accuracy, leaving an error rate (8.3%) that falls well within the noise tolerance afforded by our interval-gated reward (Eq. 7): an occasional mis-ranked pair does not flip the reward bucket as long as it does not cross τ_ℓ or τ_h. The ΔAUC $= 0.062$ gap between the two critics quantifies what is lost when capacity drops by $8\times$, complementing the policy-level comparison in Appendix H.

Table 7: Discriminative reliability on 1,200 verified–corrupted trajectory pairs.

	AUC ↑	Pairwise Acc. ↑
Critic-4B	0.881	84.3%
Critic-32B	**0.943**	**91.7%**

Per-Error-Type Sensitivity. A reliable learned critic should be sensitive to all three failure modes it was designed to penalize. We construct 400 single-perturbation trajectories (with only one error type per trajectory, isolating each signal) and classify a trajectory as erroneous when its score falls below $\tau_\ell = 0.5$. Table 8 reports per-type recognition accuracy. Reasoning deletion is the easiest case for both critics—a missing anchor produces a sharp logical discontinuity the critic detects from context alone. Coordinate corruption is the hardest case, exactly mirroring its dominance among naturally occurring errors (Table 5, 41.6%): subtle row/column shifts demand precise visual cross-referencing rather than logical inference. That the critic's weakest verification dimension coincides with the policy's most frequent failure mode is the strongest single argument for scaling the critic—each percentage point of coordinate-detection accuracy has outsized downstream impact.

Table 8: Per-error-type recognition accuracy on 400 single-perturbation trajectories. Decision threshold: $\tau_\ell = 0.5$.

Error Type	Critic-4B	Critic-32B
Coordinate corruption	81.4%	90.2%
Value corruption	84.7%	92.6%
Reasoning deletion	88.3%	94.1%
Macro average	84.8%	**92.3%**

Grounding Fidelity, Not Surface Fluency. A pervasive failure mode of learned verifiers is reward hacking: the critic rewards trajectories that *sound* rigorous rather than those that *are* grounded. To rule this out, we compute Spearman correlation between the critic score and two anchor-level metrics that are mechanically derivable from the table parse and therefore immune to surface fluency: (i) anchor precision, the fraction of emitted coordinates that exist in the source table; and (ii) value accuracy, the fraction of anchors whose value matches the corresponding cell. Both correlations are strong and significant (Table 9, $p < 0.001$), with Critic-32B reaching $\rho = 0.871$ on anchor precision and $\rho = 0.856$ on value accuracy. The small precision–value gap (0.015) is itself informative: it indicates the critic is *slightly* more confident in localization verification than in value verification, consistent with the intuition that a correctly anchored but numerically near-correct value is the hardest case to detect. This is precisely the failure mode our interval-gated reward is designed to absorb rather than over-trust.

Table 9: Spearman correlation ($p < 0.001$ for all entries) between critic score and mechanical anchor-level metrics on 500 test trajectories.

	Anchor Precision	Value Accuracy
Critic-4B	0.763	0.741
Critic-32B	**0.871**	**0.856**

Policy-Side Anchor Quality Table 9 establishes that the critic score correlates with anchor-level metrics, but does not yet show whether V-tableR1's policy actually produces more faithful anchors than

its SFT initialization. We measure this directly on 500 trajectories sampled uniformly from the TQA test pool, evaluating four metrics that are mechanically derivable from the parsed table and therefore independent of the critic: *anchor precision* (fraction of emitted coordinates existing in the table), *value accuracy* (fraction of anchors whose value matches the cell), *role accuracy* (fraction of anchors whose structural role is correct), and *shortcut rate* (fraction of correct-answer trajectories whose anchors are invalid or absent, i.e. P3 in Appendix D).

Table 10: Policy-side anchor quality on 500 sampled TQA trajectories. PGPO improves all anchor-level dimensions and roughly halves the shortcut rate.

Model	Anchor Prec. ↑	Value Acc. ↑	Role Acc. ↑	Shortcut ↓
Qwen3-VL-2B (SFT)	61.4%	58.7%	72.3%	13.8%
V-tableR1 (2B)	81.6%	78.4%	86.5%	5.4%
Qwen3-VL-4B (SFT)	66.8%	63.2%	75.9%	11.7%
V-tableR1 (4B)	**86.3%**	**82.7%**	**89.4%**	**3.7%**

Across both scales, PGPO improves every anchor-level dimension by 13–21 percentage points while roughly halving the shortcut rate ($13.8\% \rightarrow 5.4\%$ at 2B, $11.7\% \rightarrow 3.7\%$ at 4B). The improvement on anchor precision and value accuracy ($\sim$20 pp) is markedly larger than the improvement on role accuracy ($\sim$ 14 pp), consistent with the intuition that SFT already teaches the role schema as part of the output format, whereas faithful coordinate and value grounding is the dimension that critic-gated process supervision uniquely contributes. Together with the behavioral breakdown in Appendix D, these numbers confirm that V-tableR1's accuracy gains are mechanically traceable to higher anchor fidelity rather than to improved guessing.

Transfer to Naturally Occurring Hard Negatives. The critic is trained on synthetic perturbations whose error distribution is, by construction, clean. The decisive test is whether this clean supervision transfers to the messier, compositional failures produced by an actual policy. We mine 300 hard negatives from the early-stage PGPO policy—trajectories whose final answers are wrong but whose reasoning text is fluent and superficially plausible, the exact pattern the critic must catch during training—and measure two operationally meaningful rates: the *detection rate* (score $\leq \tau_\ell$, the bucket that triggers the β down-weight in Eq. 7) and the *false-negative rate* (score $\geq \tau_h$, the bucket that triggers the $1{+}\alpha$ bonus). Critic-32B detects 87.6% of hard negatives while erroneously rewarding only 4.3% of them (Table 11). The asymmetry between these two numbers matters: PGPO is far more sensitive to false positives (rewarding bad reasoning as if it were faithful) than to false negatives (missing a bad trajectory and assigning it the neutral $r = 1$ bucket), and the critic's error profile favors the safer side.

Table 11: Transfer to 300 hard negatives mined from real early-stage PGPO rollouts. Detection rate: score $\leq \tau_\ell$; false-negative rate: score $\geq \tau_h$.

	Detection Rate ↑	False-Neg. Rate ↓
Critic-4B	76.3%	8.7%
Critic-32B	**87.6%**	**4.3%**

The remaining 12.4% of undetected hard negatives are predominantly arithmetic errors on correctly anchored operands, a failure mode that is not explicitly modeled by the current perturbation taxonomy (cf. Table 5). We therefore treat this residual as an important scope limitation of the current critic and a motivation for online critic refinement in Appendix M.

Synthesis. Across all five evaluations, Critic-32B is well-calibrated ($r = 0.912$ with humans), discriminative (AUC$= 0.943$), broadly sensitive across the three failure modes (macro 92.3%), grounded in mechanical anchor quality rather than surface fluency ($\rho = 0.87$), and—most importantly—it transfers from synthetic to naturally occurring failures with a favorable false-positive profile (4.3%). The consistent gap between Critic-32B and Critic-4B across every evaluation quantifies why critic capacity functions as an upper bound on process supervision, and frames the Critic-4B variant deployed in Appendix H as a meaningful but lossy operating point rather than an equivalent alternative.

C Score-Adaptive Clipping in PGPO

PGPO extends DAPO by making the upper clipping range depend on the process quality of each trajectory. For rollout y_i, the adaptive upper clip is

$$\epsilon_h^{(i)} = \epsilon_h^{\text{base}} + \gamma\, \mathbb{I}[r_{\text{acc}}^{(i)} = 1]\, \max(0, r_{\text{proc}}^{(i)} - \tau_h), \tag{11}$$

where ϵ_h^{base} is the default DAPO upper bound and γ controls the maximum relaxation strength.

The resulting SAC-PGPO objective is

$$\mathcal{J}_{\text{SAC-PGPO}}(\theta) = \frac{1}{|\mathcal{G}_{\text{act}}|} \sum_{i \in \mathcal{G}_{\text{act}}} \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \min\Big(\rho_{i,t}(\theta)\hat{A}_i, \text{clip}\big(\rho_{i,t}(\theta), 1-\epsilon_l, 1+\epsilon_h\big)\hat{A}_i\Big). \tag{12}$$

Here $\hat{A}_i$ is computed from the group-normalized composite reward $R(y_i)$, and $\rho_{i,t}(\theta)$ denotes the token-level policy ratio. This design allows stronger positive updates only for trajectories that are both answer-correct and highly supported by the critic, while keeping the clipping behavior unchanged for incorrect or poorly grounded samples.

D Behavioral Breakdown: Faithful, Hallucinated, and Shortcut Trajectories

Aggregate accuracy alone cannot distinguish whether V-tableR1's gains come from genuinely faithful reasoning or merely from a different distribution of shortcut successes. To probe this directly, we audit model outputs under the three-way behavioral taxonomy introduced in Section 3.2: **(P1)** *faithful correct*—the final answer is correct and every anchor is verifiably grounded in the source table; **(P2)** *hallucinated*—at least one anchor references a coordinate or value the image does not support; **(P3)** *shortcut correct*—the final answer is correct but the declared chain provides no valid evidence, indicating reliance on lexical priors rather than visual reasoning.

Sampling and annotation protocol. We uniformly sample 500 trajectories per model from the combined TQA test pool (100 each from FinQA, HiTab, TAT-QA, TabMWP, and WTQ), covering both correct and incorrect predictions in their natural proportions. Each trajectory is independently classified by two annotators following a decision tree: (i) verify final-answer correctness against ground truth; (ii) check every emitted anchor against the parsed table grid; (iii) assign one of the four categories (P1, P2, P3, or *other error*—incorrect answer with at least partially valid anchoring, typically caused by arithmetic mistakes downstream of correct evidence). Inter-annotator agreement is Cohen's $\kappa = 0.84$, with disagreements adjudicated by a third annotator. Importantly, the P1/P3 distinction is the hardest to draw under outcome-only evaluation and is exactly what outcome supervision cannot detect—our anchor-level audit makes it operationally measurable.

Table 12: Behavioral breakdown over 500 sampled trajectories per model. P1: faithful correct; P2: hallucinated (any answer); P3: shortcut correct (answer correct but anchors invalid). Higher P1 and lower P2/P3 indicate more grounded reasoning. Inter-annotator $\kappa = 0.84$.

Model	P1 ↑	P2 ↓	P3 ↓	Other Err.
Qwen3-VL-2B (SFT)	32.4%	38.6%	13.8%	15.2%
V-tableR1 (2B)	47.1%	26.2%	5.4%	21.3%
Qwen3-VL-4B (SFT)	36.8%	35.4%	11.7%	16.1%
V-tableR1 (4B)	**53.6%**	**21.8%**	**3.7%**	20.9%

P1 rises, P2 and P3 fall together. Table 12 reveals a behavioral shift that outcome accuracy alone obscures. From SFT to V-tableR1, faithful correct trajectories (P1) increase by $+14.7$ percentage points at 2B and $+16.8$ at 4B—gains that are roughly $1.5\times$ larger than the corresponding aggregate accuracy improvement. The extra margin comes from the simultaneous collapse of shortcut correct trajectories (P3: $13.8\% \rightarrow 5.4\%$ at 2B, $11.7\% \rightarrow 3.7\%$ at 4B), which under outcome evaluation would have been indistinguishable from P1. In other words, PGPO does not merely answer more questions correctly; it increases the fraction of correct answers whose intermediate evidence is explicitly checked against the table grid.

Hallucination is reduced, not displaced. A second concern when training against an anchor-aware critic is that the policy might learn to suppress shortcut behavior by fabricating plausible-looking anchors instead—displacing P3 into P2 rather than eliminating it. The data show the opposite: hallucinated trajectories (P2) also drop substantially (-12.4 pp at 2B, -13.6 pp at 4B), indicating that the critic's process signal penalizes invalid anchors with sufficient precision that the policy cannot game it by anchor fabrication. The slight relative growth in the *other error* category ($15.2\% \rightarrow 21.3\%$ at 2B) reflects a benign shift: as the policy learns to anchor faithfully, residual failures concentrate on downstream arithmetic over correctly identified operands—the very failure mode our critic taxonomy is designed to leave for upstream improvement (Section 3.3).

Implication for the Verifiability Gap. This breakdown is the most direct empirical evidence for our central claim: process supervision changes *how* the model reaches its answer, not just *whether* it does. Under outcome-only training, P1 and P3 are observationally identical and the model is free

to maximize the cheaper of the two. Anchored Reasoning makes the distinction inspectable, and PGPO shifts the population toward P1 by a margin that exceeds the accuracy headline—precisely the regime the Verifiability Gap was introduced to characterize.

E Future Generalization Sketch

In the main paper, we define each visual anchor as $a_k = (\mathcal{C}_k, \mathcal{V}_k, \rho_k)$, where $\rho_k \in \mathcal{R}$ denotes the structural role assigned to the referenced evidence. We use an open role ontology rather than a fixed small label set because tables vary substantially in layout, semantics, and reasoning requirements. A table, for example, requires roles such as headers and operands, while forms, charts, and scientific diagrams may require roles that capture fields, legends, axes, regions, or relational components. The goal of $\mathcal{R}$ is therefore not to enumerate all possible document elements in the main text, but to provide a unified interface through which heterogeneous visual evidence can be converted into verifiable symbolic references.

Tables. For tables, we use roles including `row-header`, `column-header`, `data-cell`, `operand`, `condition`, `aggregation-set`, and `result`. These roles are designed to cover the core operations in tabular reasoning: locating evidence, associating values with headers, filtering rows or columns under conditions, aggregating multiple cells, and deriving intermediate or final results.

Forms and key-value documents. For forms, receipts, invoices, and other key-value documents, we use roles including `field-name`, `field-value`, `key`, `value`, `checkbox`, `signature`, `date`, and `total`. These roles reflect the fact that reasoning over forms is often based on matching a queried field to its corresponding value, verifying whether a field is present or selected, and checking document-level summaries such as dates or totals.

Charts. For charts, we use roles including `x-axis`, `y-axis`, `axis-label`, `legend`, `series`, `data-point`, `bar`, `line-segment`, and `trend`. These roles are necessary because chart reasoning typically involves mapping visual marks to axes and legends, comparing values across data points, and identifying trends or extrema.

Scientific diagrams and structured figures. For scientific diagrams and structured figures, we use roles including `entity`, `component`, `region`, `arrow`, `relation`, `label`, and `measurement`. These roles allow anchors to represent not only localized objects, but also relations between objects, directed flows, annotated regions, and quantitative measurements.

Design rationale. The role ontology follows three principles. First, each role should correspond to evidence that can be localized or structurally identified in the source document. Second, each role should support verification of a reasoning step, either by checking a value, a region, a relation, or an operation over multiple elements. Third, the ontology should remain extensible: new document types may introduce additional roles, but they must still instantiate the same anchor interface $a_k = (\mathcal{C}_k, \mathcal{V}_k, \rho_k)$. This design allows Anchored Reasoning to remain specific enough for verification while general enough to extend beyond tables.

F Ablation on Reward Design

We further analyze the effect of reward formulation in PGPO. In the main experiments, we adopt an interval-gated composite reward, where the process score modulates answer credit only through thresholded regions. This design avoids directly injecting the continuous critic score into the reward, which can be sensitive to critic calibration noise during early-stage RL.

To verify this choice, we compare our interval-gated reward with a direct process-reward variant on five table question answering datasets: FinQA, HiTab, TAT-QA, TabMWP, and WTQ. The direct variant replaces the interval gate with a continuous reward term:

$$R_{\text{direct}}(y) = r_{\text{fmt}} + r_{\text{acc}} + \eta r_{\text{proc}}, \qquad (13)$$

where η controls the contribution of the critic score. For a fair comparison, we keep the same 2B policy model, training data, rollout configuration, and optimization hyperparameters, and only change the reward formulation.

As shown in Table 13, directly incorporating r_{proc} into the reward leads to a consistent performance drop of 1–4 percentage points across datasets. We attribute this degradation to the fact that critic scores are continuous but not perfectly calibrated, especially during early RL iterations. Small fluctuations in r_{proc} may introduce unstable reward differences between otherwise similar

Table 13: Ablation on reward formulation with the 2B policy model. Directly adding the continuous process score to the reward consistently underperforms the interval-gated design, indicating that thresholded process gating is more robust to critic-score noise.

Dataset	Interval-gated	Direct Process	Δ
FinQA	24.49	22.86	-1.63
HiTab	45.45	42.97	-2.48
TAT-QA	49.83	47.28	-2.55
TabMWP	78.94	75.31	-3.63
WTQ	56.49	53.74	-2.75
Avg.	51.04	48.43	-2.61

trajectories. In contrast, our interval-gated formulation only uses the critic score to distinguish high-confidence faithful reasoning, low-confidence shortcut reasoning, and ordinary acceptable reasoning. This reduces sensitivity to noisy critic scores while preserving the ability to penalize reward hacking.

Threshold sensitivity. We further study the sensitivity of the interval-gated reward to the process thresholds τ_h and τ_ℓ. We evaluate several threshold combinations on five TQA datasets using the same 2B policy model and training configuration. For compactness, we report the average accuracy across FinQA, HiTab, TAT-QA, TabMWP, and WTQ.

Table 14: Sensitivity analysis of the process-score thresholds in the interval-gated reward. Results are averaged over FinQA, HiTab, TAT-QA, TabMWP, and WTQ with the 2B policy model.

τ_ℓ	τ_h	Avg. Accuracy
0.4	0.7	49.92
0.4	0.8	50.36
0.4	0.9	49.71
0.5	0.7	50.62
0.5	0.8	**51.04**
0.5	0.9	50.41
0.6	0.8	50.17
0.6	0.9	49.85

As shown in Table 14, the configuration $\tau_\ell = 0.5$ and $\tau_h = 0.8$ achieves the best average performance. A lower high-confidence threshold, e.g., $\tau_h = 0.7$, tends to over-reward moderately reliable reasoning trajectories, while a higher threshold, e.g., $\tau_h = 0.9$, makes the bonus too conservative and reduces the number of high-confidence positive updates. Similarly, a low τ_ℓ weakens the penalty for unreliable reasoning, whereas a high τ_ℓ may over-penalize partially correct processes. We therefore use $\tau_h = 0.8$, $\tau_\ell = 0.5$, $\alpha = 0.5$, and $\beta = 0.2$ in all PGPO experiments unless otherwise specified.

G Ablation on Length-aware Active Set

We further validate the length-aware active set used in PGPO. In the main method, PGPO updates the policy using rollouts from two selected length regions: the shortest 0–25% and the 60–85% percentile range within each sampled group. This design is motivated by the observation that rollout length provides a coarse but useful stratification signal for anchored visual reasoning. Short trajectories often expose concise anchoring decisions, while moderately long trajectories are more likely to contain multi-step evidence aggregation, comparison, or self-correction. In contrast, overlong trajectories tend to introduce redundant reasoning or unstable exploration.

To verify that this selection is empirically justified, we compare our active-set strategy with several alternatives. All variants use the same 2B policy model, training data, rollout configuration, reward design, and optimization hyperparameters; only the active set used for policy updates is changed. We evaluate on five TQA datasets: FinQA, HiTab, TAT-QA, TabMWP, and WTQ.

As shown in Table 15, using all sampled rollouts is suboptimal, suggesting that not every trajectory contributes equally useful gradients. Updating only on the shortest rollouts improves compactness but misses multi-hop reasoning patterns, while using only the 60–85% region preserves more complex reasoning but lacks concise high-confidence anchoring cases. The overlong 85–100% tail performs the worst, indicating that excessively long trajectories are more likely to contain redundant reasoning or unstable exploration.

The proposed 0–25% and 60–85% partition achieves the best average performance. Compared with the broader 0–30% and 60–90% partition, our final setting removes more boundary cases from both the short and long regimes, leading to more stable policy updates. These results support the use of rollout length as a coarse but effective stratification signal for selecting high-value trajectories in anchored visual reasoning.

H Effect of Critic Model Scale

A natural concern is whether the performance gains of V-tableR1 are contingent on the large-scale 32B critic, or whether the PGPO framework itself pro-

Table 15: Ablation on length-aware active-set construction with the 2B policy model. The proposed 0–25% and 60–85% partition achieves the best overall performance across five TQA datasets.

Active Set	FinQA	HiTab	TAT-QA	TabMWP	WTQ	Avg.
All rollouts	23.71	44.12	48.63	77.25	55.58	49.86
Shortest 0–25% only	22.98	43.86	48.21	77.43	54.62	49.42
Middle 25–60% only	22.64	43.11	47.82	76.91	54.07	48.91
Long 60–85% only	23.95	44.58	48.91	77.80	55.66	50.18
Overlong 85–100% only	21.93	42.75	47.26	76.32	53.59	48.37
0–30% and 60–90%	24.08	44.91	49.26	78.31	55.49	50.41
0–25% **and** 60–85%	**24.49**	**45.45**	**49.83**	**78.94**	**56.49**	**51.04**

vides the core benefit. To isolate this, we train an additional variant in which the critic $\mathcal{C}_\phi$ is instantiated from Qwen3-VL-4B rather than Qwen3-VL-32B, keeping all other hyperparameters identical to the main experiments.

Table 16 reports results for the 4B policy model under both critic configurations, together with the SFT baseline. Two conclusions follow. First, V-tableR1 with a 4B critic still consistently outperforms the SFT baseline across all benchmarks, confirming that the benefit of process-supervised RL is not an artifact of critic scale. Second, replacing the 32B critic with a 4B critic leads to a measurable performance drop (TFV: $\Delta = -1.08$ pp; TQA: $\Delta = -3.51$ pp on average), indicating that critic perceptual fidelity does constitute a meaningful upper bound on the quality of the process signal. Notably, the performance gap between the two critic configurations is larger on TQA than on TFV, consistent with the intuition that complex multi-hop numerical reasoning imposes higher demands on anchor verification precision than binary fact checking. These results motivate future work on lightweight critic distillation as discussed in Section M.

The gains are not passive distillation of the critic. A related concern is whether V-tableR1's gains merely reflect the critic's own task capability being distilled into a smaller policy. The evidence in Table 2 and Table 16 argues against this interpretation. The Qwen3-VL-32B critic backbone, evaluated zero-shot, attains TFV/TQA averages of 80.34% and 54.92%, whereas V-tableR1 (4B) with the same critic reaches 88.45% and 55.44%—surpassing its own supervisor on both task families despite having 8× fewer parameters. Furthermore, V-tableR1 (4B) with the 4B critic still reaches 51.93% on TQA, exceeding the 32B critic's zero-shot score even when the critic and policy share the same scale. Since the policy systematically outperforms the model from which its supervision is derived, the improvement cannot be attributed to passive capability transfer; the process-level reward shaping introduced by PGPO contributes a signal that is not present in the critic's own task predictions.

Robustness across critic model families. A complementary concern is whether V-tableR1's gains depend specifically on a Qwen-family critic, given that the policy backbone (Qwen3-VL) and the deployed critic ($\mathcal{C}_\phi$, initialized from Qwen3-VL-32B) share a common pretraining lineage. To probe this, we train an additional critic variant initialized from InternVL-2.5-38B, which differs from Qwen3-VL in both its visual encoder (InternViT) and its multimodal alignment recipe, while remaining at a comparable parameter scale (∼38B vs. 32B). All other components of the data-construction and PGPO training pipelines are held fixed. We then evaluate V-tableR1 (4B) under this alternative critic across the same seven benchmarks.

As shown in Table 17, swapping the critic to an architecturally distinct backbone leads to only a marginal performance change (-0.83 pp on TFV and -0.93 pp on TQA), well within the fluctuation range observed under label-weight perturbations in Appendix **??**. This small gap suggests that PGPO's benefit is not tied to the Qwen pretraining lineage shared between the deployed critic and the policy. Combined with the scale ablation above and the cross-architecture agreement study (Appendix **??**), the evidence indicates that the process signal extracted by $\mathcal{C}_\phi$ tracks anchor fidelity itself rather than family-specific perceptual artifacts.

Table 16: Effect of critic model scale on V-tableR1 (4B policy). All variants share the same PGPO training configuration. †: critic fixed at 4B; ‡: critic fixed at 32B (default).

Model	TFV			TQA					
	TabFact	InfoTabs	Avg.	FinQA	HiTab	TAT-QA	TabMWP	WTQ	Avg.
Qwen3-VL-4B (SFT)	83.88	87.33	85.61	18.88	36.58	46.63	76.60	54.32	46.60
V-tableR1 (4B, Critic-4B†)	86.42	88.31	87.37	24.93	43.12	51.19	80.67	59.75	51.93
V-tableR1 (4B, Critic-32B‡)	**87.95**	**88.94**	**88.45**	**28.98**	**47.24**	**54.23**	**83.38**	**63.37**	**55.44**

Table 17: Effect of critic model family on V-tableR1 (4B policy). All variants share the same PGPO configuration and perturbation-based label construction.

Critic Backbone	TFV Avg.	TQA Avg.
Qwen3-VL-32B (default)	**88.45**	**55.44**
InternVL2.5-38B	87.62	54.51
Δ (cross-family)	-0.83	-0.93

I Text-Only Baseline: Table-as-Markdown Verification

A natural concern is whether the gains of V-tableR1 stem from genuine *visual grounding*, or could be replicated by parsing the table into symbolic text and applying a text-only LLM. To isolate this, we construct a controlled text-only baseline that removes the visual modality while keeping the backbone, reasoning prompt, and verification protocol identical to the multimodal setting.

I.1 Protocol

For every test instance across the seven benchmarks, we replace the rendered table image with its Markdown serialization, obtained by parsing the ground-truth table structure (rows, columns, hierarchical headers, merged cells) into pipe-delimited Markdown. Parent headers are duplicated across spanned columns to preserve column-to-header mappings. This yields a lossless symbolic representation, removing any perceptual difficulty associated with OCR or layout parsing.

We prompt the text-only backbone with the *same anchored reasoning instruction* as the multimodal pipeline, including the `(coordinate, value, role)` triple requirement at each step. Final answers are verified under the same evaluation rule. We test Qwen3-4B and Qwen3-2B (text-only) to match the parameter scale of V-tableR1. This baseline operates under an *oracle parse* assumption: it receives a perfectly structured Markdown table, a strictly stronger input condition than any practical OCR pipeline. The resulting accuracies thus represent an *upper bound* on what text-only pipelines can achieve.

I.2 Results

Table 18 reports the comparison. Even under the oracle-parse assumption, the text-only baselines fall substantially behind V-tableR1 at the same parameter scale. Qwen3-4B (Markdown) trails V-tableR1 (4B) by 6.84 points on TFV and 7.91 points on TQA, despite operating on a perfectly structured input. The gap is most pronounced on HiTab (hierarchical headers) and TAT-QA (hybrid table–text evidence), where the flattened Markdown loses two-dimensional spatial cues that the anchored visual chain-of-thought exploits.

I.3 Discussion

Two observations follow. First, text-only baselines themselves perform reasonably on TFV, confirming that a non-trivial portion of table reasoning can be mitigated by symbolic representation alone. Second, the consistent multimodal advantage of $+6$–8 points across both scales shows that V-tableR1's gains are *not* reducible to implicit OCR plus text reasoning: even given an oracle parse, the text-only model lacks the visual evidence binding that anchored reasoning enforces over the rendered image. The gap scales with structural complexity—larger on HiTab and TAT-QA than on the regular tables of TabMWP—indicating that visual anchors preserve information that flattening to Markdown discards. In realistic deployment, where OCR introduces additional parsing error, this gap can only widen.

J Reasoning Quality Assessment

Beyond terminal accuracy, we further evaluate whether V-tableR1 improves the quality of intermediate reasoning trajectories. We conduct an LLM-as-a-judge evaluation on 1,000 randomly sampled instances from the combined test sets. For each instance, we provide the judge with the table image,

Table 18: Text-only baselines (oracle Markdown) versus multimodal V-tableR1. All models share the same anchored prompt and verification protocol. Best results in **bold**.

Model	TFV			TQA					
	TabFact	InfoTabs	Avg.	FinQA	HiTab	TAT-QA	TabMWP	WTQ	Avg.
Text-only (Markdown, oracle parse)									
Qwen3-2B (Markdown)	76.34	80.12	78.23	18.92	33.27	40.18	71.43	49.86	42.73
Qwen3-4B (Markdown)	80.81	82.41	81.61	22.46	38.74	44.05	75.62	56.78	47.53
Multimodal (V-tableR1)									
V-tableR1 (2B)	84.69	87.54	86.12	24.49	45.45	49.83	78.94	56.49	51.04
V-tableR1 (4B)	**87.95**	**88.94**	**88.45**	**28.98**	**47.24**	**54.23**	**83.38**	**63.37**	**55.44**

the question, and the generated reasoning trajectory, and ask it to score the trajectory along three dimensions: *Accuracy*, *Logic*, and *Visual Grounding*. Scores are assigned on a 1–5 scale, where higher values indicate better reasoning quality.

Evaluation criteria. *Accuracy* measures whether intermediate factual claims, extracted values, and numerical operations are correct. *Logic* measures whether the reasoning steps are structurally coherent, non-contradictory, and sufficient to support the final answer. *Visual Grounding* measures whether the reasoning trajectory correctly localizes the relevant table regions and binds intermediate claims to explicit visual evidence, such as cell coordinates or structural anchors.

As shown in Table 19, the GPT-5 judge scores V-tableR1 higher on all three dimensions, with the largest gain in Visual Grounding. The grounding score increases from 2.3 to 3.8, indicating that process-supervised RL substantially improves the model's ability to bind reasoning steps to correct table regions. The baseline already exhibits relatively strong logical fluency, but its low grounding score suggests that fluent reasoning does not necessarily imply faithful visual evidence use. In contrast, V-tableR1 improves intermediate accuracy and grounding simultaneously, supporting our claim that critic-gated process supervision helps bridge the gap between abstract reasoning and verifiable visual evidence.

Judge prompt. We use the following instruction for reasoning-quality assessment:

Prompt for Reasoning Quality Assessment

System Instruction:
You are an expert evaluator of multimodal table reasoning. Your task is to assess the quality of a model-generated reasoning trajectory given a table image, a question, and the model's reasoning process.

User Input:
You are given:

- a table image;
- a natural-language question;
- a model-generated reasoning trajectory.

Evaluation Dimensions:

- **Accuracy:** whether the intermediate factual claims, extracted values, and calculations are correct.
- **Logic:** whether the reasoning steps are coherent, non-contradictory, and sufficient to support the final answer.
- **Visual Grounding:** whether the trajectory correctly grounds intermediate claims to relevant table cells, coordinates, headers, or visual regions.

Scoring Rule:
Assign each dimension a score from 1 to 5, where 1 indicates very poor quality and 5 indicates excellent quality.

Output Format:
Return only the three scores and a brief justification:

```
{
  "accuracy": 1-5,
  "logic": 1-5,
  "visual_grounding": 1-5,
  "justification": "brief explanation"
}
```

Figure 6: Prompt used for LLM-as-a-judge reasoning quality assessment. The judge evaluates each generated trajectory along accuracy, logic, and visual grounding dimensions.

K Code and Data Release

To support reproducibility, we commit to releasing the full V-tableR1 codebase upon acceptance, including the policy and critic training pipelines, the PGPO implementation with all hyperparameters reported in Table 4, the controlled pertur-

Table 19: Reasoning quality assessment on 1,000 randomly sampled test instances. Scores are assigned on a 1–5 scale.

Model	Accuracy	Logic	Visual Grounding
Qwen3-VL-2B (SFT)	2.8	4.0	2.3
V-tableR1-2B	**3.4**	**4.2**	**3.8**

bation scripts used to construct critic supervision (Sec. 3.3), and the evaluation harness for all seven benchmarks. We will additionally release the verified anchored trajectories used for SFT, the perturbed trajectories and process-score labels used for critic training, and the trained V-tableR1 (2B/4B) and Critic-32B checkpoints under a research-friendly license. Prompts for the GPT-4.1 trajectory-generation teacher, the Gemini-2.5-Flash verifier, and the Qwen3-32B perturbation model will be released as-is so that the data-construction pipeline can be reproduced with alternative backbones when proprietary model access is unavailable.

L Prompt Templates for Structured Response Generation

We use the following prompt templates to guide structured response generation from table images. The model is instructed to output a reasoning process and a final answer within separate tags. In the reasoning process, the model is required to explicitly cite table-relative coordinates for the visual evidence it uses.

M Future Work

Several directions naturally extend Anchored Reasoning beyond its current instantiation.

Generalization beyond tables. The open role ontology introduced in Appendix E offers a conceptual foothold for extending Anchored Reasoning to charts, forms, and scientific diagrams, but the corresponding critic design and perturbation taxonomy remain to be developed. We view this generalization as the central scientific question that the paradigm ultimately seeks to answer.

Lightweight and self-rewarding verifiers. Promising directions for reducing critic overhead include distilling the critic's process signal into a lightweight verifier co-trained with the policy, exploring self-rewarding schemes in which the policy bootstraps from its own anchor consistency, and replacing the regression-based critic with a generative verifier that produces natural-language critiques alongside scalar scores—potentially improving calibration on out-of-distribution trajectories.

Online critic refinement. Our critic is trained on synthetically perturbed trajectories, which yields clean supervision but may underrepresent compositional failure modes that emerge from real policy rollouts. A natural next step is online critic refinement, in which the verifier is periodically updated on hard negatives mined from the policy's own trajectories during PGPO training.

Scaling laws of process supervision. Our results suggest that process supervision can be an effective complement to parameter scaling for structured visual reasoning, but we have not characterized this trade-off systematically. Establishing scaling laws that relate critic quality, anchor density, and policy capacity would clarify when process supervision saturates and when raw scale remains the dominant lever.

Agentic and tool-augmented settings. The verifiability afforded by anchored trajectories has natural implications beyond static QA. In agentic settings—spreadsheet manipulation, financial analysis, scientific data extraction—the explicit (coordinate, value, role) interface can serve as an auditable contract between the model and downstream tools, enabling selective intervention when anchors fail verification.

System Prompt for Structured Response Generation

A conversation between User and Assistant. The user asks a question, and the assistant solves it. The assistant first writes the reasoning process and then provides the answer. The reasoning process and answer are enclosed within `<think>` `</think>` and `<answer>` `</answer>` tags, respectively, i.e., `<think>` reasoning process here `</think>` `<answer>` answer here `</answer>`.

In the `<think>` section, the assistant should explicitly cite the table-relative coordinates of the visual evidence used in reasoning, using the format `<cell: Row x, Col y>`.

Figure 7: System prompt used to guide structured response generation.

Prompt Templates for TQA and TFV

Prompt Template for TQA

Instruction: This is a short-answer table QA task. Answer the question based on the provided table image. In the `<think>` section, cite the table-relative coordinates of the visual evidence used in reasoning, using the format `<cell: Row x, Col y>`.

Table
Table Image: `{table_image}`

Question: `{question}`

Output Format:
The output should use the following format:

```
<think>
reasoning process with cited table-relative coordinates
</think>
<answer>
{
  "answer": ["answer1", "answer2", ...]
}
</answer>
```

Prompt Template for TFV

Instruction: This is a table fact verification task. The goal is to determine whether the given statement is entailed or refuted by the provided table image. In the `<think>` section, cite the table-relative coordinates of the visual evidence used for verification, using the format `<cell: Row x, Col y>`.

Table
Table Image: `{table_image}`

Statement: `{statement}`

Output Format:
The output should use the following format:

```
<think>
reasoning process with cited table-relative coordinates
</think>
<answer>
{
  "answer": "entailed" or "refuted"
}
</answer>
```

Figure 8: Prompt templates for table question answering and table fact verification from table images.