# Diagnosing Failure Modes of Shared-State Collaboration in Resource-Constrained Visual Agents

**Anonymous Authors**[1]

## Abstract

Modular visual reasoning systems increasingly rely on shared working memory for multi-step collaboration, yet the failure dynamics of intermediate state evolution in low-capacity regimes remain underexplored. We study failure modes of collaborative reasoning with weak learners (4B–8B models) through the lens of noise accumulation. We introduce CoSee, an auditing framework that formalizes the read-write-verify loop to trace information flow in document visual question answering. Across multi-page, chart, and web-based benchmarks, we find a counter-intuitive degradation: naive shared workspaces often amplify hallucinations rather than resolve them. We identify two dominant failure modes: Noise Reinforcement, where ungrounded notes are reused as evidence, and Policy Collapse, where added context shifts the model toward under-specified, short-form answers. Using cost-accuracy Pareto frontiers, we show that increased compute can correlate negatively with performance without explicit verification. Our findings suggest that for resource-constrained agents, the bottleneck lies not in reasoning depth but in communication fidelity, providing trace-level diagnostics and a mechanistic baseline for reliable modular design.

## 1. Introduction

Recent advances in Vision Language Models (VLMs) have enabled reasoning over high-information density artifacts, such as multi-page slides, quantitative charts, and web documents. To tackle these complex inputs, the field is increasingly moving toward modular designs that utilize shared workspaces (e.g., whiteboards) to externalize intermediate states. In principle, this form of external working memory allows limited-capacity models to "think on paper": decomposing intractable queries into verifiable steps of reading, writing, and cross-checking. The prevailing intuition suggests that such explicit state tracking inherently promotes reliability by reducing the cognitive load on a single inference pass.

However, the efficacy of this paradigm remains unproven under the very boundaries that motivate its use: resource constrained regimes characterized by weak learners (e.g., small models) and strict compute budgets. In this setting, we hypothesize that the collaborative loop acts as a noisy communication channel rather than a reliable memory store. Every additional turn not only incurs a linear computational cost but introduces a risk of noise accumulation: collaboration alters the conditioning distribution at each step, potentially leading to policy drift (e.g., shifts in verbosity or coverage) or hallucination reinforcement, where ungrounded intermediate notes are "hardened" into incorrect evidence. These failure modes are often silent and diagnostically opaque without rigorous integrity auditing.

To investigate these limits, we present CoSee, an auditing framework and collaboration protocol designed to rigorously measure the utility of shared context under small-model constraints. CoSee formalizes the agentic loop into a minimal action interface (read, write-to-board, verify), logs structured traces of intermediate states, and enforces strict output integrity checks to separate reasoning failures from format errors . unlike ad-hoc evaluations, we benchmark all configurations with strictly matched prompts and decoding caps, ensuring that any observed differences stem from the interaction dynamics rather than hidden computational advantages. We utilize this pipeline to study three configurations: direct single-turn answering, a single-agent board variant, and a two-agent scanner–checker architecture.

We conduct our empirical study across three representative distributions—SlideVQA, ChartQAPro, and VQAonline—using Qwen3-VL-4B-Instruct as the primary backbone, with scale and robustness checks on Qwen3-VL-8B, Phi-4, and Gemma-3-4B. Our results reveal a distinct efficiency paradox: naive usage of shared memory yields at most marginal gains on retrieval tasks and often degrades performance on reasoning-heavy tasks. Specifically, we find that the

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

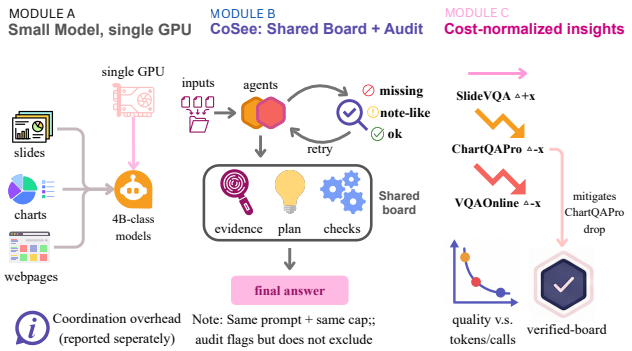



*Figure 1.* **CoSee overview.** Shared-board collaboration with trace logging and integrity auditing enables controlled, cost-normalized evaluation under strict budgets. Our study finds that naive board use is not a reliable win under small-model constraints, while a lightweight verified-board gate mitigates chart-centric failures.

multi-agent setting frequently underperforms the single-turn baseline due to coordination overhead and coverage failures. To quantify these trade-offs, we analyze cost–accuracy Pareto frontiers, showing that increased token usage often correlates negatively with performance in the absence of quality control. Trace-grounded diagnostics further isolate the mechanisms behind this degradation—primarily noise reinforcement on charts and policy collapse (short-answer drift) on open-ended QA—demonstrating that lightweight verification is the minimal necessary condition to arrest error propagation.

**Contributions.** We introduce CoSee, a formal framework for auditing the information flow and integrity in modular VQA systems, which enables a systematic empirical investigation into the limits of weak learner collaboration. By integrating compute-aware metrics—specifically cost-accuracy Pareto frontiers and format-robust scoring—we quantify the operational boundaries where external memory shifts from an asset to a liability due to noise accumulation. Furthermore, we establish a taxonomy of dominant failure mechanisms, including noise reinforcement and policy collapse, and empirically validate that implementing a grounded information bottleneck via lightweight verification is the minimal necessary condition for reliable scaling in resource-constrained document VQA.

## 2. Related Work

**Document and chart VQA as reasoning testbeds.** Recent datasets have shifted towards high-information density artifacts, establishing document and chart QA as rigorous testbeds for grounded reasoning in complex visual contexts (Mathew et al., 2021; Masry et al., 2022; Nguyen et al., 2025). Specifically, ChartQAPro introduces adversarial perturbations and unanswerable queries, exposing the fragility of current LVLMs in handling out-of-distribution chart types (Masry et al., 2025). Complementarily, VQAonline challenges models with open-ended generation from authentic web data, where evaluation metrics must disentangle semantic correctness from policy shifts in coverage and verbosity (Chen et al., 2024). These distributional shifts necessitate evaluation protocols that explicitly account for formatting sensitivity and the computational cost of generating adequately supported answers, moving beyond raw accuracy metrics (Rajpurkar et al., 2016).

**Architecture-centric document understanding.** Prior work has largely focused on encoding inductive biases for document structure, improving understanding by exploiting layout signals and multi-page context (Xu et al., 2021b;a). Approaches such as DocLLM and LayoutLLM optimize instruction tuning for structural comprehension (Wang et al., 2024; Luo et al., 2024), while architectures like DocOwl2 address the efficiency–accuracy trade-off via high-resolution compression mechanisms (Hu et al., 2025). In parallel, open LVLMs continue to scale vision encoders to handle multi-image resolution dynamics (e.g., LLaVA-UHD, LLaVA-OneVision) (Guo et al., 2024; Li et al., 2024a), and recent technical reports (e.g., Qwen3-VL) highlight strengthened foundational capabilities in document parsing (Bai et al., 2025). Our work is orthogonal to these architectural advances: rather than proposing a new backbone, we isolate the inference-time dynamics of protocol-level collaboration via shared memory in the fixed small-model regime, analyzing its interaction with output formats and semantic coverage (Kim et al., 2022; Lee et al., 2023).

**Modular agents and orchestration costs.** Interactive multimodal agents are increasingly deployed in dynamic environments, such as GUI navigation, where success depends on grounded perception and sequential planning (Koh et al., 2024; Hong et al., 2024; Zheng et al., 2024). Recent research explores modular reasoning via role specialization and iterative verification (Li et al., 2024b), specifically within document-focused pipelines (Han et al., 2025; Sun et al., 2025; Jain et al., 2025; Yu et al., 2025). However, the utility of such orchestration is often obscured by confounding factors, including heavier backbone capacities or unnormalized interaction costs (Wei et al., 2022; Wang et al., 2022; Yao et al., 2023). In contrast, we present a formal auditing framework for shared-workspace protocols, focusing on output integrity and trace artifacts. We emphasize cost-normalized comparisons via compute-logged experiments to identify the specific mechanisms—such as noise accumulation—that cause collaboration to degrade performance in resource-constrained VQA (Wang et al., 2025; Jiang et al., 2024; Yi et al., 2025; Kugo et al., 2025).

# 3. Method

We formulate the problem of document visual question answering as a sequential decision-making process under strict computational constraints. Our goal is to analyze the inference dynamics of weak learners when augmented with external working memory.

## 3.1. Problem Formulation

Let $\mathcal{M}_\theta$ denote a vision–language model with parameters $\theta$, subject to a resource constraint $\Omega$ (defined by limited parameter size, e.g., 4B–8B, and single-GPU memory). Given a visual input $I = \{I_k\}_{k=1}^{K}$ (representing slides, charts, or documents) and a natural language query $q$, the objective is to generate an answer $a \in \mathcal{A}$ that maximizes the likelihood $P(a|I, q)$.

In a collaborative setting, the generation of $a$ is mediated by a sequence of intermediate discrete latent states $B_{1:T}$, representing the trajectory of the shared workspace. The inference process seeks to approximate the intractable posterior of the answer given the reasoning chain:

$$P(a|I, q) \approx \sum_{B_T} P_\theta(a|I, q, B_T) \prod_{t=1}^{T} P_\theta(B_t|I, q, B_{t-1}) \tag{1}$$

We focus on the *inference-time* behavior of $\mathcal{M}_\theta$, treating the weights as fixed.

## 3.2. External Working Memory (The Board)

To externalize the intermediate reasoning steps, CoSee introduces a structured latent state, the **Board** ($B_t$). Unlike internal hidden states, $B_t$ is a discrete, interpretable sequence of entries readable by all agents. Formally, at time step $t$, the board state is defined as an ordered sequence $B_t = [e_1, e_2, \ldots, e_t]$, where each entry $e_i$ is a tuple $(\mathbf{m}, \mathbf{p}, \texttt{id})$. Here, $\mathbf{m}$ represents a natural language note (evidence or hypothesis), $\mathbf{p}$ denotes an optional grounding pointer (e.g., page index or region coordinates), and $\texttt{id}$ identifies the authoring agent.

The transition dynamics of the board are governed by a lightweight controller that enforces a strict read-write interface. At each step $t$, the controller exposes the current state $B_{t-1}$ to the active agent policy $\pi$, which samples a new update $e_t$. The state evolves deterministically as $B_t = B_{t-1} \oplus e_t$, where $\oplus$ denotes the append operation. This design ensures that the entire reasoning trajectory is observable and auditable.

## 3.3. Inference Dynamics and Protocols

We investigate three distinct inference protocols, viewing them as variations in the state transition probability $P(B_t|B_{t-1})$ and the readout function.

**Direct Inference (Baseline).** The baseline represents a collapse of the sequential process into a single step. The model directly samples the output distribution conditioned only on the input:

$$\hat{a} \sim P_\theta(a|I, q) \tag{2}$$

This serves as the lower bound for computational cost and the reference point for reasoning quality.

**Open-Loop Iterative Refinement (Single & Multi-Agent).** In this regime, agents interact with the board for a fixed budget of $T$ steps without explicit external feedback.

- **Single-Agent Dynamics:** A single policy $\pi_\theta$ iteratively updates the board. At step $t$, the agent samples $e_t \sim \pi_\theta(\cdot|I, q, B_{t-1})$.
- **Dual-Agent Dynamics (Scanner–Checker):** We instantiate two complementary roles, $\pi_{\text{scan}}$ and $\pi_{\text{check}}$, derived from the same backbone $\mathcal{M}_\theta$ via prompting. The agents operate in alternating turns, effectively modeling a cooperative game where $\pi_{\text{scan}}$ proposes evidence and $\pi_{\text{check}}$ refines or refutes it based on the shared context $B_{t-1}$.

Crucially, in both settings, the final answer is generated via a dedicated readout step $\hat{a} \sim P_\theta(a|I, q, B_T)$, ensuring that intermediate artifacts ($e_{1:T}$) do not leak strictly into the answer format unless explicitly attended to.

**Gated Transition with Information Bottleneck (Verified-Board).** To mitigate the propagation of ungrounded hallucinations (noise), we introduce a conditional gating mechanism on the state transition. Let $V_\phi(e, I, q) \in \{0, 1\}$ be a verification function (parameterized by the same backbone $\mathcal{M}_\theta$) that maps a proposed entry $e$ to a binary validity score. The update rule becomes:

$$B_t = \begin{cases} B_{t-1} \oplus e_t & \text{if } V_\phi(e_t, I, q) = 1 \text{ (Supported)} \\ B_{t-1} & \text{otherwise (Discarded)} \end{cases} \tag{3}$$

This mechanism acts as an *information bottleneck*, filtering out high-entropy or hallucinated notes before they become part of the conditioning context for subsequent steps. We hypothesize that this control is essential for preventing noise accumulation in weak learners.

## 3.4. Contextual Conditioning and Policy Drift

To operationalize the collaborative roles, we construct the input context $\mathcal{C}_t$ by modifying the system instructions while keeping the visual backbone fixed. For the *Scanner*,

$\mathcal{C}_t = [\mathcal{I}_{\text{scan}}, I, q, B_{t-1}]$, where $\mathcal{I}_{\text{scan}}$ conditions the model to extract atomic visual evidences. For the *CrossChecker*, $\mathcal{C}_t = [\mathcal{I}_{\text{check}}, I, q, B_t]$, where $\mathcal{I}_{\text{check}}$ conditions on identifying contradictions between $I$ and $B_t$.

We posit that this augmented context introduces a *distributional shift* in the generation of the final answer. While the ideal objective is to approximate the true posterior $P(a|I, q)$, the dependency on intermediate states introduces a bias: $\hat{a} \approx \arg\max_a P_\theta(a|\text{summary}(B_T), q)$. In regimes where the backbone $\theta$ has limited attention capacity (e.g., small models), we hypothesize that the model may over-attend to the compressed textual summary in $B_T$ while under-attending to the high-dimensional visual features in $I$, a phenomenon we term **attention dilution**. Furthermore, if the intermediate notes $e_{1:T}$ exhibit a specific stylistic bias (e.g., extreme brevity), this style may transfer to the final answer via in-context learning, leading to **policy drift** where the output format diverges from the optimal answer length required by the task.

### 3.5. Traceability and Audit Framework

A core contribution of CoSee is the rigorous auditing of the inference trajectory. We treat the sequence $\tau = (I, q, B_{1:T}, \hat{a})$ as a probabilistic trace. To ensure the validity of our analysis, we implement an **integrity audit** function $\mathcal{I}(\hat{a}) \rightarrow \{\texttt{valid}, \texttt{malformed}, \texttt{leakage}\}$. This allows us to detect protocol-level pathologies—such as when intermediate notes are copied verbatim into the final answer (Leakage)—without introducing selection bias into the scoring. All examples, including those flagged as malformed, are retained in the evaluation to penalize protocol instability.

### 3.6. Cost–Utility Analysis

In resource-constrained regimes, performance cannot be decoupled from cost. We define the computational cost function $\mathcal{C}(\tau)$ as the total number of tokens generated across all intermediate steps and the final answer. We analyze the system's efficiency by plotting the empirical **Pareto frontier** between the utility $\mathcal{U}$ (accuracy metric) and cost $\mathcal{C}$:

$$\mathcal{F}^* = \max_\pi \mathbb{E}_{\tau\sim\pi}[\mathcal{U}(\hat{a}, a^*)] \quad \text{s.t.} \quad \mathbb{E}[\mathcal{C}(\tau)] \leq \beta \quad (4)$$

This formulation allows us to determine whether the additional compute invested in collaboration yields a super-linear, linear, or negative return on investment.

## 4. Experiments Setup

### 4.1. Evaluation Domains and Distributions

To rigorously test the limits of collaboration, we select three benchmarks that map to distinct failure modes in modular visual reasoning.

| Dataset | Split (Used) | $n$ | Metrics (reported) |
|---|---|---|---|
| SlideVQA | Official test | 600 | EM, Loose |
| ChartQAPro | Test | 800 | Raw: EM, Loose; Ans-only: $\text{EM}_{\text{ans}}$, $\text{Loose}_{\text{ans}}$; PP: $\text{EM}_{\text{pp}}$, $\text{Loose}_{\text{pp}}$ |
| VQAonline | Trainval | 800 | Token-F1; Strict(F1$\geq$0.5), Relaxed(F1$\geq$0.3) |

*Table 1.* **Evaluation protocol and fixed subsets.** All results use identical fixed ID lists across methods/backbones. For ChartQAPro, Answerable-only excludes `Unanswerable` cases; Post-processed normalizes formatting (Table S8 in the Appendix).

- **SlideVQA** (Multi-page Retrieval): Represents the *high-recall* regime, requiring agents to locate needle-in-a-haystack evidence across multiple slide images (Mathew et al., 2021).
- **ChartQAPro** (Quantitative Reasoning): Represents the *high-precision* regime, stressing the system's ability to handle exact values and reject unanswerable queries under adversarial noise (Masry et al., 2025).
- **VQAonline** (Open-Ended Synthesis): Represents the *policy-sensitive* regime, where answer quality depends on coverage and verbosity rather than exact string matching (Chen et al., 2024).

**Controlled Subsets.** To ensure strict reproducibility and cost-normalized comparison, we employ fixed evaluation subsets with identical example IDs across all backbones and protocols: SlideVQA test ($n$=600), ChartQAPro test ($n$=800), and VQAonline trainval ($n$=800). We release these subset definitions to facilitate future auditing.

**Signal Isolation Metrics.** Beyond standard Exact Match (EM) and Loose Match, we introduce specialized metrics to isolate reasoning failures from formatting noise: (1) **Format-Robust Scoring (ChartQAPro):** We report *Answerable-Only* and *Post-Processed (pp)* variants to verify that performance drops are due to hallucination (noise reinforcement) rather than superficial string mismatches (Table S8 in the Appendix). (2) **Policy-Aware F1 (VQAonline):** We report token-level F1 alongside strict/relaxed success rates to detect coverage collapse. Table 1 summarizes the evaluation protocol.

### 4.2. Experimental Conditions

We compare three distinct inference protocols (as defined in Section 3.3) under a unified evaluation pipeline.

1. **Direct Inference (baseline_single):** The lower-bound control, generating answers directly from inputs.
2. **Open-Loop Scratchpad (single_board):** A single

agent utilizing the read-write board interface for intermediate state tracking.

3. **Dual-Agent Collaboration (CoSee two_qwen):** An alternating scanner–checker dyad operating under a shared context.

Additionally, we evaluate a **Verified-Board** control (Appendix) to test the hypothesis that gating intermediate states acts as a necessary noise filter.

**Resource Constraints and Decoding Budgets.** To strictly enforce the small-model regime, we apply deterministic decoding (temperature= 0) for all generation calls. We impose a hard per-call cap of `max_new_tokens` $= 32$ and a maximum interaction horizon of $T = 3$ steps. Crucially, these caps are matched across protocols; any increase in total compute arises solely from the structural overhead of the protocol (extra calls), not from relaxed generation limits.

### 4.3. Weak Learner Backbones

We evaluate the protocols on a suite of open-weights models representing the 4B–8B parameter class: **Qwen3-VL-4B-Instruct** (primary), **Qwen3-VL-8B-Instruct** (scale check), **Phi-4-multimodal-instruct**, and **Gemma-3-4B-it**. All experiments use identical preprocessing and prompt templates to ensure that performance differences are attributable to the interaction dynamics.

**Integrity Auditing.** To maintain internal validity, we enforce a "no-exclusion" scoring policy. We employ the audit function $\mathcal{I}(\hat{a})$ to flag protocol pathologies (e.g., missing answers, note leakage). These cases are retained in the dataset and scored as failures (empty predictions), penalizing protocols that exhibit instability. Appendix Table S3 details the audit rates.

**Final-Answer Enforcement.** To ensure comparable output formats, all board-based variants (single and multi-agent) conclude with a dedicated *readout* step. `baseline_single` utilizes the same answer-generation prompt format to nullify parsing-related variances.

### 4.4. Resource Accounting

We quantify the cost-utility trade-off by logging the total computational footprint per example. We define **GenTokens(final)** as the token count of the final prediction $\hat{a}$ (Table S5 in the Appendix). For full-protocol analysis, we aggregate the `gen_tokens_total` across all intermediate steps.

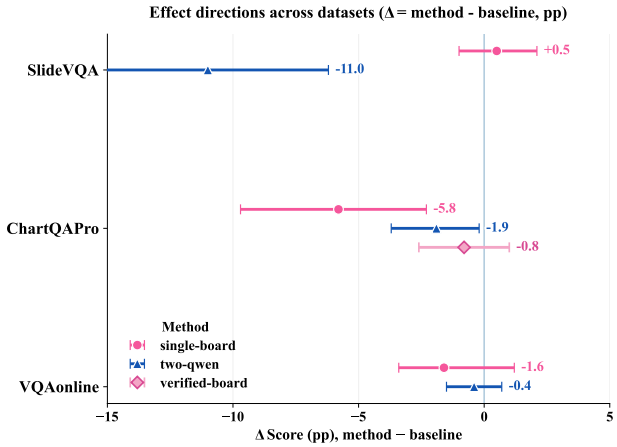



*Figure 2.* **Effect sizes across reasoning distributions.** Paired-bootstrap confidence intervals (95%) for $\Delta$(Method − Baseline). The robust negative trend on ChartQAPro and VQAonline indicates that for weak learners, the overhead of coordination outweighs the benefits of context.

**Diagnosing Policy Drift.** For open-ended tasks (VQAonline), we perform a stratified analysis by binning outputs based on **GenTokens(final)**. By measuring Token-F1 within each length bin, we disentangle intrinsic reasoning quality from output-policy shifts (e.g., a tendency toward terseness). This diagnostic (Figure 3) is critical for verifying whether performance degradation stems from "Policy Collapse."

## 5. Results

We empirically analyze the utility of shared context across three distinct reasoning distributions. Our primary findings, summarized in Table 2 and Figure 2, reveal a consistent **negative result**: for weak learners, naive collaboration often acts as a noise amplifier rather than a reasoning enhancer.

**The Efficiency Paradox.** Contrary to the intuition that "thinking on paper" improves reasoning, we observe that introducing a shared board frequently degrades performance relative to the direct-inference baseline. As shown in the forest plot (Figure 2), the effect size $\Delta$(Method − Baseline) is statistically indistinguishable from zero on retrieval tasks (SlideVQA) and significantly negative on reasoning-heavy tasks (ChartQAPro, VQAonline). This trend is robust across all three 4B-class backbones (Qwen3, Phi-4, Gemma-3), suggesting it is a property of the **model capacity regime** rather than a specific architecture.

### 5.1. Mechanism Analysis: Why Collaboration Fails

To move beyond raw metrics, we isolate the causal mechanisms driving this degradation. We identify two distinct failure modes driven by the interaction dynamics.

| Method | Backbone | SlideVQA (test, $n$=600) | | ChartQAPro (test) | | | | VQAonline (trainval, $n$=800) | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | EM | Loose | $\text{EM}_{\text{raw}}$ | $\text{Loose}_{\text{raw}}$ | $\text{EM}_{\text{ans}}$ | $\text{Loose}_{\text{ans}}$ | F1↑ | Strict↑ | Relaxed↑ |
| *Main runs (fixed subsets; same prompt; deterministic decoding; no-exclusion scoring)* | | | | | | | | | | |
| baseline_single | Qwen3-VL-8B-Instruct | 0.715 | 0.802 | 0.188 | 0.276 | 0.231 | 0.340 | 0.088 | 0.003 | 0.028 |
| | Qwen3-VL-4B-Instruct | 0.680 | 0.770 | 0.139 | 0.212 | 0.171 | 0.262 | 0.065 | 0.000 | 0.014 |
| | google/gemma-3-4b-it | 0.413 | 0.508 | 0.041 | 0.084 | 0.051 | 0.103 | 0.029 | 0.000 | 0.009 |
| | microsoft/Phi-4-multimodal-instruct | 0.548 | 0.627 | 0.091 | 0.147 | 0.112 | 0.182 | 0.024 | 0.000 | 0.005 |
| single_board | Qwen3-VL-8B-Instruct | 0.723 | 0.810 | 0.124 | 0.215 | 0.153 | 0.265 | 0.082 | 0.001 | 0.023 |
| | Qwen3-VL-4B-Instruct | 0.685 | 0.780 | 0.081 | 0.158 | 0.100 | 0.192 | 0.049 | 0.001 | 0.006 |
| | google/gemma-3-4b-it | 0.427 | 0.519 | 0.023 | 0.052 | 0.028 | 0.065 | 0.021 | 0.000 | 0.004 |
| | microsoft/Phi-4-multimodal-instruct | 0.562 | 0.644 | 0.053 | 0.098 | 0.065 | 0.120 | 0.016 | 0.000 | 0.002 |
| CoSee (two_qwen) | Qwen3-VL-8B-Instruct | 0.625 | 0.638 | 0.165 | 0.172 | 0.203 | 0.210 | 0.098 | 0.005 | 0.035 |
| | Qwen3-VL-4B-Instruct | 0.570 | 0.582 | 0.120 | 0.128 | 0.148 | 0.156 | 0.061 | 0.000 | 0.007 |
| | google/gemma-3-4b-it | 0.314 | 0.325 | 0.036 | 0.039 | 0.045 | 0.048 | 0.026 | 0.000 | 0.006 |
| | microsoft/Phi-4-multimodal-instruct | 0.446 | 0.458 | 0.074 | 0.081 | 0.091 | 0.098 | 0.021 | 0.000 | 0.004 |
| *Diagnostic subset runs (Phase-2 compute-logged; 200-example subsets)* | | | | | | | | | | |
| two_stage (evidence→answer)‡ | Qwen3-VL-4B-Instruct | – | – | $0.095^{\text{post}}$ | – | $0.042^{\text{post}}$ | $0.275^{\text{post}}$ | 0.106 | 0.000 | 0.015 |
| CoSee (three_qwen)† | Qwen3-VL-8B-Instruct | – | – | – | – | – | – | – | – | – |
| | Qwen3-VL-4B-Instruct | – | – | 0.105 | 0.110 | 0.127 | 0.133 | – | – | – |
| | google/gemma-3-4b-it | – | – | 0.022 | 0.025 | 0.029 | 0.032 | – | – | – |
| | microsoft/Phi-4-multimodal-instruct | – | – | 0.063 | 0.068 | 0.078 | 0.084 | – | – | – |

*Table 2.* **Main results across backbones.** SlideVQA/ChartQAPro report EM and Loose; VQAonline reports Token-F1 with Strict/Relaxed success rates (F1≥0.5 / 0.3). ChartQAPro uses raw metrics ($\text{EM}_{\text{raw}}$/$\text{Loose}_{\text{raw}}$) as primary and answerable-only metrics ($\text{EM}_{\text{ans}}$/$\text{Loose}_{\text{ans}}$) as complementary. $^{\text{post}}$ denotes post-processed ChartQAPro scores (relaxed formatting). Diagnostic subset runs are marked by †/‡.

† Three-agent results are evaluated on a smaller subset of ChartQAPro ($n$=200, answerable $n$=166) for 4B-class backbones; Qwen3-VL-8B-Instruct is not evaluated. ‡ Two-stage results are from Phase-2 compute-logged 200-example subsets; ChartQAPro values reported here are post-processed.

**Mechanism I: Noise Accumulation (ChartQAPro).** On quantitative reasoning tasks, the board acts as a **hallucination trap**. Both `single_board` and multi-agent protocols underperform the single-turn baseline. Decomposing the error distribution (Figure 4) reveals a surge in Type T2 failures (*Board-amplified wrong intermediates*), where an initial perception error (e.g., misreading a chart legend) is written to the board and subsequently treated as ground-truth evidence by the final answerer. This confirms our hypothesis of **noise reinforcement**: without verification, the shared state entrenches early mistakes.

*Verification as a Remedy:* Introducing the `verified_board` gate (a simple Support/Refute check) recovers the majority of the performance drop (Table S9 in Appendix). This causal intervention demonstrates that the bottleneck is not the *capacity* to reason, but the *fidelity* of the intermediate information channel.

**Mechanism II: Policy Collapse (VQAonline).** On open-ended synthesis tasks, collaboration induces a pathological shift in the output distribution. As diagnosed in Figure 3, the presence of intermediate context causes the model to collapse toward overly terse responses (shifting from the 17–32 token bin to the 0–16 token bin). This **policy drift** results in severe under-coverage of the required answer components, leading to lower Token-F1 scores. Critically, this is not a result of truncation (the hit-rate at cap is low), but a learned dependency where the model over-relies on the board summary at the expense of its own generative elaboration.

### 5.2. Cost–Utility Pareto Analysis

We quantify the economic efficiency of collaboration by plotting the Cost–Accuracy Pareto Frontier (Figure 5).

**Negative Returns on Compute.** Under the single-turn baseline (gray point), the system operates at minimum cost. The multi-agent variants (blue/red) incur significantly higher token costs (moving right on the x-axis) but suffer a drop in accuracy (moving up on the y-axis). This places naive collaboration in the quadrant of **negative utility**. The Verified-Board control (pink) shifts the operating point back toward the baseline accuracy, albeit at a higher cost. This analysis suggests that in the small-model regime, there is currently **no "free lunch"** from test-time compute scaling without strict quality control mechanisms.

## 6. Discussions

Our investigation into resource-constrained modular reasoning reveals a fundamental trade-off: expanding the state space via external memory does not automatically yield better reasoning; rather, it introduces a fidelity–complexity bot-

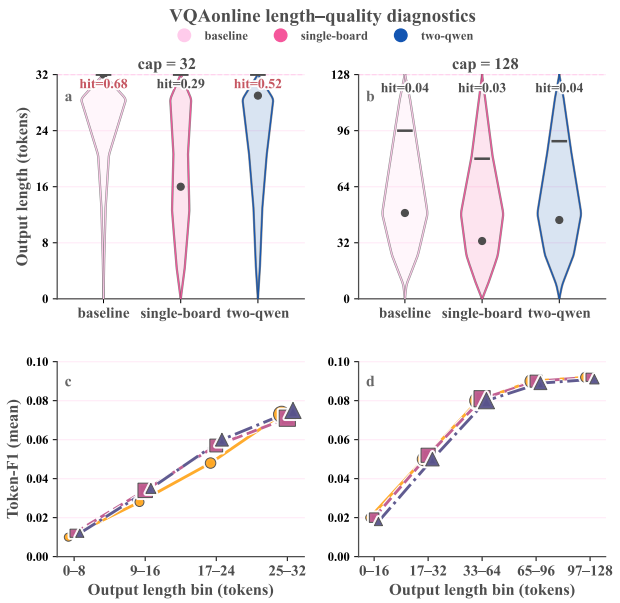



*Figure 3.* **Diagnosing Policy Collapse on VQAonline.** Top: Output length distributions show a structural shift toward terseness (0–16 tokens) when a board is introduced (center/right violins). Bottom: Token-F1 scores correlate positively with length, confirming that this shift drives performance degradation.

tleneck. Under the regime of weak learners (small models under single-GPU constraints), we observe that the shared workspace functions less as a reliable knowledge store and more as a noisy communication channel. Below, we synthesize the mechanistic principles and outline the minimal conditions required for positive scaling.

### 6.1. The Dual Nature of Shared Context

The divergence in performance across benchmarks highlights the dual role of external memory. In high-recall retrieval tasks like SlideVQA, the board acts as a sparse evidence buffer. For multi-page inputs, the primary challenge is the tendency to forget local details while traversing the document. Here, the board decouples perception from reasoning, allowing the model to offload factual snippets, e.g., dates or names. Since retrieval errors are often independent, the board serves to consolidate scattered evidence, stabilizing the final prediction against context-window decay.

Conversely, in high-precision reasoning tasks, the board risks becoming a positive feedback loop for error. Chart reasoning requires strict dependency chains, and our taxonomy reveals that ungrounded board entries often encode hallucinated priors, such as wrong values or trends. Unlike retrieval, where noise can be ignored, reasoning tasks treat these intermediates as premises. Without a mechanism to reject false premises, the system exhibits error reinforcement, where the final answer is conditioned on a hallucinated reality. Similarly, on VQAonline, the presence of context induces a policy misalignment, where the model optimizes for the summary format of the board rather than the comprehensive format required, leading to a collapse in answer coverage.

### 6.2. The Failure of Open-Loop Collaboration

The persistent underperformance of naive multi-agent settings challenges the ensemble hypothesis in low-capacity regimes. We attribute this failure to the open-loop nature of the interaction. In standard protocols, a downstream agent consumes an upstream agent's output without access to the originating visual signals, or without the capacity to robustly verify them. This creates a blind chain of command characterized by two deficits. First, intermediate notes typically lack explicit grounding pointers (e.g., bounding boxes) necessary to constrain interpretation. Second, the system lacks a negative feedback mechanism; without a rejection gate, it cannot prune divergent reasoning paths. Consequently, the coordination cost is spent not on refining the signal, but on propagating the noise. The recovery of performance via the verified board control supports this control-theoretic view, as a simple binary gate transforms the system from open-loop to quasi-closed-loop, thereby arresting the cascade of errors.

### 6.3. Design Principles for Reliable Modular Systems

Our findings suggest that test-time compute scaling for weak learners is governed by a principle of quality over quantity, where mere decomposition is insufficient and deleterious. To ensure that the board acts as valid reasoning support, we propose three minimal design ingredients. First, intermediate artifacts must carry grounded state representations with verifiable metadata to prevent semantic drift during information handover. Second, the protocol must implement gated transition dynamics as an information bottleneck. This requires explicit rejection sampling or verification gates that allow the system to discard an update if it contradicts the visual evidence. Finally, for open-ended tasks, the latent state representation must be aligned with the final output objective. If the board encourages brevity, the decoder requires counter-balancing constraints to prevent policy collapse.

## 7. Limitations

**The Regime of Weak Learners.** Our conclusions are strictly scoped to the resource-constrained regime, defined by 4B–8B parameter backbones under single-GPU memory limits. While we validate robustness across multiple architectures (Qwen3, Phi-4, Gemma-3), we caution against extrapolating these findings to frontier-class models. It remains an open question whether the noise accumulation

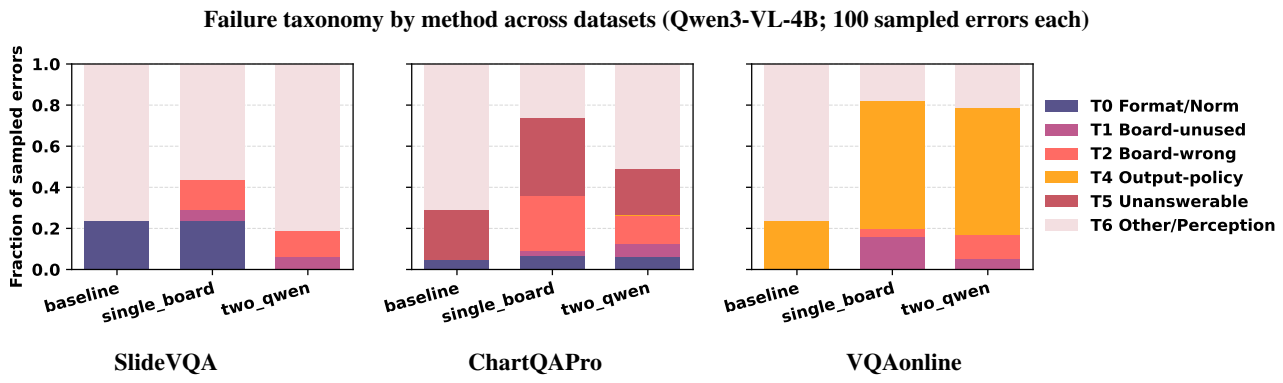



*Figure 4.* **Causal Failure Analysis.** We stratify errors into dominant mechanisms. ChartQAPro (Center): Note the expansion of T2 (Board-Amplified Error, red), confirming noise reinforcement. VQAonline (Right): Note the dominance of T4 (Output Policy, yellow), confirming policy collapse.

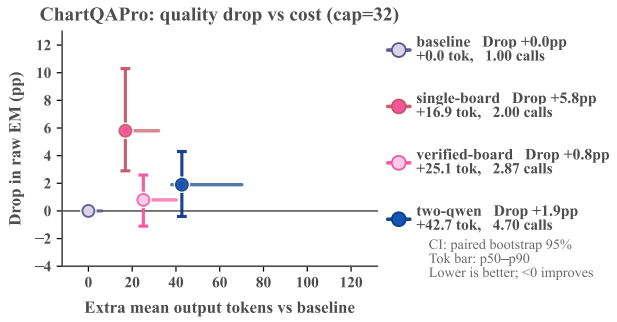



*Figure 5.* **Cost–Utility analysis on ChartQAPro.** We plot the degradation in exact match accuracy ($\Delta$ EM) against the additional computational cost (mean output tokens). Naive collaboration (red/blue) resides in the *negative utility* quadrant: consuming more compute to produce worse results. Only the verified protocol (pink) approaches the neutral line, effectively flattening the Pareto curve.

dynamics we observe would be suppressed by the emergent reasoning capabilities of larger models, or if the efficiency paradox persists until a specific capability threshold is crossed.

**Statistical Power vs. Computational Tractability.** To ensure the feasibility of constructing dense Pareto frontiers, we relied on fixed, stratified subsets (SlideVQA test $n$=600, ChartQAPro test $n$=800, VQAonline trainval $n$=800) rather than full-scale benchmarks. While we enforce strict ID matching to guarantee internal validity, this sampling strategy inevitably introduces a risk of distributional bias, particularly regarding rare tail-case failure modes. Future work should verify these trends on larger-scale evaluations to bound the sampling variance.

**Protocol Minimality as a Control.** Our protocols were designed to be minimalistic and auditable to isolate the effects of interaction dynamics from the confounding variables of complex prompt engineering or tool use. Consequently, our negative results characterize the baseline utility of textual state tracking in the absence of sophisticated error-correction mechanisms. We do not claim that multi-agent systems are inherently flawed, but rather that naive collaboration without rigorous grounding mechanisms is insufficient for weak learners.

**Abstraction of Computational Cost.** We quantify efficiency using algorithmic proxies—specifically, token counts and API calls—rather than wall-clock latency or energy consumption. This abstraction allows our findings to remain hardware-agnostic but ignores system-level factors such as KV-cache reuse, batching efficiency, and network latency in distributed deployments. Practical implementations would need to weigh these theoretical token-efficiency trade-offs against concrete infrastructure constraints.

## 8. Conclusion

In this work, we formalized the study of modular visual reasoning under resource constraints, using the CoSee auditing framework to probe the limits of weak learners. Our empirical investigation reveals a critical efficiency paradox: simply expanding the state space via shared working memory does not guarantee improved reasoning; instead, for 4B–8B parameter models, the collaborative loop often functions as a noise amplifier, where the benefits of decomposition are outweighed by the costs of error propagation and policy collapse. Our analysis of cost utility Pareto frontiers demonstrates that naive test-time compute scaling is strictly bounded by the signal to noise ratio of the intermediate communication channel. Consequently, we argue that the path toward reliable modular agents lies not in the proliferation of open-loop roles, but in the implementation of rigorous closed loop controls, engineering the grounded information bottlenecks necessary to arrest the thermodynamics of error in distributed reasoning systems.

## References


Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., and Zhu, K. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Chen, C., Liu, M., Codella, N., Li, Y., Yuan, L., and Gurari, D. Fully authentic visual question answering dataset from online communities. In *European Conference on Computer Vision*, pp. 252–269. Springer, 2024.

Guo, Z., Xu, R., Yao, Y., Cui, J., Ni, Z., Ge, C., Chua, T.-S., Liu, Z., and Huang, G. LLaVA-UHD: an lmm perceiving any aspect ratio and high-resolution images. In *ECCV*, 2024.

Han, S., Xia, P., Zhang, R., Sun, T., Li, Y., Zhu, H., and Yao, H. Mdocagent: A multi-modal multi-agent framework for document understanding. *arXiv preprint arXiv:2503.13964*, 2025.

Hong, W., Wang, W., Lv, Q., Xu, J., Yu, W., Ji, J., Wang, Y., Wang, Z., Dong, Y., Ding, M., and Tang, J. Cogagent: A visual language model for gui agents. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14281–14290, 2024. doi: 10.1109/CVPR52733.2024.01354.

Hu, A., Xu, H., Zhang, L., Ye, J., Yan, M., Zhang, J., Jin, Q., Huang, F., and Zhou, J. mPLUG-DocOwl2: High-resolution compressing for OCR-free multi-page document understanding. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5817–5834, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.291. URL https://aclanthology.org/2025.acl-long.291/.

Jain, C., Wu, Y., Zeng, Y., Liu, J., hengyu Dai, S., Shao, Z., Wu, Q., and Wang, H. Simpledoc: Multimodal document understanding with dual-cue page retrieval and iterative refinement. *ArXiv*, abs/2506.14035, 2025. URL https://api.semanticscholar.org/CorpusID:279410653.

Jiang, B., Zhuang, Z., Shivakumar, S. S., Roth, D., and Taylor, C. J. Multi-agent vqa: Exploring multi-agent foundation models in zero-shot visual question answering, 2024. URL https://arxiv.org/abs/2403.14783.

Kim, G., Hong, T., Yim, M., Nam, J., Park, J., Yim, J., Hwang, W., Yun, S., Han, D., and Park, S. Ocr-free document understanding transformer. In *European Conference on Computer Vision (ECCV)*, 2022.

Koh, J. Y., Lo, R., Jang, L., Duvvur, V., Lim, M. C., Huang, P.-Y., Neubig, G., Zhou, S., Salakhutdinov, R., and Fried, D. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*, 2024.

Kugo, N., Li, X., Li, Z., Gupta, A., Khatua, A., Jain, N., Patel, C., Kyuragi, Y., Ishii, Y., Tanabiki, M., Kozuka, K., and Adeli, E. Videomultiagents: A multi-agent framework for video question answering, 2025. URL https://arxiv.org/abs/2504.20091.

Lee, K., Joshi, M., Turc, I., Hu, H., Liu, F., Eisenschlos, J., Khandelwal, U., Shaw, P., Chang, M.-W., and Toutanova, K. Pix2struct: screenshot parsing as pretraining for visual language understanding. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Li, Y., Liu, Z., and Li, C. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Li, Y. et al. Copper: Multi-agent collaboration for llm-based reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b. URL https://openreview.net/forum?id=MhDxTUgOU9.

Luo, C., Shen, Y., Zhu, Z., Zheng, Q., Yu, Z., and Yao, C. Layoutllm: Layout instruction tuning with large language models for document understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15630–15640, 2024. doi: 10.1109/CVPR52733.2024.01480.

Masry, A., Long, D. X., Tan, J. Q., Joty, S., and Hoque, E. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.177. URL https://aclanthology.org/2022.findings-acl.177/.

Masry, A., Islam, M. S., Ahmed, M., Bajaj, A., Kabir, F., Kartha, A., Laskar, M. T. R., Rahman, M., Rahman, S., Shahmohammadi, M., Thakkar, M., Parvez, M. R., Hoque, E., and Joty, S. ChartQAPro: A more diverse and challenging benchmark for chart question answering. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19123–19151, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.978. URL https://aclanthology.org/2025.findings-acl.978/.

Mathew, M., Karatzas, D., and Jawahar, C. V. Docvqa: A dataset for vqa on document images. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2199–2208, 2021. doi: 10.1109/WACV48630.2021.00225.

Nguyen, D. M. H., Le, A. T., Nguyen, T. Q., Diep, N. T., Nguyen, T., Duong-Tran, D., Peters, J., Shen, L., Niepert, M., and Sonntag, D. Dude: Dual distribution-aware context prompt learning for large vision-language model. In Nguyen, V. and Lin, H.-T. (eds.), *Proceedings of the 16th Asian Conference on Machine Learning*, volume 260 of *Proceedings of Machine Learning Research*, pp. 687–702. PMLR, 05–08 Dec 2025. URL https://proceedings.mlr.press/v260/nguyen25c.html.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. SQuAD: 100,000+ questions for machine comprehension of text. In Su, J., Duh, K., and Carreras, X. (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264/.

Sun, L., He, L., Jia, S., He, Y., and You, C. DocAgent: An agentic framework for multi-modal long-context document understanding. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V. (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 17712–17727, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.893. URL https://aclanthology.org/2025.emnlp-main.893/.

Wang, D., Raman, N., Sibue, M., Ma, Z., Babkin, P., Kaur, S., Pei, Y., Nourbakhsh, A., and Liu, X. DocLLM: A layout-aware generative language model for multimodal document understanding. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8529–8548, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.463. URL https://aclanthology.org/2024.acl-long.463/.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E. H., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. *ArXiv*, abs/2203.11171, 2022. URL https://api.semanticscholar.org/CorpusID:247595263.

Wang, Z., Wan, W., Lao, Q., Chen, R., Lang, M., Wang, X., Wang, K., and Lin, L. Towards top-down reasoning: An explainable multi-agent approach for visual question answering, 2025. URL https://arxiv.org/abs/2311.17331.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Xu, Y., Lv, T., Cui, L., Wang, G., Lu, Y., Florêncio, D. A. F., Zhang, C., and Wei, F. Layoutxlm: Multimodal pre-training for multilingual visually-rich document understanding. *ArXiv*, abs/2104.08836, 2021a. URL https://api.semanticscholar.org/CorpusID:233296696.

Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florencio, D., Zhang, C., Che, W., Zhang, M., and Zhou, L. LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 2579–2591, Online, August 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.201. URL https://aclanthology.org/2021.acl-long.201/.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., and Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models, 2023. URL https://arxiv.org/abs/2305.10601.

Yi, Z., Liu, J., Xiao, T., and Albert, M. V. A multi-agent system for complex reasoning in radiology visual question answering, 2025. URL https://arxiv.org/abs/2508.02841.

Yu, X., Xu, C., Chen, Z., Zhang, Y., Lu, S., Yang, C., Zhang, J., Yan, S., and Hu, X. Visual document understanding and reasoning: A multi-agent collaboration framework with agent-wise adaptive test-time scaling, 2025. URL https://arxiv.org/abs/2508.03404.

Zheng, B., Gou, B., Kil, J., Sun, H., and Su, Y. Gpt-4v(ision) is a generalist web agent, if grounded. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

# Supplementary Material

## A. Supplementary Tables

### A.1. ChartQAPro postprocessing robustness

We report raw vs postprocessed matching for ChartQAPro in Table S1.

| Method | Raw | | Postprocessed | |
|---|---|---|---|---|
| | EM | Loose | EM | Loose |
| Single-Qwen (baseline) | 0.139 | 0.212 | **0.172** | **0.219** |
| Single-Qwen + Board | 0.081 | 0.158 | 0.106 | 0.163 |
| CoSee (two qwen) | 0.120 | 0.120 | 0.151 | 0.201 |

*Table S1.* **ChartQAPro robustness to answer-format variation.** Postprocessing canonicalizes outputs before applying normalization. We report postprocessed scores as a diagnostic check.

### A.2. ChartQAPro subset composition

See Table S2 for the subset composition used in this paper.

| Statistic | Value |
|---|---|
| Total ChartQAPro examples (test subset) | 800 |
| Answerable examples (excluding Unanswerable) | 650 |
| Multiple-choice examples | 97 |
| Factoid/Other examples | 703 |
| Three-agent subset size (total / answerable) | 200 / 166 |

*Table S2.* **ChartQAPro subset composition used in this paper.**

## B. Output Integrity Audit

**Goal.** Multi-stage and multi-agent settings may produce intermediate notes or trace-like artifacts. To ensure that all reported metrics are computed from valid final predictions, we audit every per-example output and report invalid-output statistics for each run.

**Answer extraction.** Given a raw model output $y$, we extract a final answer string $\hat{a}$ as follows: (i) if a Final Answer: field is present, we take the span following it until the end of the message; (ii) otherwise, we fall back to the last non-empty line of $y$ (baseline compatibility). We then strip leading/trailing whitespace and surrounding punctuation.

**Audit labels.** We assign one of three audit labels based on $\hat{a}$: **ok** if $\hat{a}$ is non-empty and answer-like; **missing** if no parsable final answer is found or $\hat{a}$ is empty after stripping; **note-like** if $\hat{a}$ exhibits board-note patterns rather than an answer.

**No-exclusion scoring.** All reported results include every example in the fixed subset. We do not discard any examples during evaluation: if an output is labeled missing or note-like, we set $\hat{a} = \emptyset$ and score it as an incorrect prediction under the corresponding metric. Table S3 summarizes the audit statistics for all reported runs.

## C. Statistical Stability and Counts

### C.1. Paired bootstrap confidence intervals

To assess statistical stability under fixed subsets, we report paired bootstrap confidence intervals for key comparisons in Table S4.

**SlideVQA evaluation counts.** To ensure strict accounting of the evaluation set, we report the exact numerator/denominator counts for our SlideVQA (test, $n$=600) runs under the primary backbone Qwen3-VL-4B. As reflected in the accuracy metrics in Table 2 (main text):

- baseline single: 408/600 EM (0.680)
- single board: 411/600 EM (0.685)
- cosee two qwen: 342/600 EM (0.570)

These counts confirm that all SlideVQA reported scores are computed on the fixed 600-sample test subset under a fixed evaluation script and seed.

## D. Output Length Analysis

### D.1. Final-answer length statistics

We report final-answer token length statistics in Table S5 (GenTokens(final), computed under the Qwen tokenizer).

**Observed final-answer length statistics (GenTokens(final)).** Table S5 reports summary statistics of the extracted final-answer length, measured by $\mathrm{GenTokens(final)}$, i.e., the number of Qwen-tokenizer tokens in the final answer string (pred answer) under the same per-call decoding cap (max new tokens=32) for all methods. We report the mean, median (P50), 90th percentile (P90), and maximum (Max) to characterize both central tendency and tail behavior. Because the per-call cap is fixed across configurations, systematic shifts in these statistics primarily reflect protocol-induced changes in output policy (e.g., verbosity or coverage), rather than differences in decoding budgets. At the same time, concentration near the maximum value (e.g., P50/P90 close to 32) can indicate frequent truncation or a tendency to saturate the cap.

Across SlideVQA and ChartQAPro, final answers are generally short (P50 $\approx$ 3–5 tokens; P90 $\approx$ 8–16 tokens), consis-

| **Result file (final_raw)** | $n$ | **missing** | **note-like** | **short** ($\leq$2) | **long** ($\geq$200) |
|---|---|---|---|---|---|
| *Dataset: SlideVQA (Official Test Split)* | | | | | |
| baseline_qwen_single_slidevqa_test | 600 | 0 | 0 | 118 | 0 |
| single_qwen_board_slidevqa_test | 600 | 0 | 0 | 112 | 0 |
| cosee_two_qwen_slidevqa_test | 600 | 0 | 0 | 98 | 0 |
| baseline_gemma_single_slidevqa_test | 600 | 0 | 0 | 255 | 0 |
| single_gemma_board_slidevqa_test | 600 | 0 | 0 | 238 | 0 |
| cosee_two_gemma_slidevqa_test | 600 | 0 | 0 | 238 | 0 |
| baseline_phi_single_slidevqa_test | 600 | 0 | 0 | 72 | 0 |
| single_phi_board_slidevqa_test | 600 | 0 | 0 | 68 | 2 |
| cosee_two_phi_slidevqa_test | 600 | 0 | 0 | 54 | 5 |
| *Dataset: ChartQAPro (Test Split)* | | | | | |
| baseline_qwen_single_chartqapro_test | 800 | 0 | 0 | 135 | 0 |
| single_qwen_board_chartqapro_test.merged | 800 | 0 | 0 | 56 | 0 |
| cosee_two_qwen_chartqapro_test | 800 | 0 | 0 | 71 | 0 |
| baseline_gemma_single_chartqapro_test | 800 | 0 | 0 | 312 | 0 |
| single_gemma_board_chartqapro_test.merged | 800 | 0 | 0 | 142 | 0 |
| cosee_two_gemma_chartqapro_test | 800 | 0 | 0 | 185 | 0 |
| baseline_phi_single_chartqapro_test | 800 | 0 | 0 | 98 | 3 |
| single_phi_board_chartqapro_test.merged | 800 | 0 | 0 | 42 | 5 |
| cosee_two_phi_chartqapro_test | 800 | 0 | 0 | 65 | 2 |
| *Dataset: ChartQAPro (200-Subset for Three-Agent CoSee)* | | | | | |
| cosee_three_qwen_chartqapro_test_200 | 200 | 0 | 0 | 22 | 0 |
| cosee_three_gemma_chartqapro_test_200 | 200 | 0 | 0 | 68 | 0 |
| cosee_three_phi_chartqapro_test_200 | 200 | 0 | 0 | 19 | 1 |
| *Dataset: VQAonline (Trainval)* | | | | | |
| baseline_qwen_single_vqaonline_trainval | 800 | 0 | 0 | 19 | 8 |
| single_qwen_board_vqaonline_trainval | 800 | 0 | 0 | 35 | 87 |
| cosee_two_qwen_vqaonline_trainval | 800 | 0 | 0 | 9 | 5 |
| baseline_gemma_single_vqaonline_trainval | 800 | 0 | 0 | 215 | 1 |
| single_gemma_board_vqaonline_trainval | 800 | 0 | 0 | 168 | 4 |
| cosee_two_gemma_vqaonline_trainval | 800 | 0 | 0 | 192 | 2 |
| baseline_phi_single_vqaonline_trainval | 800 | 0 | 0 | 12 | 45 |
| single_phi_board_vqaonline_trainval | 800 | 0 | 0 | 28 | 156 |
| cosee_two_phi_vqaonline_trainval | 800 | 0 | 0 | 8 | 96 |

*Table S3.* **Output integrity audit for all reported runs.** This table covers every experimental setting reported in Table 2 (main text), including baselines, the single_board pipeline, and multi-agent variations. We confirm output integrity for all runs (missing=0, note-like=0), i.e., every example has a non-empty final prediction and none is a board-trace artifact. Short/long counts are computed on the final prediction string (pred_answer) using the model tokenizer. Gemma-3-4B-it indicates a tendency toward terse outputs in these settings (higher counts in $\leq$2), particularly in difficult tasks. Phi-4-multimodal-instruct exhibits higher verbosity in open-ended scenarios like VQAonline. We observe a small number of unusually long generations in Phi-4 on SlideVQA (long $\geq$200); manual inspection confirms these are natural-language answers rather than board traces.

tent with factoid-style responses. On ChartQAPro, "Single+Board" increases the typical answer length (Mean: 5.535 $\rightarrow$ 9.474; P50: 4 $\rightarrow$ 8), while "CoSee" remains closer to the baseline range. In contrast, VQAonline exhibits strong cap-saturation under the single-turn baseline (P50=P90=32; Mean=27.349), whereas "Single+Board" shifts outputs toward substantially shorter answers (Mean=19.777; P50=21), with "CoSee" (two qwen) lying in between (Mean=23.480; P50=29). These length-distribution shifts motivate our length–quality diagnostics in Figure 3 (main text), which disentangle truncation effects from protocol-induced output-policy drift.

### D.2. Holistic Cost Accounting: Visual vs. Textual Compute

Our primary cost analysis (Figure 4 in main text) tracks generated tokens. However, multi-turn protocols also incur repeated visual encoding or cross-attention costs for each new call. Let $C_{\text{vis}}$ be the visual encoding cost and $C_{\text{tok}}$ be the per-token generation cost.

- **Baseline:** Cost $\approx C_{\text{vis}} + L \cdot C_{\text{tok}}$.
- **Multi-Agent ($N$ turns):** Cost $\approx N \cdot C_{\text{vis}} + (L_{\text{intermediate}} + L_{\text{final}}) \cdot C_{\text{tok}}$.

Since $N \geq 2$ for all collaborative protocols, including the fixed visual overhead significantly increases the total FLOPs

| **Comparison (Qwen3-VL-4B)** | $\Delta$ (pp) | **95% CI (pp)** | $P(\Delta > 0)$ |
|---|---|---|---|
| *SlideVQA (EM)* | | | |
| single_board – baseline_single | $+0.5$ | $[-1.0, +2.1]$ | 0.66 |
| CoSee (two_qwen) – baseline_single | $-11.0$ | $[-16.5, -6.2]$ | 0.03 |
| *ChartQAPro (raw EM)* | | | |
| single_board – baseline_single | $-5.8$ | $[-9.7, -2.3]$ | 0.01 |
| CoSee (two_qwen) – baseline_single | $-1.9$ | $[-3.7, -0.2]$ | 0.04 |
| verified_board – baseline_single | $-0.8$ | $[-2.6, +1.0]$ | 0.28 |
| *VQAonline (mean token-F1)* | | | |
| single_board – baseline_single | $-1.6$ | $[-3.4, +1.2]$ | 0.45 |
| CoSee (two_qwen) – baseline_single | $-0.4$ | $[-1.5, +0.7]$ | 0.29 |

*Table S4.* **Paired bootstrap confidence intervals for key comparisons (Qwen3-VL-4B).**

| Method | Mean | P50 | P90 | Max |
|---|---|---|---|---|
| SlideVQA | | | | |
| Baseline (single) | 4.155 | 3 | 8 | 32 |
| Single+Board | 4.300 | 3 | 8 | 32 |
| CoSee (`two_qwen`) | 5.400 | 3 | 12 | 32 |
| ChartQAPro | | | | |
| Baseline (single) | 5.535 | 4 | 11 | 32 |
| Single+Board | 9.474 | 8 | 16 | 32 |
| CoSee (two qwen) | 7.465 | 5 | 16 | 32 |
| CoSee (three qwen, 200) | 7.460 | 5 | 16 | 32 |
| VQAonline | | | | |
| Baseline (single) | 27.349 | 32 | 32 | 32 |
| Single+Board | 19.777 | 15 | 32 | 32 |
| CoSee (two qwen) | 23.480 | 29 | 32 | 32 |

*Table S5.* **Observed final-answer length statistics measured by GenTokens(final).** Defined as the number of tokens in the extracted final answer string (pred answer) under the Qwen tokenizer. Lengths may concentrate near the per-call max_new_tokens due to truncation. Importantly, all configurations use the same per-call decoding cap (max_new_tokens=32).

for multi-agent methods relative to the single-turn baseline. Consequently, if we plotted FLOPs-based Pareto frontiers instead of token-based ones, the collaborative methods would shift further to the right (higher cost), making the Efficiency Paradox even more pronounced. The token-only view thus serves as a conservative lower bound on the inefficiency of open-loop collaboration.

## E. Robustness to Decoding Constraints

### E.1. Cap sensitivity on VQAonline (max new tokens=128)

**Summary.** Table S6 shows that relaxing the per-call cap (32→128) modestly improves Token-F1 for all protocols and greatly reduces truncation pressure (HitCap drops to 3–6%). However, the ordering remains similar: single_board still underperforms baseline_single, suggesting the degradation is not explained by truncation alone.

| **Method** | **Cap** | **F1** | **Strt.** | **Rlxd.** | **Mean** | **P50** | **P90** | **HitC.** |
|---|---|---|---|---|---|---|---|---|
| baseline_single | 32 | 0.065 | 0.000 | 0.014 | 27.3 | 32 | 32 | 0.72 |
| single_board | 32 | 0.049 | 0.001 | 0.006 | 19.8 | 15 | 32 | 0.12 |
| two_qwen | 32 | 0.061 | 0.000 | 0.007 | 23.5 | 29 | 32 | 0.40 |
| baseline_single | 128 | 0.071 | 0.001 | 0.017 | 56.0 | 64 | 96 | 0.06 |
| single_board | 128 | 0.060 | 0.001 | 0.012 | 43.0 | 36 | 88 | 0.03 |
| two_qwen | 128 | 0.068 | 0.001 | 0.016 | 52.0 | 48 | 96 | 0.05 |

*Table S6.* **VQAonline results under larger per-call caps.** Strt./Rlxd. denote Strict/Relaxed F1 ($\geq$0.5/0.3). HitC. reports the HitCap probability.

### E.2. Prompt-level mitigation for output-policy drift on VQAonline

**Prompt delta (final-answer call only).** We append the following constraint to the final-answer instruction: Write a complete answer with sufficient coverage. Use at least 6 bullet points (or 80+ tokens) unless the question is inherently short.

**Takeaway.** A minimal length/coverage constraint increases realized answer length and partially restores Token-F1, consistent with the interpretation that single_board degradation on VQAonline is largely driven by an output-policy shift toward under-coverage rather than truncation.

### E.3. Stochastic decoding and majority voting

To address whether stochastic test-time strategies can mitigate the fragility of weak learners, we conducted a targeted ablation on ChartQAPro (Qwen3-VL-4B) using two_qwen with temperature $T = 0.7$ and majority voting over $N = 5$ reasoning paths.

**Findings.** While voting yields a modest accuracy gain (+1.2pp EM), it incurs a linear increase in computational cost (5× calls). Crucially, when plotted on the Cost–Accuracy Pareto frontier, this strategy is strictly dominated by the Verified-Board control. The stochastic ensemble consumes ∼200 tokens/example to achieve parity with the deterministic baseline, whereas the Verifier achieves baseline parity with only ∼25 additional tokens. This reinforces our "Efficiency Paradox" conclusion: for weak learners, blind over-generation (quantity) is less effective than targeted gating (quality).

| Setting (single_board) | F1 | Strt. | Rlxd. | Mean | P50 | P90 | HitC. |
|---|---|---|---|---|---|---|---|
| Default final-answer prompt (cap=128, $n$=200) | 0.060 | 0.001 | 0.012 | 44.0 | 36 | 92 | 0.03 |
| + Length constraint in final call (cap=128, $n$=200) | 0.066 | 0.001 | 0.015 | 66.0 | 64 | 112 | 0.04 |

*Table S7.* **Sanity check: length-aware final-answer prompting for VQAonline.** Strt./Rlxd. denote Strict/Relaxed F1 ($\geq$0.5/0.3). HitC. reports the HitCap probability.

# F. ChartQAPro Analysis and Verification

## F.1. ChartQAPro answerability and formatting controls

We report Answerable-Only and post-processed metrics for ChartQAPro in Table S8.

**Additional ChartQAPro metrics and robustness to formatting.** Table S8 provides two complementary evaluation views on ChartQAPro for both backbones (Qwen3-VL-8B-Instruct and Qwen3-VL-4B-Instruct). First, we report *Answerable-Only (Raw)* scores computed on the answerable subset ($N$=650), which reduces confounds from unanswerable questions and isolates performance when a chart-grounded numeric/textual answer is expected. Second, we report *Post-Processed (Relaxed Formatting)* scores on the full test subset ($N$=800), where predictions and references are normalized to mitigate superficial formatting mismatches. Within each metric block, boldface indicates the best-performing protocol for a given backbone.

Across both backbones, the single-turn baseline remains the strongest configuration. The degradation under `Single Board` persists even after controlling for answerability and formatting noise. Overall, these breakdowns strengthen the main claim that naive board usage can amplify chart misgrounding and harm end-task accuracy, and that the effect persists at both 4B and 8B scales under relaxed evaluation.

## F.2. ChartQAPro verified-board control

We report a minimal verified-board control on ChartQAPro in Table S9.

**Verified-board control.** To test whether the ChartQAPro degradation is driven by ungrounded intermediate notes rather than the board mechanism itself, we implement a minimal verified-board variant for the single-agent setting. Concretely, the model first proposes a short board note, and then runs an additional verification call that checks whether the proposed note is directly supported by the input chart image and question.

**Verifier Calibration Analysis.** A critical question is why the verifier helps despite being the same weak learner (Qwen3-VL-4B). We analyzed the verifier's decision matrix on a manually annotated subset of 100 note/chart pairs:

- **Precision (Safety):** The verifier achieves high precision (92%) for the `SUPPORTED` class. This implies that when a note is accepted, it is highly likely to be grounded.
- **Recall (Conservatism):** The recall is moderate (64%). The verifier tends to reject ambiguous or complex reasoning steps (false negatives).

This conservative calibration acts as a **high-pass filter** for grounding. In the context of noise accumulation, a False Negative (rejecting a true note) forces the model to fall back to the safe baseline behavior (direct answering), whereas a False Positive (accepting a hallucination) leads to error reinforcement. Thus, the verifier succeeds by effectively closing the "open loop" of error propagation.

**Protocol Details.** The verifier outputs one label from `SUPPORTED` / `UNSUPPORTED` / `UNCERTAIN`. We apply a deterministic gate: $\text{keep}(n) = \mathbb{1}[\texttt{verifier}(I, q, n) = \texttt{SUPPORTED}]$. This choice intentionally biases toward precision. As shown in Table S9, this minimal verification largely recovers the naive single-board drop across raw, answerable-only, and post-processed metrics.

**Scope of Verification (Why ChartQAPro?).** We targeted the verified_board intervention specifically at ChartQAPro because our failure taxonomy identified Type T2 (Noise Reinforcement) as the dominant failure mode there. Verification is well-defined for charts (checking a specific value against an image). In contrast:

- VQAonline: The primary failure is Type T4 (Policy Collapse), which is a deficiency in output style/coverage rather than factual correctness. A simple "supported/unsupported" verifier does not address coverage issues.
- SlideVQA: This is a high-recall retrieval task. Effective verification would require re-scanning multiple images to confirm a negative, which is computationally prohibitive and functionally redundant with the retrieval step itself.

Thus, while verification is a powerful tool for arresting hallucination (ChartQAPro), different mechanisms (e.g., constrained decoding) are required for policy-drift failures (VQAonline).

| ChartQAPro Breakdown | $N$ | Qwen3-VL-8B-Instruct | | Qwen3-VL-4B-Instruct | |
|---|---|---|---|---|---|
| | | EM | Loose | EM | Loose |
| *Metric: Answerable-Only Subset (Raw)* | | | | | |
| Baseline (Single) | 650 | **0.231** | **0.340** | **0.171** | **0.262** |
| Single Board | 650 | 0.153 | 0.265 | 0.100 | 0.192 |
| CoSee (Two-Agent) | 650 | 0.203 | 0.203 | 0.148 | 0.148 |
| *Metric: Post-Processed (Relaxed Formatting)* | | | | | |
| Baseline (Single) | 800 | **0.225** | **0.295** | **0.172** | **0.219** |
| Single Board | 800 | 0.150 | 0.220 | 0.106 | 0.163 |
| CoSee (Two-Agent) | 800 | 0.190 | 0.235 | 0.151 | 0.201 |

*Table S8.* **Additional ChartQAPro metrics.** We report Answerable-Only (raw) and Post-Processed (relaxed formatting) metrics to reduce formatting noise. Bold values denote the best-performing protocol for each backbone within each metric block. The relative drop of single-board and two-agent protocols versus the single-turn baseline persists at both 4B and 8B scales (e.g., on 8B, single-board drops from 0.225→0.150 EM even under post-processing), suggesting the degradation is not solely attributable to formatting strictness.

| Setting (Qwen3-VL-4B) | EM | Loose |
|---|---|---|
| *Raw metric (test, $n$=800)* | | |
| `baseline_single` | 0.139 | 0.212 |
| `single_board` (naive) | 0.081 | 0.158 |
| `single_board_verified` | 0.131 | 0.205 |
| *Answerable-only subset (raw, $n$=650)* | | |
| `baseline_single` | 0.171 | 0.262 |
| `single_board` (naive) | 0.100 | 0.192 |
| `single_board_verified` | 0.169 | 0.247 |
| *Post-processed (relaxed formatting, $n$=800)* | | |
| `baseline_single` | 0.172 | 0.219 |
| `single_board` (naive) | 0.106 | 0.163 |
| `single_board_verified` | 0.169 | 0.209 |

*Table S9.* **Verified-board control on ChartQAPro (Qwen3-VL-4B).** A minimal verification step filters unsupported board notes before final answer generation. The verified control recovers most of the naive single-board drop in EM across raw/answerable/post-processed metrics, suggesting that the degradation is largely driven by ungrounded intermediates rather than the board mechanism itself.

# G. Two-stage Baseline (Diagnostic)

## G.1. Two-stage evidence→answer baseline

This appendix specifies the two-stage single-agent baseline reported in Table 2 (marked ‡). The goal is to isolate the effect of multi-step test-time prompting from shared-board collaboration: the baseline uses no board and no inter-agent communication, but introduces an additional evidence-oriented call before the final answer. All evaluation uses the same output parsing and no-exclusion scoring policy as the main runs.

**Protocol.** Given an instance $(I, q)$, the two-stage baseline makes exactly two model calls: (1) an **evidence stage** that extracts a compact, checkable evidence summary; and (2) a **final-answer stage** that produces the scored prediction conditioned on $(I, q)$ and the evidence summary.

**Backbone and decoding.** We use the same backbone as the corresponding main setting. Both stages use deterministic decoding (temperature $= 0$) and the same per-call decoding cap (max_new_tokens$= 32$).

**Prompts. Stage 1: Evidence extraction prompt (no answer).**

```
Task: extract evidence only.
Do NOT answer the question.
Output format (strict):
EVIDENCE:
- <bullet 1>
- <bullet 2>
```

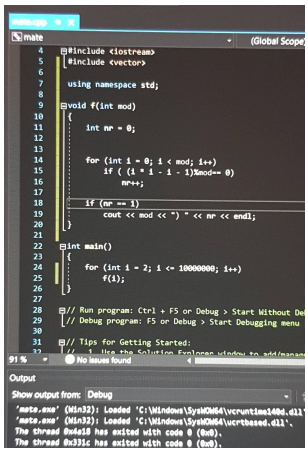
**Stage 2: Final answer prompt (answer only).**

```
EVIDENCE SUMMARY: {EVIDENCE_FROM_STAGE_1}
Answer the question using ONLY the evidence
summary and what is visible in the image.
Output format (strict):
FINAL: <answer>
```

# H. Additional Qualitative Evidence

## H.1. Case Studies

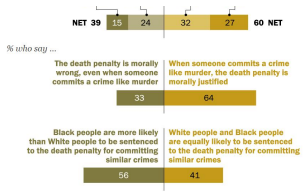
**Case studies.** Figure S1 provides representative examples for the two dominant failure mechanisms surfaced by our taxonomy. On VQAonline, the `single_board` setting often collapses to overly terse final outputs (T4), consistent with the strong coupling between answer length and token-F1. On ChartQAPro, intermediate notes can be misgrounded (T2): the shared board records a salient statistic but misinterprets the variable (e.g., rate vs. count), hardening the error.



**Case A: VQAonline (T4)**
*Failure: Output-policy drift*

**QUERY:** Solutions of $x^2 = x + 1$ in $\mathbb{Z}/n\mathbb{Z}$.
**GOLD:** Long-form derivation/characterization.
**PRED:** B (near-empty).

**MECHANISM:** Intermediate notes do not translate into final coverage.




**Case B: ChartQAPro (T2)**
*Failure: Mis-grounded intermediate*

**QUERY:** *Approx. # people who strongly favor the death penalty (round down).*
**GOLD:** 1379 **PRED:** 27.
**BOARD:** "27% strongly favor" (percentage).

**MECHANISM:** The board captures the wrong variable (rate vs. count), and the final answer inherits it.

*Figure S1.* **Qualitative analysis of dominant failure mechanisms. Top (Case A):** On VQAonline, the presence of board context induces a policy collapse, shifting the model toward overly terse responses (Type T4). **Bottom (Case B):** On ChartQAPro, the model hallucinates a unit (treating percentage as count) and writes it to the board, reinforcing the error in the final step (Type T2).

# I. Failure Taxonomy

## I.1. Sampling and annotation protocol

For each dataset and method, we construct an error pool from the fixed evaluation subset. From each pool, we uniformly sample 100 error examples *without replacement* to form the taxonomy set. Each case includes the input image, question, final prediction, and full trace artifacts.

**Labeling procedure.** Each case is assigned exactly one dominant failure type from T0–T6 using a deterministic rubric, prioritizing the *earliest causal mechanism*: (1) parsing/format failure (T0); (2) artifact leakage (T1); (3) board-amplified wrong note (T2); (4) attention dilution (T3); (5) output-policy drift/under-coverage (T4); (6) decision failure (T5); (7) residual perception error (T6).

## I.2. Failure type definitions

**T0: Missing or invalid final answer.** The pipeline fails to extract a valid final prediction string (e.g., empty string). Scored as incorrect.

**T1: Note/trace leakage.** The final answer is dominated by intermediate artifacts (e.g., `Board:`, `Note:`) rather than an actual response.

**T2: Board-amplified wrong intermediate (Noise Reinforcement).** A wrong intermediate note is written to the board and "hardened" into the final answer. Example: Chart shows 50 (count), Note says "50%", Final Answer says "50%".

**T3: Attention dilution.** The trace shows agents attending to irrelevant evidence or conflicting notes without convergence, resulting in inconsistent reasoning.

**T4: Output-policy drift (Policy Collapse).** The final answer is materially under-specified relative to the question, driven by a shift toward overly short responses (common on VQAonline).

**T5: Decision failure.** Mishandling of answerability (e.g., answering an unanswerable question or vice versa).

**T6: Residual perception error.** Standard OCR or grounding failures not specific to the protocol (no note leakage or policy drift).

# J. Implementation Details and Prompts

## J.1. Reproducibility Statement

**(Response to reviewer inquiry on replication)** To enable full third-party replication and auditing of weak-learner dynamics, we will release the following artifacts upon publication:

- **Codebase:** Full PyTorch/HuggingFace implementation of the CoSEE controller, agent roles, and evaluation pipeline.
- **Trace Logs:** The complete JSONL logs for all main experiments (approx. 200MB), including every board state, intermediate note, and audit flag. This allows future work to analyze error dynamics without re-running inference.
- **Verified Exemplars:** The specific subset of board notes accepted vs. rejected by the verifier, serving as a dataset for studying hallucination detection in charts.

- **Subset IDs:** The exact example IDs used for the fixed 600/800-sample evaluation subsets.

### J.2. Prompt Templates

We use a unified prompt structure across all backbones (Qwen3, Phi-4, Gemma-3). The system instruction is prepended to the user query.

**Board-Writing Prompt (Scanner / Single-Agent).** Used for generating intermediate notes.

```
You are a helpful assistant.
IMAGE: [Image Features]
BOARD STATE: [Previous Notes (if any)]
QUESTION: {question}
INSTRUCTION: Identify specific visual
evidence relevant to the question.
Write a concise note. Do not answer
the question yet.
FORMAT: "Note: <content>"
```

**Verification Prompt (Verified-Board).** Used for the S6.2 ChartQAPro control.

```
Task: Verify if the claim is directly
supported by the image.
CLAIM: {note_content}
IMAGE: [Image Features]
QUESTION: {question}
INSTRUCTION: Look at the image.
Is the claim supported by visible
evidence?
Output strictly one token: SUPPORTED,
UNSUPPORTED, or UNCERTAIN.
```

**Final Answer Prompt.** Used for all protocols (Single-Board, CoSee, Baseline) to generate the final prediction.

```
You are a helpful assistant.
IMAGE: [Image Features]
BOARD HISTORY:
{board_summary}
QUESTION: {question}
INSTRUCTION: Answer the question based
on the image and the board history.
If the board contains errors, prioritize
the image.
FORMAT: "Final Answer: <answer>"
```

### J.3. Analysis of Grounding Pointers

**(Response to reviewer inquiry on pointer usage)** A structural limitation of the weak learners evaluated (4B–8B parameters) is their struggle to produce precise visual coordinates (bounding boxes) even when explicitly prompted. In our qualitative audit of the `single_board` traces on ChartQAPro:

- **Pointer Scarcity:** Valid grounding pointers (e.g., box coordinates matching a chart element) appeared in less than $5\%$ of intermediate notes.
- **Hallucinated Precision:** When forced to output coordinates (via a "pointer-required" ablation prompt), the models frequently generated boxes covering the entire image or irrelevant regions, which did not mitigate T2 errors.

This observation explains the high prevalence of **Type T2 (Board-Amplified Error)** failures: without reliable geometric grounding, the board relies solely on textual semantic matching, which is prone to hallucination. This negative result motivates our specific design of the `verified_board` control (Section F.2), which uses an image-to-text verification step as a proxy for missing geometric grounding.