BURIED IN TEXTUAL DEBT: CONTEXT PRUNING WITH VISUAL EVIDENCE PRESERVATION FOR MLLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal Large Language Models (MLLMs) are increasingly deployed as multi-step agents, where explicit reasoning supports task decomposition and tool use but also accumulates self-generated text. Over long trajectories, this text may draw attention away from visual evidence and anchor later decisions to prior interpretations, creating *textual debt*. Redundant elaboration, mistaken inferences, and outdated assumptions can further impair subsequent reasoning. We propose SPARE, a KL-guided framework for selectively pruning accumulated reasoning while preserving visual evidence. SPARE uses a compact task-state summary as temporary context for diagnosis and the next reasoning step. It replays historical reasoning with and without summary conditioning and measures token-level residual sensitivity using reverse KL. This signal guides segment selection, with low-sensitivity reasoning removed and concise visual evidence extracted from high-sensitivity segments through programmatic rules. The summary is not stored in the persistent trajectory, and tool calls, observations, and images remain unchanged. We further fine-tune the summarizer with supervised fine-tuning, improving the accuracy–pruning trade-off under the same pruning rule. Across three benchmarks and three MLLM backbones, SPARE achieves 50.34% aggregate accuracy compared with 48.81% for Full Trace while pruning 57.81% of reasoning tokens, extending the empirical Pareto frontier of the evaluated baselines. Controlled diagnostics and attention analysis further support the view that reducing textual debt can limit reliance on stale reasoning and encourage greater use of visual evidence.

1 INTRODUCTION

Multimodal Large Language Models (MLLMs) are increasingly deployed as agents that reason, call tools, inspect intermediate observations, and revise their answers over time (Li et al., 2026c). Explicit reasoning supports this process by helping agents decompose visual tasks, coordinate multi-step tool use, and maintain progress across turns (Ke et al., 2026). However, each interaction also adds self-generated plans, descriptions, and assumptions to the working context. Over long trajectories, this accumulated text may increasingly draw the model's attention away from the original image and newly returned tool observations, anchoring later decisions to its own prior interpretations.

We refer to this growing burden as *textual debt*, which includes not only redundant elaboration but also mistaken inferences, tentative guesses, and outdated conclusions that may impair subsequent decisions. Figure 1(a) illustrates how continued textual elaboration can overshadow visual evidence returned by the tools and reinforce an earlier hypothesis. This raises a central question: *when and how should an agent prune accumulated reasoning without losing useful visual evidence?*

Existing approaches offer only partial answers to this question. Visual-token compression methods reduce redundancy in the *visual* stream by dropping or merging image tokens (Chen et al., 2024; Yang et al., 2026; Shang et al., 2026; Zhang et al., 2025b; Tan et al., 2025). While effective for reducing computation, it targets a modality that is often already under-attended in deep layers, and further compression can weaken the relative contribution of image evidence (Takezoe et al., 2026). Concise-reasoning methods primarily shorten newly generated rationales, while generic summarization compresses history at a coarse level. Neither identifies which historical segments

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



Figure 1: Overview of SPARE. (a) In the full reasoning trace, accumulated self-generated text reinforces a stale hypothesis and obscures visual evidence returned by the tools, leading to an incorrect answer. (b) SPARE generates a compact task-state summary and replays historical reasoning with and without summary conditioning. Normalized reverse-KL scores measure token-level residual sensitivity. A count-based rule selects high-sensitivity segments for reconstruction as concise visual evidence and removes low-sensitivity segments. Existing tool calls, observations, and images remain unchanged. The summary temporarily guides the next reasoning step and is not stored in the trajectory.

remain functionally necessary. In multi-step agents, the more pressing source of redundancy is the self-generated text that accumulates across rounds.

We therefore reframe context compression as a light-weight, *evidence-preserving selective forgetting* problem and instantiate it as **SPARE** (Selective Pruning of Accumulated Reasoning with Visual Evidence Preservation), a KL-guided reasoning-pruning framework for multimodal tool-use agents. The key idea is to use a compact task-state summary as a reference for measuring the residual sensitivity of historical reasoning. SPARE quantifies this sensitivity through reverse KL between predictions obtained by replaying the same reasoning with and without summary conditioning. The resulting signal guides selective forgetting, with low-sensitivity segments removed and high-sensitivity segments reconstructed as concise visual evidence. The summary also guides the next reasoning step without entering the persistent trajectory. This procedure requires neither an external verifier nor a separate reward model. We further fine-tune a summarizer with supervised fine-tuning (SFT) to better consolidate task progress and visual information into compact task-state summaries.

Experiments span four multi-step visual tool-use benchmarks and three MLLM backbones. In the aggregate accuracy–pruning analysis on BLINK-Jigsaw, GTA, and MNMS, SPARE achieves 50.34% accuracy while pruning 57.81% of reasoning tokens, extending the Pareto frontier of the evaluated baselines. On a controlled subset with conflicting textual and visual or tool evidence, SPARE reduces stale-answer copying and improves forced-choice accuracy by 27.27 percentage points for both Qwen3-VL backbones, with only the retained reasoning history changed (Appendix D.1). A separate attention analysis finds increased attention to image tokens after pruning, providing complementary support for the view that reducing textual debt can limit reliance on stale reasoning and encourage greater use of visual evidence. Our contributions are summarized as follows:

- We introduce **SPARE**, a reasoning-pruning framework that uses summary-conditioned residual sensitivity to guide selective deletion and visual evidence extraction in multimodal tool-use agents.
- We train the summarizer through supervised fine-tuning to produce compact task-state summaries, further improving the accuracy–pruning trade-off under the same pruning rule.
- Experiments across multiple benchmarks and backbones evaluate the effectiveness of SPARE, while controlled diagnostics and attention analysis provide complementary evidence of reduced reliance on stale reasoning and increased attention to visual inputs.

2 Related Work

Token Pruning for MLLMs Existing MLLM token pruning methods primarily remove redundancy from the visual stream (Chen et al., 2024; Yang et al., 2026; Shang et al., 2026; Tan et al., 2025; Zhang et al., 2025a; Lee et al., 2025). While effective for efficiency, this objective can be misaligned with multimodal reasoning, where questions are often partially answerable from textual cues and visual evidence is already under-attended in deep layers (Zhang et al., 2026b; 2025a; 2026a; Takezoe et al., 2026). Rather than compressing the visual stream, SPARE targets accumulated textual reasoning while keeping the original visual inputs unchanged.

Reasoning Compression and Agent Memory. Reasoning compression methods reduce the length of intermediate reasoning while retaining task performance. TokenSkip (Xia et al., 2025) trains models on compressed reasoning traces to selectively skip less important tokens. Step-entropy (Li et al., 2026b) pruning ranks reasoning steps by predictive entropy and uses compressed traces to train more concise reasoning policies. For multi-step agents, HiAgent (Hu et al., 2024) organizes working memory around subgoals and replaces completed interaction histories with summarized observations. These approaches motivate compression at different levels, from individual reasoning traces to agent memory. SPARE focuses on accumulated reasoning in multimodal tool-use trajectories, combining summary-conditioned residual sensitivity with rule-based visual evidence extraction while leaving tool calls, observations, and images unchanged.

Visual Evidence in Multimodal Reasoning Multimodal reasoning requires both extracting relevant visual information and using it rather than relying primarily on linguistic priors. Prior work addresses language-prior bias through counterfactual training (Niu et al., 2021), contrastive decoding (Zhao et al., 2025), attention calibration (Woo et al., 2025; Fazli et al., 2025; Zhao et al., 2026a), and preference optimization (Xie et al., 2024; Chaubey et al., 2026), but leaves the competing textual context intact. A complementary line strengthens perception itself. Thinking-with-images methods actively manipulate or construct intermediate images during reasoning (Zhou et al., 2024; Su et al., 2025), while Vision-OPD and Visual-OPSD distill signals from privileged visual evidence or visual thoughts (Yuan et al., 2026; Li et al., 2026a). SPARE complements these approaches by managing the textual context available during inference. It uses summary-conditioned policy divergence to guide reasoning pruning and retains concise visual evidence through rule-based extraction, without modifying the original images or tool observations.

3 Preliminaries

3.1 Multi-Step Tool Use in MLLMs

We consider a vision language model (VLM) π_θ that answers a user question through multi-step, tool-augmented reasoning. The input consists of a question q and a set of image tokens $\mathcal{I}$, and the model may query a fixed set of external tools $\mathcal{T}$ over at most T interaction steps.

At step t, the model conditions on the current message history $\mathbf{H}_t$ and produces an assistant message

$$m_t = (h_t, a_t), \tag{1}$$

where h_t is a free-form reasoning span and a_t is either a tool invocation $\langle$tool_call$\rangle$ or a final answer $\langle$response$\rangle$. If a_t invokes a tool, the corresponding tool result is appended to the history, yielding $\mathbf{H}_{t+1}$, and the interaction proceeds to the next step. If a_t emits a final response, the episode terminates. We denote the sequence of past reasoning-action segments by $\mathcal{S}_t = \{\, s_i = (h_i, a_i) \,\}_{i<t}$.

3.2 On-policy Self-Distillation

Knowledge distillation trains a student model π_θ to match a teacher distribution π_{teacher} over next tokens. Conventional off-policy distillation applies this objective on fixed teacher or ground-truth sequences, which creates exposure bias because the training prefixes differ from the student-generated prefixes encountered at inference (Agarwal et al., 2024). *On-policy* distillation reduces this mismatch by applying the distillation loss to trajectories sampled from the student itself, so that the model is supervised on its own inference-time states (Agarwal et al., 2024).

In the on-policy self-distillation (OPSD) instantiation, the teacher and student share the same model, so no external teacher is required (Zhao et al., 2026b). Given a multimodal input $(q, \mathcal{I})$, we first sample a response $y \sim \pi_\theta(\cdot \mid q, \mathcal{I})$ and then align the student's per-token distribution with a teacher distribution defined over the same model:

$$\mathcal{L}_{\text{OPSD}}(\theta) = \mathbb{E}_{(q,\mathcal{I})}\, \mathbb{E}_{y\sim\pi_\theta(\cdot|q,\mathcal{I})} \left[\frac{1}{L} \sum_{k=1}^{L} D_{\text{KL}}\Big(\pi_{\text{teacher}}(\cdot \mid y_{<k}, q, \mathcal{I}) \,\Big\|\, \pi_\theta(\cdot \mid y_{<k}, q, \mathcal{I}) \Big) \right]. \quad (2)$$

Here the gradient is taken with respect to the student parameters θ, while the teacher distribution is treated as a fixed target. Sampling y from π_θ makes the objective on-policy, and defining π_{teacher} from the same model makes it self-distillation. Our method adopts the OPSD perspective for diagnostic comparison. Our method draws on this distributional comparison as a diagnostic rather than a training objective. We replay historical reasoning with and without summary conditioning and use the resulting distributional shift as a residual-sensitivity signal for pruning. No distillation loss is optimized during this procedure.

4 Method

We propose **SPARE** (Selective Pruning of Accumulated Reasoning with Visual Evidence Preservation), a post-hoc context-pruning method for multi-step MLLM agents, as shown in Figure 1. The motivation is that explicit reasoning is useful for decomposing visual questions, planning tool calls, and integrating intermediate observations, but retaining the entire self-generated textual trace indefinitely can create *textual debt*: the agent's later predictions become increasingly conditioned on its own textual history rather than on the visual evidence underlying the task. SPARE therefore does not aim to suppress reasoning generation. Instead, it uses residual sensitivity to determine how historical reasoning is compressed. Low-sensitivity segments are removed, while high-sensitivity segments are reconstructed as concise visual evidence. Existing tool calls, external observations, and image tokens are preserved.

4.1 Summary-Conditioned Residual Sensitivity

Adaptive summary trigger. SPARE is invoked only after the student agent itself calls an internal `summarize_the_task` tool and produces a compact task-state summary σ. This self-generated summary contains the original question, completed tool calls, accumulated visual evidence, and remaining uncertainty. We use σ to compute the residual-sensitivity signal and to temporarily guide the next reasoning step. This adaptive trigger avoids fixed-interval compression: short trajectories incur no pruning cost, and longer trajectories are considered for pruning only after the student has explicitly produced a compact state.

Summary-conditioned replay. Let $\boldsymbol{x} = (x_1, \ldots, x_N)$ be the tokenized concatenation of candidate reasoning segments, separated by a fixed delimiter. To measure sensitivity to summary conditioning, we replay $\boldsymbol{x}$ under two contexts:

$$\rho_{\text{stu}} = \rho_0, \quad (3)$$

$$\rho_{\text{tea}} = \rho_0 \,\|\, \sigma, \quad (4)$$

where ρ_0 contains the shared system, tool-use, and task prefix. The teacher context receives σ as privileged information, while the student context does not. Importantly, this does not introduce an external teacher model: the teacher distribution is produced by the same model, conditioned on the

self-generated summary. The same model π_θ is then queried under both contexts:

$$p_k^{\mathrm{stu}}(u) = P_{\pi_\theta}(u \mid \rho_{\mathrm{stu}}, \boldsymbol{x}_{<k}), \tag{5}$$

$$p_k^{\mathrm{tea}}(u) = P_{\pi_\theta}(u \mid \rho_{\mathrm{tea}}, \boldsymbol{x}_{<k}), \tag{6}$$

where k indexes replay tokens and u indexes vocabulary tokens. A small distributional shift indicates limited sensitivity to the added summary, whereas a larger shift identifies positions whose predictions depend more strongly on summary conditioning. We use this signal to guide selective pruning and evidence preservation.

Top-K reverse-KL sensitivity score. For each replay token position k, we compute a truncated reverse-KL score over the student's top-K vocabulary support Ω_k^{stu}:

$$d_k = \left[\sum_{u \in \Omega_k^{\mathrm{stu}}} p_k^{\mathrm{stu}}(u) \left(\log p_k^{\mathrm{stu}}(u) - \log p_k^{\mathrm{tea}}(u) \right) \right]. \tag{7}$$

We use $K = 20$ in all experiments. If $u \in \Omega_k^{\mathrm{stu}}$ is absent from the teacher top-K support, $\log p_k^{\mathrm{tea}}(u)$ is set to -20. The resulting score d_k quantifies the distributional shift induced by adding the summary at replay position k. We use these token-level scores to guide segment selection.

KL scales vary across pruning events, we normalize scores within each event:

$$\tilde{d}_k = \frac{d_k - d_{\min}}{d_{\max} - d_{\min} + \epsilon}, d_{\min} = \min_{k'} d_{k'}, d_{\max} = \max_{k'} d_{k'}. \tag{8}$$

We set $\epsilon = 10^{-12}$ and assign all normalized scores to zero when $d_{\max} = d_{\min}$. Segment-level pruning aggregates $\tilde{d}_k$ to select segments for deletion or evidence reconstruction. We interpret this score as a residual-sensitivity heuristic. Appendices D.4 and D.5 analyze its relationship to summary coverage and discuss its limitations.

4.2 Segment-Level Pruning

From token scores to segment decisions. The replay sequence $\boldsymbol{x}$ is formed by concatenating historical assistant reasoning segments. Let h_i be the i-th candidate segment and

$$\mathcal{P}_i = \{k : x_k \text{ belongs to } h_i\} \tag{9}$$

denote its token positions in the replay sequence. Since segment boundaries are recorded during concatenation, the normalized token-level scores $\tilde{d}_k$ can be mapped back to their original reasoning segments. A segment is selected for evidence preservation if it contains at least κ high-sensitivity tokens:

$$\mathrm{Preserve}(h_i) = \mathbf{1}\left\{ \left| \left\{ k \in \mathcal{P}_i : \tilde{d}_k > \tau \right\} \right| \geq \kappa \right\}. \tag{10}$$

We refer to segments satisfying this rule as high-sensitivity segments and route them to evidence-preserving reconstruction. The remaining segments are treated as low-sensitivity segments and their reasoning text is removed.

Why a count-based threshold? We use a count rule rather than an average score because task-critical evidence is often sparse. A long reasoning segment may be mostly redundant while still containing a few crucial tokens, such as an OCR string, coordinate, bounding box, or candidate label. Averaging can dilute these localized signals, whereas the count rule identifies segments containing sparse but potentially important evidence.

4.3 Visual Evidence Preservation

KL-driven reconstruction. The segment-level decision determines how each reasoning span is compressed. For high-KL segments, SPARE replaces the original lengthy reasoning with concise, verifiable visual evidence, such as crop locations, recognized text, numeric values, or relative relations among candidates. For low-KL segments, the reasoning text is removed entirely. In both cases, original tool actions, tool outputs, visual observations, and image tokens remain unchanged.

Role of the task-state summary. The task-state summary serves both as diagnostic context for KL estimation and as temporary context for the next reasoning step. It is not appended to the persistent trajectory. The resulting reasoning and action are recorded as usual, while the summary itself is omitted from subsequent contexts.

4.4 Strengthening the Summarizer via SFT

The task-state summary provides a consolidated reference for residual-sensitivity estimation and supports the next reasoning step after pruning. Its quality therefore matters both for selecting historical content and for continuing inference with a compressed history. We further fine-tune the summarizer via supervised fine-tuning (SFT) to consolidate task progress, visual evidence, and remaining uncertainty into a compact state. A more informative summary can support subsequent reasoning with less reliance on the original textual history, motivating more aggressive compression under SPARE. We retain the same KL-based selection rule and evidence reconstruction procedure and examine how improved summarization affects pruning and task performance.

5 Experiments

5.1 Experimental Settings

Benchmarks. We evaluate SPARE on four benchmarks spanning visual reasoning, tool use, and multimodal planning. These settings are related to the broader paradigm of *thinking with images*, in which intermediate visual operations support reasoning (Su et al., 2025). **GTA** (Wang et al., 2024) provides real-world tasks with human-written queries and executable tool chains across perception, operation, logic, and creativity. **V*** (Wu and Xie, 2023) evaluates guided visual search on high-resolution images, requiring fine-grained target localization before answering. **BLINK-Jigsaw** (B-Jig.) (Fu et al., 2024) tests spatial perception by requiring models to reassemble image fragments. **m&m's** (MNMS) (Ma et al., 2024) includes over 4,000 multi-step multimodal tasks involving 33 tools, covering multimodal models, public APIs, and image-processing modules. We use its planning-only setting, which evaluates tool-plan generation without executing the tools. Direct Answer is therefore marked as inapplicable, as it does not produce a tool plan.

Backbones. We evaluate SPARE with Qwen3-VL-8B-Instruct, Qwen3-VL-30B-A3B-Instruct, and Llama-3.1-Nemotron-Nano-VL-8B-V1. For each backbone, all compared methods use the same decoding parameters, generation limits, and tool interface.

Compared methods. For each backbone, our main comparison includes five inference strategies under an identical tool-use harness. (i) **Tool Baseline (Full Trace)** executes the standard multi-round tool loop and retains the complete reasoning-action-observation history. (ii) **- Tools (Direct Answer)** answers the query directly without invoking tools and serves as a non-agentic reference. (iii) **+ Delete All Reasoning** removes every completed reasoning span while preserving the original action blocks, tool observations, and images. (iv) **+ Visual Evidence-Only** non-selectively rewrites every completed reasoning span as compact structured visual evidence while preserving its original action block. (v) **+ SPARE (Ours)** uses summary-conditioned KL to estimate the residual sensitivity of each reasoning span, removing low-residue reasoning and reconstructing high-residue reasoning as structured visual evidence. Full Trace provides the complete-context reference, while Delete All Reasoning and Visual Evidence-Only isolate the effects of non-selective deletion and evidence reconstruction, respectively. Additional random and KL-guided controls used for the accuracy-pruning analysis are defined in Appendix C.3.

Metrics. Following each benchmark's official protocol, we report task success or accuracy, where higher is better. We also report *Pruned (%)*, the net fraction of original reasoning tokens removed from each method's own generated trajectories. Retained tokens include both unmodified reasoning and reconstructed visual evidence in the persistent history (see details in Appendix C). Temporary summaries, tool calls, tool observations, image tokens, and final answers are excluded from this metric. Pruning rates are computed from aggregate token counts within each benchmark. Cross-benchmark accuracy and pruning rates are then weighted by the number of evaluated examples. Full

Table 1: Task performance and average reasoning pruning across four benchmarks. Benchmark columns report accuracy. Acc. and Pr. denote sample-weighted average accuracy and reasoning-pruning rate. MNMS is inapplicable to Direct Answer. The highest and second-highest accuracies are bold and underlined, respectively.

Model / Metric	**GTA** ↑	**V*** ↑	**B-Jig.** ↑	**MNMS** ↑	**Acc.** ↑	**Pr.** %
Qwen3-VL-30B-A3B-Instruct						
Tool Baseline (Full Trace)	45.76	**74.87**	67.33	36.84	56.53	0.00
− Tools (Direct Answer)	33.90	52.36	**70.67**	–	56.50	0.00
+ Delete All Reasoning	50.85	59.69	57.33	24.56	45.54	96.43
+ Visual Evidence-Only	**59.32**	61.78	64.00	25.44	48.89	69.67
+ **SPARE (ours)**	**59.32**	67.02	64.00	**49.56**	**59.24**	69.40
Qwen3-VL-8B-Instruct						
Tool Baseline (Full Trace)	42.37	69.63	**74.00**	**60.96**	64.97	0.00
− Tools (Direct Answer)	42.37	60.21	72.00	–	62.00	0.00
+ Delete All Reasoning	40.68	69.63	62.00	57.02	60.51	96.35
+ Visual Evidence-Only	44.07	69.63	59.33	56.58	60.03	63.38
+ **SPARE (ours)**	**49.15**	**74.35**	72.00	57.46	**65.29**	61.20
Llama-3.1-Nemotron-Nano-VL-8B-V1						
Tool Baseline (Full Trace)	40.68	38.22	45.33	**26.75**	**35.99**	0.00
− Tools (Direct Answer)	**44.07**	**56.02**	**51.33**	–	52.50	0.00
+ Delete All Reasoning	37.29	43.46	46.67	21.49	35.67	91.00
+ Visual Evidence-Only	37.29	48.17	44.00	17.98	35.19	47.63
+ **SPARE (ours)**	42.37	38.74	48.67	21.93	35.35	41.34

Trace has zero pruning. Direct Answer is assigned zero for reporting consistency since it produces no comparable tool-use reasoning history.

Implementation details. SPARE is applied purely at test time. Pruning is triggered only when the model invokes the internal `summarize_the_task` tool, so short trajectories without summaries incur no compression cost. We compute the truncated reverse-KL sensitivity score on the student's top-K support and map token scores to reasoning segments using the count-based rule with fixed $\tau = 0.2$ and $\kappa = 2$ across all models and benchmarks (Section 4). The same model provides both student and teacher distributions by conditioning on prompts with and without the task-state summary σ, avoiding any auxiliary model.

5.2 Main Results

Table 1 reports task accuracy and reasoning pruning across four benchmarks. SPARE achieves the highest average accuracy among the compared pruning methods on both Qwen backbones. On Qwen3-VL-30B-A3B, SPARE improves accuracy from 45.76% to 59.32% on GTA and from 36.84% to 49.56% on MNMS, yielding gains of 13.56 and 12.72 percentage points, respectively. Across the four benchmarks, it increases average accuracy from 56.53% to 59.24% while pruning 69.40% of reasoning tokens.

Nemotron shows a different pattern. Direct Answer outperforms Full Trace on all applicable benchmarks, suggesting that this backbone benefits less from tool use and may be more susceptible to interference from accumulated interaction context. Even so, SPARE improves over Full Trace on GTA, V*, and BLINK-Jigsaw and nearly matches its average accuracy (35.35% vs. 35.99%) while pruning 41.34% of reasoning.

Accuracy–pruning trade-off. Figure 2 compares the aggregate accuracy–pruning trade-off across three models on BLINK-Jigsaw, GTA, and MNMS. Indiscriminate pruning substantially reduces accuracy. Aggregate accuracy drops from 48.81% with Full Trace to 41.95% with Random-33 and remains near 42% under more aggressive random or complete deletion. KL-guided pruning recovers 48.36% accuracy but removes only 16.40% of reasoning, while Visual Evidence-Only achieves

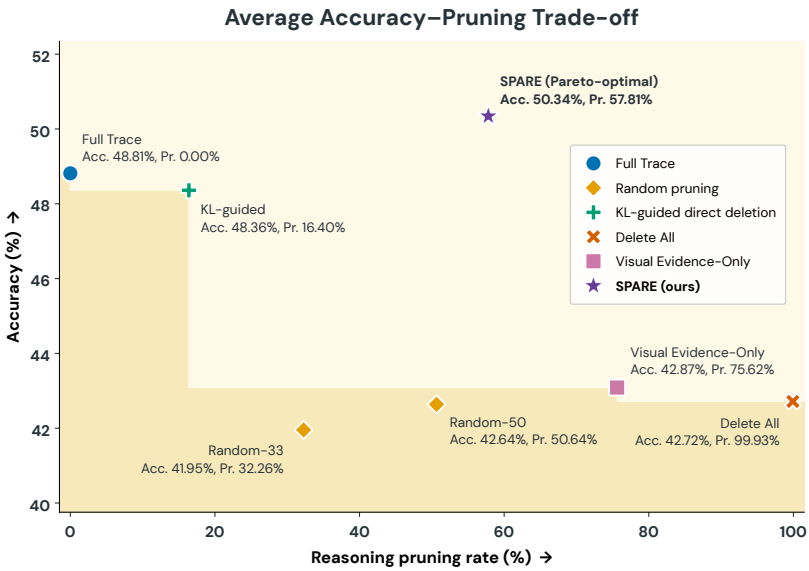


Figure 2: Accuracy–pruning trade-off averaged across three models on BLINK-Jigsaw, GTA, and MNMS. Accuracy and reasoning-pruning rates are weighted by the number of evaluation samples. The staircase connects the Pareto-optimal comparison methods, and the upper-right region indicates a better trade-off. SPARE lies above this frontier, attaining 50.34% accuracy while pruning 57.81% of reasoning tokens.

42.87% accuracy with 75.62% pruning. In contrast, SPARE achieves the highest accuracy of 50.34% while pruning 57.81%, placing it above the baseline Pareto frontier. These results show that KL-based selection identifies which reasoning remains important, while evidence reconstruction preserves that information compactly. Complete per-model results are reported in Table 4 in Appendix C.3.

Textual interference and visual attention. We examine the effects of pruning through three complementary analyses. First, a controlled diagnostic on a constructed GTA subset holds the question, images, and tool interactions fixed while varying the retained reasoning history (Appendix D.1). When early reasoning conflicts with later evidence, SPARE improves forced-choice accuracy by 27.27 pp and reduces stale-answer copying by the same margin for both Qwen3-VL backbones. Random deletion yields smaller gains. Second, we measure text-to-image attention across decoder layers with the metric defined in Appendix B. Averaged over 10 tool-use trajectories, attention to image tokens increases across nearly all decoder layers after pruning (Appendix D.2). Finally, a qualitative case study on visual counting illustrates how pruning is followed by targeted cropping and a correct answer, whereas the unpruned agent continues elaborating on weak evidence (Appendix D.3). Together, these analyses are consistent with reduced textual interference and renewed use of visual evidence.

5.3 SFT for Summarization Capability

We further investigate whether stronger task-state summarization allows SPARE to prune more aggressively. Using 376 on-policy trajectories collected from Qwen3-VL-8B on VTC-Bench (VTC) (Zhu et al., 2026), we prompt Qwen3-235B to generate compact task-state summaries for intermediate contexts. We then fine-tune the 8B agent on the resulting (context, summary) pairs, distilling the stronger model's summarization capability into the agent. Table 2 compares task accuracy under SPARE before and after this fine-tuning. Although the agent's summarization capability is trained on VTC, evaluation on the out-of-distribution benchmarks V* shows that the

Table 2: SFT-based summarization results on an OOD benchmark.

Model / Metric	V* Acc.	Pr.
Base (Qwen3-VL-8B)	69.63	—
+ SPARE	74.35	70.20
+ **SPARE with SFT summarizer (ours)**	**77.49**	**78.84**

stronger summarizer enables SPARE to prune more tokens while achieving higher task accuracy. These results suggest that stronger task-state summaries expand the safe-to-prune region and further demonstrate the effectiveness of our method beyond the training benchmark.

5.4 Ablation Study

Why combine selective pruning with evidence reconstruction? We conduct ablations using Llama-3.1-Nemotron-Nano-VL-8B-V1 on GTA, as shown in Table 3. *KL selection* removes low-sensitivity segments while retaining high-sensitivity segments verbatim, achieving 38.98% accuracy with 38.07% pruning of reasoning tokens. *Evidence-only reconstruction* instead rewrites all reasoning segments as structured visual evidence. It achieves 56.08% pruning but reduces accuracy from 40.68% with Full Trace to 37.29%. Combining selection and reconstruction achieves 42.37% accuracy with 62.42% pruning. Compared with evidence-only reconstruction, the complete method improves accuracy by 5.08 percentage points and increases pruning by 6.34 percentage points. These results support the complementary roles of selective deletion and evidence reconstruction.

Table 3: Ablation study on GTA benchmark.

Ablation Setting	**Acc.**	**Pr.**
Selective Pruning and Evidence Reconstruction		
Full trace baseline	40.68	0.00
Summary probe only	35.59	0.00
+ KL selection	38.98	38.07
+ Evidence	37.29	56.08
+ KL selection + Evidence (ours)	**42.37**	**62.42**
Adaptive Summary Triggering		
Every round	32.20	52.54
Before final answer	38.98	51.38
Model-selected (ours)	**42.37**	**62.42**
Parameter Robustness ($\tau = 0.2$)		
$\kappa = 1$	**42.37**	62.42
$\kappa = 2$ (default)	**42.37**	62.42
$\kappa = 3$	38.98	**63.16**

When should the context be summarized and pruned? The second block compares three triggering policies. **Every round** forces the model to summarize after each reasoning round, whereas **Before final answer** invokes the summary once immediately before answering. **Model-selected** allows the model to decide whether and when summarization is necessary. Frequent forced summarization can disrupt reasoning, while summarizing only before the final answer cannot control earlier context accumulation. The model-selected policy achieves the highest accuracy (42.37%) and pruning rate (62.42%), demonstrating that adaptive summarization is more effective than fixed triggering schedules. We next ablate the KL-based decision rule by varying the minimum number of high-sensitivity tokens required to select a segment for evidence reconstruction, $\kappa \in 1, 2, 3$, while fixing $\tau = 0.2$. The settings $\kappa = 1$ and $\kappa = 2$ yield identical results. Increasing κ to 3 slightly improves the pruning rate but reduces accuracy. Based on this local stability, we adopt $(\tau, \kappa) = (0.2, 2)$ as the default setting.

6 Conclusion

In this paper, we identify *textual debt* as a key failure mode of multi-step MLLM agents, where accumulated reasoning dominates the context and weakens reliance on visual evidence. To address this problem, we propose SPARE, an OPSD-inspired, KL-guided framework that uses compact task-state summaries as privileged diagnostic context to prune redundant reasoning while preserving visual evidence. Across four visual tool-use benchmarks, SPARE achieves the highest average accuracy among the compared pruning methods on both Qwen backbones, improving task accuracy by up to 13.56 percentage points over Full Trace. In the aggregate analysis on BLINK-Jigsaw, GTA, and MNMS, it removes 57.81% of reasoning tokens while extending the empirical Pareto frontier of the evaluated baselines. Attention analysis further shows that pruning redirects attention toward image tokens. More broadly, our findings suggest that reasoning history should be treated as dynamic working memory rather than immutable context. These results demonstrate that selective forgetting can reduce textual dominance while preserving task-relevant evidence in long-context multimodal reasoning.

Ethics Statement

This work uses publicly available benchmarks and pretrained vision-language models and does not involve the collection of new personal data, human-subject studies, or private user information. We follow the licenses and intended research uses of the datasets, models, and tools employed in our experiments. We do not anticipate significant ethical risks beyond those already associated with the underlying models and public benchmarks.

Generative AI Use Disclosure

Generative AI tools were used solely for language editing to improve the grammar, clarity, and readability of the manuscript. They were not used to formulate the research questions or draw the conclusions. All AI-assisted edits were reviewed and validated by the authors, who take full responsibility for the final content of this work.

Reproducibility Statement

To support reproducibility, we will release the implementation of SPARE, together with the training and inference code, evaluation scripts, prompts, model and decoding configurations, and the metadata necessary to reproduce the reported experiments. All experiments are conducted on publicly available benchmarks, which can be obtained from their official sources subject to their respective licenses. The paper and appendices provide the benchmark settings, comparison methods, token-accounting protocol, hyperparameters, evaluation metrics, and computational resources used in our experiments.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes, 2024. URL https://arxiv.org/abs/2306.13649.

Ashutosh Chaubey, Jiacheng Pang, and Mohammad Soleymani. Mod-dpo: Towards mitigating cross-modal hallucinations in omni llms using modality decoupled preference optimization, 2026. URL https://arxiv.org/abs/2603.03192.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models, 2024. URL https://arxiv.org/abs/2403.06764.

Mehrdad Fazli, Bowen Wei, Ahmet Sari, and Ziwei Zhu. Mitigating hallucination in large vision-language models via adaptive attention calibration, 2025. URL https://arxiv.org/abs/2505.21472.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive, 2024. URL https://arxiv.org/abs/2404.12390.

Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. Hiagent: Hierarchical working memory management for solving long-horizon agent tasks with large language model, 2024. URL https://arxiv.org/abs/2408.09559.

Fucai Ke, Joy Hsu, Zhixi Cai, Zixian Ma, Xin Zheng, Xindi Wu, Sukai Huang, Weiqing Wang, Pari Delir Haghighi, Gholamreza Haffari, Ranjay Krishna, Jiajun Wu, and Hamid Rezatofighi. Explain before you answer: A survey on compositional visual reasoning, 2026. URL https://arxiv.org/abs/2508.17298.

Jaewoo Lee, Keyang Xuan, Chanakya Ekbote, Sandeep Polisetty, Yi R. Fung, and Paul Pu Liang. Tamp: Token-adaptive layerwise pruning in multimodal large language models, 2025. URL https://arxiv.org/abs/2504.09897.

Pengyu Li, Zhitao Gao, Lingling Zhang, Muye Huang, Yuanming Li, Fangzhi Xu, and Jun Liu. Visual-opsd: Cross-modal on-policy self-distillation for efficient unified multimodal reasoning, 2026a. URL https://arxiv.org/abs/2606.18974.

Zeju Li, Jianyuan Zhong, Ziyang Zheng, Xiangyu Wen, Zhijian Xu, Yingying Cheng, Fan Zhang, and Qiang Xu. Making slow thinking faster: Compressing llm chain-of-thought via step entropy, 2026b. URL https://arxiv.org/abs/2508.03346.

Zongxia Li, Xiyang Wu, Hongyang Du, Fuxiao Liu, Huy Nghiem, and Guangyao Shi. A survey of state of the art large vision language models: Alignment, benchmark, evaluations and challenges, 2026c. URL https://arxiv.org/abs/2501.02189.

Zixian Ma, Weikai Huang, Jieyu Zhang, Tanmay Gupta, and Ranjay Krishna. m&m's: A benchmark to evaluate tool-use for multi-step multi-modal tasks, 2024. URL https://arxiv.org/abs/2403.11085.

Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. Counterfactual vqa: A cause-effect look at language bias, 2021. URL https://arxiv.org/abs/2006.04315.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models, 2026. URL https://arxiv.org/abs/2403.15388.

Zhaochen Su, Peng Xia, Hangyu Guo, Zhenhua Liu, Yan Ma, Xiaoye Qu, Jiaqi Liu, Yanshu Li, Kaide Zeng, Zhengyuan Yang, Linjie Li, Yu Cheng, Heng Ji, Junxian He, and Yi R. Fung. Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers, 2025. URL https://arxiv.org/abs/2506.23918.

Rinyoichi Takezoe, Yaqian Li, Zihao Bo, Anzhou Hou, Mo Guang, and Kaiwen Long. Learnpruner: Rethinking attention-based token pruning in vision language models, 2026. URL https://arxiv.org/abs/2604.23950.

Xudong Tan, Peng Ye, Chongjun Tu, Jianjian Cao, Yaoxin Yang, Lin Zhang, Dongzhan Zhou, and Tao Chen. Tokencarve: Information-preserving visual token compression in multimodal large language models, 2025. URL https://arxiv.org/abs/2503.10501.

Jize Wang, Zerun Ma, Yining Li, Songyang Zhang, Cailian Chen, Kai Chen, and Xinyi Le. Gta: A benchmark for general tool agents, 2024. URL https://arxiv.org/abs/2407.08713.

Sangmin Woo, Donguk Kim, Jaehyuk Jang, Yubin Choi, and Changick Kim. Don't miss the forest for the trees: Attentional vision calibration for large vision language models, 2025. URL https://arxiv.org/abs/2405.17820.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms, 2023. URL https://arxiv.org/abs/2312.14135.

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. Tokenskip: Controllable chain-of-thought compression in llms, 2025. URL https://arxiv.org/abs/2502.12067.

Yuxi Xie, Guanzhen Li, Xiao Xu, and Min-Yen Kan. V-dpo: Mitigating hallucination in large vision language models via vision-guided direct preference optimization, 2024. URL https://arxiv.org/abs/2411.02712.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models, 2026. URL https://arxiv.org/abs/2412.04467.

Qianhao Yuan, Jie Lou, Xing Yu, Hongyu Lin, Le Sun, Xianpei Han, and Yaojie Lu. Vision-opd: Learning to see fine details for multimodal llms via on-policy self-distillation, 2026. URL https://arxiv.org/abs/2605.18740.

Dongxu Zhang, Yiding Sun, Cheng Tan, Wenbiao Yan, Ning Yang, Jihua Zhu, and Haijun Zhang. Chain-of-thought compression should not be blind: V-skip for efficient multimodal reasoning via dual-path anchoring, 2026a. URL https://arxiv.org/abs/2601.13879.

Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms, 2025a. URL https://arxiv.org/abs/2412.01818.

Weichen Zhang, Zhui Zhu, Ningbo Li, Shilong Tao, Kebin Liu, and Yunhao Liu. Adaptinfer: Adaptive token pruning for vision-language model inference with dynamical text guidance, 2026b. URL https://arxiv.org/abs/2508.06084.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. Sparsevlm: Visual token sparsification for efficient vision-language model inference, 2025b. URL https://arxiv.org/abs/2410.04417.

Haozhe Zhao, Shuzheng Si, Liang Chen, Yichi Zhang, Maosong Sun, Mingjia Zhang, and Baobao Chang. Looking beyond text: Reducing language bias in large vision-language models via multimodal dual-attention and soft-image guidance, 2026a. URL https://arxiv.org/abs/2411.14279.

Jianfei Zhao, Feng Zhang, Xin Sun, Lingxing Kong, Zhixing Tan, and Chong Feng. Cross-image contrastive decoding: Precise, lossless suppression of language priors in large vision-language models, 2025. URL https://arxiv.org/abs/2505.10634.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models, 2026b. URL https://arxiv.org/abs/2601.18734.

Qiji Zhou, Ruochen Zhou, Zike Hu, Panzhong Lu, Siyang Gao, and Yue Zhang. Image-of-thought prompting for visual reasoning refinement in multimodal large language models, 2024. URL https://arxiv.org/abs/2405.13872.

Xuanyu Zhu, Yuhao Dong, Rundong Wang, Yang Shi, Zhipeng Wu, Yinlun Peng, YiFan Zhang, Yihang Lou, Yuanxing Zhang, Ziwei Liu, Yan Bai, and Yuan Zhou. Vtc-bench: Evaluating agentic multimodal models via compositional visual tool chaining, 2026. URL https://arxiv.org/abs/2603.15030.

A Limitations

SPARE is currently designed for multi-step multimodal agents with explicit reasoning and tool-use histories, making its application to single-turn or unstructured agents less direct. It also introduces additional computation for summary generation and context replay. However, adaptive triggering avoids this cost on short trajectories. The method requires no auxiliary model and allows the pruned context to be reused in subsequent steps. Although consistent results across three backbones and five benchmarks under the same pruning configuration support its generality, extending SPARE to additional modalities and agent architectures, together with more comprehensive end-to-end latency evaluation, remains an important direction for future work.

B Additional Preliminaries

B.1 Multi-Step Tool Use in MLLMs

We consider a vision language model (VLM) π_θ that answers a user question through multi-step, tool-augmented reasoning. The input consists of a question q and a set of image tokens $\mathcal{I}$, and the model may query a fixed set of external tools $\mathcal{T}$ over at most T interaction steps.

At step t, the model conditions on the current message history $\mathbf{H}_t$ and produces an assistant message

$$m_t = (h_t, a_t), \tag{11}$$

which is composed of a free-form reasoning span h_t and an action block a_t. The action block is either a tool invocation $\langle$tool_call$\rangle$ or a final answer $\langle$response$\rangle$. If a_t invokes a tool, the corresponding tool result is appended to the history, yielding $\mathbf{H}_{t+1}$, and the interaction proceeds to the next step; if a_t emits a final response, the episode terminates. We denote the sequence of past reasoning–action segments by $\mathcal{S}_t = \{\, s_i = (h_i, a_i) \,\}_{i<t}$.

B.2 On-policy self-distillation.

Knowledge distillation trains a student model π_θ to match a teacher distribution π_{teacher} over next tokens. Conventional off-policy distillation minimizes this divergence on fixed teacher or ground-truth sequences, which creates exposure bias because training prefixes differ from the student-generated prefixes encountered at inference (Agarwal et al., 2024). *On-policy* distillation reduces this mismatch by applying the distillation loss to trajectories sampled from the student itself, so the model is supervised on its own inference-time states (Agarwal et al., 2024).

In the on-policy self-distillation instantiation, the teacher and the student share the same model, so no external teacher is required (Zhao et al., 2026b). Concretely, given a multimodal input $(q, \mathcal{I})$, we first roll out a response on-policy, $y \sim \pi_\theta(\cdot \mid q, \mathcal{I})$, and then align the student's per-token distribution with a teacher distribution π_{teacher} defined over the same model:

$$\mathcal{L}_{\text{OPSD}}(\theta) = \mathbb{E}_{(q,\mathcal{I})}\, \mathbb{E}_{y\sim\pi_\theta(\cdot|q,\mathcal{I})} \Bigg[\frac{1}{L} \sum_{k=1}^{L} D_{\text{KL}}\Big(\pi_{\text{teacher}}(\cdot \mid y_{<k}, q, \mathcal{I}) \,\Big\|\, \pi_\theta(\cdot \mid y_{<k}, q, \mathcal{I})\Big) \Bigg], \tag{12}$$

where the gradient is taken with respect to the student parameters θ and the teacher distribution is treated as a fixed target (stop-gradient). Sampling y from π_θ makes the objective on-policy, while defining π_{teacher} from the same model makes it self-distillation. This formulation lets the model learn from its own generated trajectories. Our method draws on this perspective to compare predictions with and without summary conditioning, without optimizing a distillation loss during pruning.

B.3 Self-attention over modalities.

Within a transformer layer ℓ with attention heads indexed by b, the attention weight from a query token i to a key token j is

$$A_{ij}^{(\ell,b)} = \text{softmax}_j\left(\frac{q_i^{(\ell,b)} \cdot k_j^{(\ell,b)}}{\sqrt{d}}\right), \qquad \sum_{j=1}^{N} A_{ij}^{(\ell,b)} = 1, \tag{13}$$

where q and k are the query and key projections and d is the head dimension. For a query token i we measure how much of its attention is directed to the visual stream by summing over visual keys and averaging over heads,

$$a_{i\to\mathcal{I}}^{(\ell)} = \frac{1}{H} \sum_{b=1}^{H} \sum_{j\in\mathcal{I}} A_{ij}^{(\ell,b)}, \tag{14}$$

which we refer to as the visual attention ratio of token i at layer ℓ. Aggregating $a_{i\to\mathcal{I}}^{(\ell)}$ over the textual query tokens gives a scalar summary of how strongly the model attends to the image while producing its response. A well-documented empirical observation is that this ratio is high in the first few layers but decays sharply with depth, so that in deep layers the model attends almost entirely to textual tokens (Chen et al., 2024).

C Additional Experimental Details

C.1 Pruning Procedure and Summary Prompt

All benchmarks use the same SPARE pruning procedure. The agent generates a structured task-state summary when it invokes the internal `summarize_the_task` tool. This summary provides

Algorithm 1 SPARE pruning at a summary event

Require: Persistent history H, shared replay prefix ρ_0, task-state summary σ generated at the current event
Require: $K = 20$, $\tau = 0.2$, $\kappa = 2$, $\epsilon = 10^{-12}$
Ensure: Pruned persistent history H'
Extract candidate reasoning segments $(h_1, \ldots, h_M)$ from H
Concatenate the segments with a fixed delimiter to obtain replay tokens $x = (x_1, \ldots, x_N)$
Record the token positions $\mathcal{P}_i$ belonging to each segment h_i
for $k = 1, \ldots, N$ **do**
    Obtain $p_k^{\mathrm{stu}}(\cdot) = P_{\pi_\theta}(\cdot \mid \rho_0, x_{<k})$
    Obtain $p_k^{\mathrm{tea}}(\cdot) = P_{\pi_\theta}(\cdot \mid \rho_0, \sigma, x_{<k})$
    Let Ω_k^{stu} be the student top-K vocabulary support
    Use -20 for teacher log probabilities unavailable in the teacher top-K support
    Compute $d_k = \sum_{u \in \Omega_k^{\mathrm{stu}}} p_k^{\mathrm{stu}}(u)\big(\log p_k^{\mathrm{stu}}(u) - \log p_k^{\mathrm{tea}}(u)\big)$
end for
$d_{\min} \leftarrow \min_k d_k$, $d_{\max} \leftarrow \max_k d_k$
if $d_{\max} = d_{\min}$ **then**
    Set $\tilde{d}_k \leftarrow 0$ for every k
else
    Set $\tilde{d}_k \leftarrow (d_k - d_{\min})/(d_{\max} - d_{\min} + \epsilon)$ for every k
end if
$H' \leftarrow H$
for $i = 1, \ldots, M$ **do**
    $c_i \leftarrow \left|\{k \in \mathcal{P}_i \mid \tilde{d}_k > \tau\}\right|$
    if $c_i \geq \kappa$ **then**
        $e_i \leftarrow$ EXTRACTVISUALEVIDENCE(h_i, max_items $= 4$, max_chars $= 600$)
        Replace only the reasoning text of h_i in H' with e_i
    else
        Remove only the reasoning text of h_i from H'
    end if
end for
return H'

the conditioning context for the residual-sensitivity computation and temporarily guides the next reasoning step. It is not stored in the persistent trajectory.

Evidence reconstruction is implemented through programmatic extraction rules rather than an additional LLM rewriting call. For each high-sensitivity segment, the extractor selects concrete visual evidence from the original reasoning, retaining at most four evidence items with a total length of at most 600 characters. If no evidence satisfies the extraction rules, the reasoning segment is removed. Low-sensitivity segments are removed entirely. External tool calls, existing tool observations, and image content remain unchanged in both cases. Algorithm 1 summarizes the procedure. The function EXTRACTVISUALEVIDENCE denotes the rule-based extractor applied to the original reasoning. The limits on evidence items and characters are applied separately to each selected segment.

After pruning, the agent uses H' together with σ to generate the next reasoning step and action. The generated reasoning and action are recorded as usual. The summary itself is omitted from subsequent contexts.

Summary instruction. Across all benchmarks, the agent receives the summary-tool instruction shown in Figure 3. It specifies the task-state schema and encourages compact records of verified visual and tool facts. Evidence extraction is performed separately by programmatic rules and does not require an additional LLM prompt.

Task-State Summary Instruction

```
summarize_the_task: use the reason_mine_code visual-state schema:
`reasoning_so_far`, `completed_steps`, `current_state`,
`visual_evidence`, `key_observations`, `uncertainties`,
and `next_decision`.

`visual_evidence` has `coordinates`, `bboxes`, `crop_ids`,
`ocr_text`, `astar_outputs`, and `candidate_differences`.

Keep verified visual/tool facts compact; do not put unsupported
guesses or long prose in reasoning_so_far.

It is internal memory, not a final answer; after it returns,
continue with the next needed tool or final answer.
```

Figure 3: Summary instruction used across all benchmarks. The structured summary conditions the KL computation and temporarily guides the next reasoning step. Visual evidence is extracted separately by programmatic rules without an additional LLM rewriting call.

C.2 Reasoning-Token Accounting

All tool-based methods use the same backbone, decoding configuration, tool set, and execution environment. They differ only in how accumulated assistant reasoning is retained or reconstructed. Original tool calls, tool observations, and images remain unchanged. Task-state summaries are used temporarily for diagnosis and the next reasoning step. They are not stored in the persistent trajectory and are excluded from persistent-history token accounting.

Counting scope. For method m, let $\mathcal{D}_b$ denote the evaluated examples from benchmark b, and let $\mathcal{H}^m_{j,b}$ denote all completed tool-use reasoning segments generated for example $j \in \mathcal{D}_b$.

For each segment $h \in \mathcal{H}^m_{j,b}$, $R_{\text{orig}}(h)$ is the original assistant reasoning preceding its corresponding <tool_call> block, and $\widetilde{R}^m_{\text{final}}(h)$ is its final persistent reasoning-side representation. The latter equals the original reasoning for an unpruned segment, is empty when the reasoning is deleted, and contains the reconstructed structured visual evidence when the segment is compressed. Tool-call JSON, tool observations, images, final answers, and temporary task-state summaries are excluded. All token counts use the tokenizer and chat serialization of the evaluated backbone.

Multiple summary events. A trajectory may invoke the summary tool multiple times. Each reasoning segment is counted exactly once using its original text and final persistent representation, regardless of how many summary events inspect or modify it.

Original and retained reasoning tokens. The total number of original reasoning tokens generated by method m on benchmark b is

$$O_{m,b} = \sum_{j \in \mathcal{D}_b} \sum_{h \in \mathcal{H}^m_{j,b}} \text{Tok}(R_{\text{orig}}(h)) \,. \tag{15}$$

The number of reasoning tokens remaining in the persistent history is

$$K_{m,b} = \sum_{j \in \mathcal{D}_b} \sum_{h \in \mathcal{H}^m_{j,b}} \text{Tok}\Big(\widetilde{R}^m_{\text{final}}(h)\Big) \,. \tag{16}$$

Removed reasoning tokens. The total number of net reasoning tokens removed is

$$P_{m,b} = O_{m,b} - K_{m,b}. \tag{17}$$

The difference is not clipped if an evidence reconstruction is longer than its original reasoning segment.

Table 4: Complete per-model results on the three benchmarks. Acc. denotes accuracy, and Pr. denotes the reasoning-pruning rate. All values are percentages. Figure 2 aggregates the results using benchmark-size weighting: 150 samples for BLINK-Jigsaw, 59 answer-scored samples for GTA, and 228 samples for MNMS.

		BLINK-Jigsaw		GTA		MNMS	
Model	Method	Acc.	Pr.	Acc.	Pr.	Acc.	Pr.
Qwen3-VL-8B	Full Trace	74.00	0.00	42.37	0.00	60.96	0.00
	Random-33	70.67	34.97	38.98	46.59	57.46	34.49
	Random-50	60.67	46.31	44.07	54.99	58.33	50.37
	KL-guided	67.33	15.66	40.68	36.08	60.96	14.15
	Delete All	62.00	100.00	40.68	100.00	57.02	100.00
	Visual Evidence-Only	59.33	59.47	44.07	73.69	56.58	87.40
	SPARE	72.00	30.32	49.15	55.49	57.46	67.89
Qwen3-VL-30B-A3B	Full Trace	67.33	0.00	45.76	0.00	36.84	0.00
	Random-33	61.33	28.14	52.54	35.02	17.11	34.30
	Random-50	68.67	46.92	61.02	51.93	16.23	50.60
	KL-guided	67.33	6.93	57.63	35.69	42.11	22.97
	Delete All	57.33	99.42	50.85	100.00	24.56	100.00
	Visual Evidence-Only	64.00	79.94	59.32	78.84	25.44	89.68
	SPARE	64.00	78.36	59.32	10.44	49.56	67.83
Nemotron-Nano-VL-8B	Full Trace	45.33	0.00	40.68	0.00	26.75	0.00
	Random-33	44.67	32.87	38.98	6.20	16.67	30.86
	Random-50	48.67	64.37	38.98	36.61	16.23	49.41
	KL-guided	46.00	7.36	38.98	38.07	20.61	9.06
	Delete All	46.67	100.00	37.29	100.00	21.49	100.00
	Visual Evidence-Only	44.00	48.53	37.29	56.08	17.98	80.14
	SPARE	48.67	67.17	42.37	62.42	21.93	47.80

Reasoning-token pruning rate. The reasoning-token pruning rate reported in the main table is

$$r_{m,b} = 100 \times \frac{O_{m,b} - K_{m,b}}{O_{m,b}}. \tag{18}$$

Higher values indicate stronger compression of the persistent reasoning history. The rate is computed from aggregate token counts rather than by averaging per-example percentages. Full Trace has $r_{m,b} = 0\%$. Direct Answer contains no comparable tool-use reasoning history and is reported as zero for consistency with the main table.

C.3 Accuracy-Pruning Trade-off

Figure 2 compares SPARE with controls that isolate three aspects of reasoning pruning: whether to prune indiscriminately, which reasoning steps to prune, and what information to retain after pruning. *Full Trace* preserves the complete reasoning history. *Random-33/50* independently removes each completed reasoning span with probability 0.33 or 0.50, while *Delete All* removes all historical reasoning. *Visual Evidence-Only* compresses every eligible reasoning span into structured visual evidence. Finally, *KL-guided* uses the same summary-conditioned KL signal as SPARE, but removes only low-KL reasoning and retains high-KL spans in full. All pruning methods leave the original tool actions, observations, and images unchanged.

Indiscriminate pruning. The first group of controls shows that reasoning cannot be removed indiscriminately. Aggregate accuracy drops from 48.81% with *Full Trace* to 41.95% with *Random-33*, and remains low at 42.64% with *Random-50* and 42.71% with *Delete All*. Although their pruning rates increase from 32.26% to 50.64% and 99.93%, the three pruning controls perform similarly

and substantially below *Full Trace*. Thus, randomly or uniformly removing reasoning discards task-relevant information, and the pruning rate alone does not determine performance.

Which reasoning steps should be pruned? Compared with *Delete All*, *KL-guided* improves accuracy from 42.71% to 48.36% by retaining high-KL reasoning and removing only low-KL spans. This confirms the importance of selecting reasoning steps according to their residual contribution. However, because high-KL spans are retained in full, *KL-guided* removes only 16.40% of reasoning, showing that selection alone is accurate but overly conservative.

What should be retained? *Visual Evidence-Only* retains structured visual information instead of deleting every reasoning span, but improves aggregate accuracy only marginally over *Delete All* (42.86% vs. 42.71%). This indicates that visual-evidence reconstruction alone is insufficient when applied indiscriminately. In contrast, SPARE applies reconstruction only to high-KL spans and removes low-KL reasoning entirely. Compared with *KL-guided*, it increases the pruning rate from 16.40% to 57.81% while improving accuracy from 48.36% to 50.34%. These comparisons show that KL-based selection determines which reasoning steps remain important, while evidence-preserving reconstruction determines how their useful information can be retained compactly.

Consequently, SPARE lies above the baseline Pareto frontier in Figure 2, achieving the highest aggregate accuracy of 50.34% while pruning 57.81% of reasoning tokens. The per-model results in Table 4 further show that this advantage includes both accuracy-improving and accuracy-preserving cases. On MNMS with Qwen3-VL-30B-A3B, SPARE achieves 49.56% accuracy with 67.83% pruning, compared with 42.11% accuracy and 22.97% pruning for *KL-guided*. On BLINK-Jigsaw with Qwen3-VL-8B, it removes 30.32% of reasoning while remaining within two accuracy points of *Full Trace*. These aggregate trends are consistent with the controlled ablations in Table 3, which further isolate the complementary roles of KL-based selection and visual-evidence reconstruction.

C.4 Computational Cost

All training and test-time experiments were conducted using eight NVIDIA A100 GPUs. At each pruning event, SPARE generates a task-state summary and performs two forward replays of the candidate reasoning history to compute the summary-conditioned KL signal. This introduces temporary inference overhead but requires neither an auxiliary model nor parameter updates at test time. Adaptive triggering avoids this cost for short trajectories, while longer trajectories can amortize it by reusing the reduced context over subsequent interaction steps.

D Supplementary Analyses of Textual Interference

D.1 Controlled Diagnostic of Textual Interference

Relation to the main experiments. The main experiments evaluate the complete SPARE pipeline across multiple MLLM backbones and full benchmarks. Complementary to those end-to-end results and the attention analysis, we conduct an additional controlled diagnostic to test whether pruning stale reasoning makes the model less likely to follow an obsolete textual hypothesis when later visual or tool evidence supports a different answer. This diagnostic is separate from the aggregate benchmark evaluation.

Controlled construction. We construct a subset from GTA containing audited cases in which an early reasoning segment expresses a plausible but incorrect answer and a later tool observation provides corrective evidence. Across conditions, the original question, image, tool calls, action blocks, and tool observations are held fixed; only the retained reasoning history changes.

We compare three conditions. **Full Trace** keeps the complete history, including the stale hypothesis. **Random-1** removes one eligible reasoning segment at random, providing a control for context shortening without targeted selection. **SPARE** automatically applies its pruning procedure to the eligible reasoning history while preserving the tool interaction and visual evidence. Every audited conflict case is retained in the evaluation, including cases in which pruning does not activate and the context therefore remains unchanged.

Metrics. We report forced-choice accuracy (Acc.), which measures how often the model selects the answer supported by the later evidence, and stale-copy rate (SCR), which measures how often the final prediction instead repeats the obsolete textual hypothesis. Formally, for method m,

$$\mathrm{Acc}_m = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left[\hat{y}_i^m = y_i\right], \tag{19}$$

$$\mathrm{SCR}_m = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left[\hat{y}_i^m = s_i\right], \tag{20}$$

where y_i and s_i denote the ground-truth and stale answers, respectively. A lower SCR indicates that the final decision is less likely to follow obsolete textual reasoning when it conflicts with later evidence.

Table 5: Controlled textual-interference results on the constructed GTA subset. All values are percentages. Random-1 results are averaged over random draws; cases without an eligible deletion remain unchanged. SPARE is evaluated as an automatic procedure rather than with a manually specified deletion mask.

Model	Method	Acc. ↑	SCR ↓
Qwen3-VL-8B	Full Trace	63.64	36.36
	Random-1	72.73	27.27
	SPARE	**90.91**	**9.09**
Qwen3-VL-30B-A3B	Full Trace	54.55	45.45
	Random-1	72.73	22.73
	SPARE	**81.82**	**18.18**

Results. For Qwen3-VL-8B, SPARE improves forced-choice accuracy from 63.64% to 90.91% and reduces stale-answer copying from 36.36% to 9.09%. For Qwen3-VL-30B-A3B, accuracy increases from 54.55% to 81.82%, while SCR decreases from 45.45% to 18.18%. Thus, across both backbones, SPARE improves accuracy and reduces stale copying by 27.27 percentage points.

Random-1 produces smaller improvements than SPARE for both backbones. This comparison indicates that the benefit cannot be explained solely by shortening the reasoning history: selecting which reasoning content to remove is important for reducing interference from stale textual hypotheses. Free-form answer accuracy remains unchanged between Full Trace and SPARE, suggesting that the observed effect reflects reduced reliance on conflicting textual history rather than a general change in task-solving ability.

Scope. This deliberately controlled diagnostic is intended as mechanism-level support for the textual-debt motivation, rather than as a separate benchmark result. It shows that the automatic SPARE procedure can reduce interference from stale reasoning while retaining the later visual and tool evidence. Because the diagnostic isolates a specific form of text–evidence conflict, it should not be interpreted as a general comparison between SPARE and every possible reasoning-compression strategy.

D.2 Attention Visualization After Pruning

To further examine whether pruning reduces textual dominance and restores reliance on visual evidence, we visualize per-layer text-to-image attention averaged over 10 tool-use trajectories. After applying SPARE, attention to image tokens increases across nearly all layers. This supports our motivation: redundant accumulated text can dominate the context and distract the model from visual evidence, while pruning it reallocates computation back to the image. The cross-layer attention shift provides additional evidence that suppressing textual dominance restores the role of visual evidence in downstream reasoning.

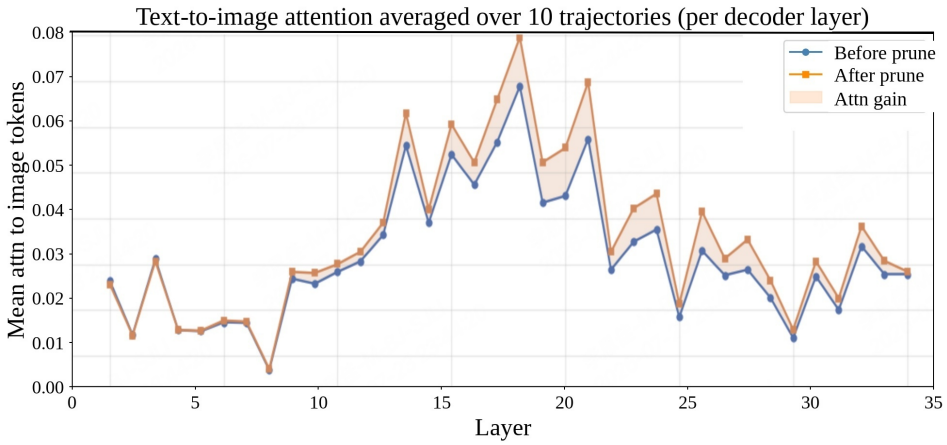


Figure 4: Text-to-image attention across decoder layers, averaged over 10 tool-use trajectories. Blue and orange denote attention before and after applying SPARE, respectively, and the shaded region shows the gain. Applying SPARE increases the attention allocated to image tokens across nearly all decoder layers.

D.3 Case Study

We use a single visual counting example to illustrate how SPARE improves agent behaviour. The question asks how many purple balloons appear in the image, with ground truth B. Without pruning, the agent obtains only a weak perceptual signal in its first observation and, instead of re-examining the image, invokes a grayscale tool and produces long stretches of elaborative text on top of this uncertain cue. The added prose dilutes the visual evidence, and the agent settles on the wrong answer D after several redundant reasoning cycles.

With SPARE, 399 tokens of reasoning are removed. The removed passages largely elaborate on the agent's weak initial visual observation, while selected segments are reconstructed as concise visual evidence. After pruning, the agent shifts from repeated textual elaboration to renewed inspection of the image. It performs a targeted bounding-box crop and reaches the correct answer B. This case illustrates how pruning accumulated reasoning can redirect the agent toward visual evidence and more targeted tool use.

D.4 Analysis of the High-KL Signal

Table 6: Label distribution of 66 high-KL reasoning paragraphs, with one paragraph selected from each trajectory that invokes the summary tool among 100 randomly sampled trajectories.

Label	**Proportion**
`covered_but_more_specific`	39.39%
`partially_covered_by_summary`	48.48%
`not_covered_by_summary`	4.55%
`conflicts_with_summary`	7.58%
Coverage gap (union of last three)	*60.61%*

Experimental Setup. To characterize the relationship between high residual sensitivity and summary coverage, we randomly sample 100 agent trajectories, of which 66 invoke the summary tool. From each of these 66 trajectories, we select one high-KL reasoning paragraph and pair it with the task-state summary used to compute its KL scores, yielding 66 instances for semantic annotation. The scores are computed by replaying the same reasoning prefixes with and without summary conditioning.

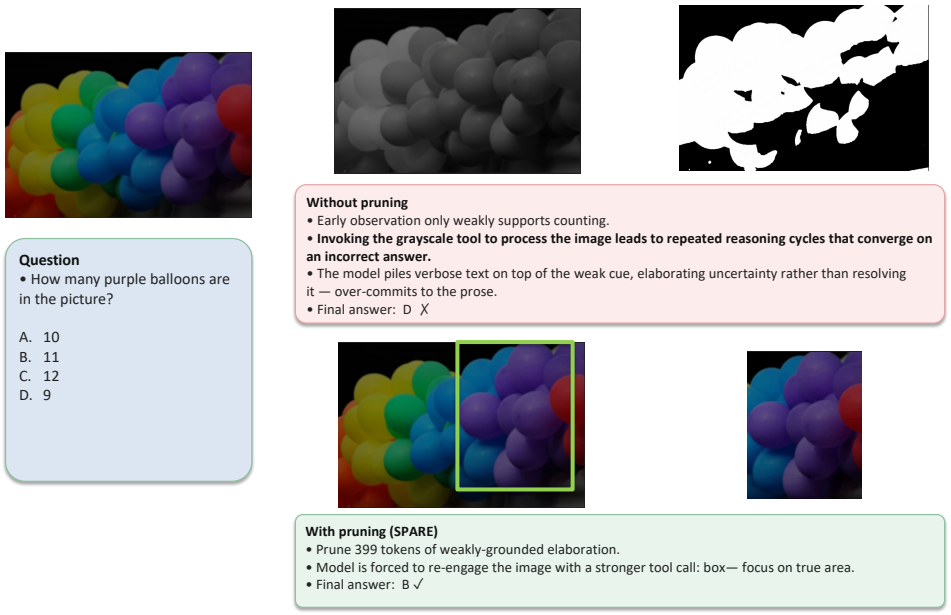


Figure 5: Case study on a visual counting question (ground truth: B). Without pruning (top), the agent obtains only a weak perceptual signal, invokes a grayscale tool, and piles verbose elaboration on top of the uncertain cue, converging on the wrong answer D. With SPARE (bottom), 399 tokens of weakly grounded reasoning are pruned while evidence-critical segments are preserved, which restores the agent's attention to the image and prompts it to re-engage the visual input with a decisive bounding-box crop, arriving at the correct answer B. This example illustrates that SPARE reduces textual redundancy rather than visual evidence, redirecting the agent toward stronger image-grounded actions.

To ensure annotation quality, each passage is evaluated by **GPT-5.6**, a stronger judge model, which assesses the semantic relationship between the high-KL source passage and its corresponding summary. Specifically, the judge is prompted to assign one of four mutually exclusive labels that characterize the degree to which the summary accounts for the information present in the passage (see label definitions in the analysis paragraph below).

High-KL Segments and Summary Coverage. Table 6 summarizes the annotations of 66 high-KL reasoning paragraphs. We distinguish four relationships between each source passage and its summary.

- `conflicts_with_summary` denotes passages that directly contradict a claim made in the summary.
- `not_covered_by_summary` denotes passages whose content is entirely absent from the summary.
- `partially_covered_by_summary` denotes passages that are acknowledged in the summary but have key details omitted or compressed.
- `covered_but_more_specific` denotes passages whose main point is captured but whose finer details, such as specific entities, quantities, or conditions, are not retained.

Among the annotated paragraphs, 48.48% are partially covered, 4.55% are entirely absent from the summary, and 7.58% conflict with its claims. Together, these categories account for 60.61% of the sample. Partial coverage is the most common category, indicating that incomplete representation of source content is more prevalent than direct contradiction in these samples.

The remaining 39.39% fall into the `covered_but_more_specific` category. Although the summary captures their overall meaning, it omits finer details. This mismatch in specificity may

contribute to the observed high KL, as summary conditioning can affect predictions involving details that the summary does not fully represent.

These observations support using high KL as a signal for selecting segments whose visual evidence should be preserved. SPARE reconstructs such segments as concise visual evidence, while the KL-selection ablation retains their original text.

D.5 Analysis of the Low-KL Signal

Table 7: Label distribution of 82 pruned low-KL reasoning paragraphs, with one paragraph selected from each trajectory that invokes the summary tool among a separate sample of 100 trajectories.

Label	Proportion
`summary_covers_segment`	68.29%
`local_prefix_sufficient`	17.07%
`summary_irrelevant_to_segment`	14.63%

Experimental Setup. To examine the relationship between low residual sensitivity and reasoning redundancy, we independently sample another 100 agent trajectories, of which 82 invoke the summary tool. From each of these 82 trajectories, we select one low-KL reasoning paragraph selected for deletion and pair it with the corresponding task-state summary, yielding 82 instances for semantic annotation. Low-KL paragraphs are identified using the normalized token scores and the segment-level count rule used in the main method.

Low-KL Segments and Reasoning Redundancy. Table 7 summarizes the annotations of 82 pruned low-KL reasoning paragraphs. We distinguish three categories based on their relationship to the corresponding summary and local reasoning prefix.

- `summary_covers_segment` denotes passages whose information is faithfully represented in the summary.
- `local_prefix_sufficient` denotes passages whose content is recoverable from the local reasoning prefix without relying on the summary.
- `summary_irrelevant_to_segment` denotes passages with no meaningful coverage relationship to the summary. Their content is neither confirmed nor contradicted by it.

Among the annotated paragraphs, 68.29% are covered by the summary, while 17.07% contain information recoverable from the local prefix. The remaining 14.63% have no meaningful coverage relationship with the summary. These results show that summary coverage and local redundancy are common among the sampled low-KL paragraphs.

These observations support using low KL as a signal for identifying pruning candidates. The segments unrelated to the summary also indicate that low KL does not exclusively reflect semantic coverage. The annotations provide an empirical interpretation of the selection signal, while the component ablations evaluate its downstream effectiveness.

E Theoretical Interpretation of Residual Sensitivity

What the KL signal measures. We use the notation from Section 4.1. At replay position k, p_k^{stu} and p_k^{tea} are the model's predictive distributions without and with the task-state summary σ, respectively. Both distributions condition on the same historical reasoning prefix $x_{<k}$. Their difference therefore measures how the prediction at this position responds to summary conditioning.

For the theoretical analysis, we define the full-vocabulary reverse KL as

$$d_k^{\mathrm{full}} = \sum_u p_k^{\mathrm{stu}}(u) \log \frac{p_k^{\mathrm{stu}}(u)}{p_k^{\mathrm{tea}}(u)}, \tag{21}$$

where the sum covers the entire vocabulary and logarithms are natural. The superscript distinguishes this quantity from the truncated score d_k used in the implementation.

Why small KL indicates predictive stability. Pinsker's inequality gives

$$\sum_u \left| p_k^{\mathrm{stu}}(u) - p_k^{\mathrm{tea}}(u) \right| \le \sqrt{2 d_k^{\mathrm{full}}}. \tag{22}$$

The left-hand side sums the absolute changes in token probabilities caused by adding the summary. Thus, a small d_k^{full} guarantees that the total probability change is small at this replay position. This provides a mathematical interpretation of low residual sensitivity as limited predictive response to summary conditioning.

Connection to the pruning rule. In practice, SPARE computes d_k over the student top-K support Ω_k^{stu}, using the teacher-log-probability fallback specified in Section 4.1. It then obtains $\tilde{d}_k$ through event-level normalization. These normalized scores express relative sensitivity within a pruning event rather than an absolute bound on probability changes.

For a reasoning segment h_i, the count-based rule selects it for evidence extraction when at least κ positions in $\mathcal{P}_i$ satisfy $\tilde{d}_k > \tau$. The remaining segments are deleted. This rule uses localized sensitivity to decide where to attempt visual evidence preservation, without requiring the entire reasoning segment to be retained.

The theoretical bound explains the full-KL sensitivity measure, rather than proving the segment-level pruning rule. It does not directly apply to the truncated or normalized scores, and it concerns summary conditioning rather than history deletion. The semantic analyses and component ablations provide the empirical evidence connecting this sensitivity signal to effective pruning.