Retrieve, Integrate, and Synthesize: SPATIAL-SEMANTIC GROUNDED LATENT VISUAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal Large Language Models (MLLMs) have made remarkable progress on vision-language reasoning, yet most methods still compress visual evidence into discrete textual thoughts, creating an information bottleneck for fine-grained perception. Recent latent visual reasoning methods attempt to reason in continuous hidden states, but we find that they suffer from insufficient manifold compatibility: latent trajectories drift away from pretrained reasoning circuits, collapse into instance-agnostic patterns, and are often bypassed during answer generation. To address these issues, we propose ***RIS*** *(**R**etrieve, **I**ntegrate, and **S**ynthesize)*, a spatial-semantic grounded framework that develops latent reasoning as a compatible extension of pretrained MLLM computation. We first construct a step-wise grounded reasoning dataset with bounding boxes and region-specific semantic descriptions. Built on this supervision, ***RIS*** anchors latent tokens to both spatial and semantic evidence, enforces their causal role through a progressive attention bottleneck, and introduces short language transition tokens to bridge synthesized latent states back to vocabulary-aligned decoding. Experiments on V*, HRBench4K, HRBench8K, MMVP, and BLINK show consistent improvements over closed/open-source and latent reasoning baselines. Further analyses demonstrate that ***RIS*** learns diverse, interpretable, and progressively integrated latent trajectories, offering a practical path toward faithful internal visual reasoning in MLLMs.

1 INTRODUCTION

Multimodal Large Language Models (MLLMs) have achieved remarkable success across diverse vision-language tasks, largely due to Chain-of-Thought (CoT) reasoning(Wei et al., 2022; Kojima et al., 2022). However, these models still treat visual information as static preconditions, converting continuous visual features into discrete textual tokens and reasoning only within the textual domain(Li et al., 2022). This creates an inherent bottleneck: *fine-grained visual evidence must be compressed into language tokens before it can participate in reasoning*. Recent "thinking with images"(Su et al., 2025) methods alleviate this issue by injecting visual evidence through external tools(Yang et al., 2023; Liu et al., 2024) or programmatic operations(Gupta & Kembhavi, 2023; Surís et al., 2023; Hu et al., 2024), but their flexibility is limited by predefined tool interfaces and external execution. This necessitates a more unified solution to move intermediate visual reasoning inside the model, allowing it to manipulate question-relevant visual evidence directly in continuous hidden representations.

Latent visual reasoning offers a promising path toward this goal. Unlike text-based CoT, latent states provide an expressive workspace where visual patterns and abstract concepts can be represented without being discretized into language(Su et al., 2025; Yu et al., 2026). Yet this freedom also introduces a fundamental tension. Since the model's reasoning behavior and decoding interface are largely shaped by language pretraining, effective latent visual reasoning must not only exploit the expressive capacity of a latent visual manifold $\mathcal{M}_{vis}$, but also remain compatible with the vocabulary-aligned manifold $\mathcal{M}_{vocab}$ where pretrained reasoning circuits and language-grounded decoding are organized. Existing methods such as LVR(Li et al., 2025) and Monet(Wang et al., 2025a) take important steps by reconstructing visual tokens from latent states or generating continuous embeddings as intermediate visual thoughts, but they do not fully resolve this compatibility problem.

In this work, we first analyze *why existing latent visual reasoning methods remain ineffective despite forming distinct latent visual representations*. Recent causal mediation study(Li et al., 2026) reveals

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.

pronounced *Input–Latent* and *Latent–Answer* disconnects, where latent tokens are weakly grounded in visual inputs and exert limited influence on final predictions. Our empirical analysis further shows that these failures are closely tied to manifold divergence. Specifically, (1) weakly supervised hidden states drift from the pretrained vocabulary-aligned manifold and tend to collapse into highly similar, instance-agnostic trajectories; (2) answer tokens usually bypass latent tokens and rely directly on the input image and question; (3) the model must converge high-entropy latent visual states abruptly to low-entropy answer tokens, which can induce representation mismatch during language decoding.

To address these challenges, we propose ***RIS*** ***(Retrieve, Integrate, and Synthesize)***, a grounded latent visual reasoning framework that develops latent space as a compatible extension of pretrained reasoning circuits rather than a detached visual manifold. To support training, we first construct a step-wise grounded visual reasoning dataset with $96k$ samples in which each reasoning step is paired with bounding-box spatial supervision and a region-specific semantic description. Built on this spatial-semantic supervision, ***RIS*** structures latent tokens as directed visual evidence retrieval states: bounding-box supervision anchors *where to look*, semantic alignment specifies *what is seen*, and a progressive attention mask forces task-relevant evidence to flow through latent tokens instead of being bypassed during answer generation. Slots beyond the annotated reasoning steps are optimized solely through the final-answer objective, endowing them with the emergent ability to integrate and synthesize evidence retrieved by grounded slots. Finally, we demonstrate that generating a slightly elaborated answer between latent reasoning and final option-level answer acts as manifold transition tokens since it gradually reduces the entropy of reasoning paths from latent states to low-entropy answer tokens rather than abrupt degradation, while providing dense supervision during training.

We evaluate ***RIS*** on five challenging visual reasoning benchmarks. ***RIS*** consistently outperforms strong baselines, with particularly clear gains on tasks requiring localization, structured visual search, and multi-step perceptual reasoning. Further analyses show that ***RIS*** produces more diverse, interpretable, and task-dependent latent trajectories. Our contributions are summarized as follows:

- We systematically analyze latent visual reasoning in MLLMs, identifying the interaction between the vocabulary-aligned manifold $\mathcal{M}_{vocab}$ and the latent visual manifold $\mathcal{M}_{vis}$, and revealing manifold divergence, trajectory collapse, and answer bypassing as key obstacles.
- We construct a $96k$-sample ***Grounded Latent Supervision Dataset (GLSD)*** and propose ***RIS***, a spatial-semantic grounded latent reasoning framework that structures latent tokens to retrieve task-relevant visual evidence while developing latent space as a compatible extension of pretrained reasoning circuits rather than a detached visual manifold.
- We demonstrate improvements across visual reasoning benchmarks, especially on localization and multi-step visual reasoning tasks, and show that ***RIS*** learns diverse, interpretable, and progressively integrated latent reasoning trajectories with state-of-the-art performance.

2 Related Work

From Static Perception to Internal Visual Imagination. Most current MLLMs adopt text-space CoT reasoning to solve complex visual tasks, treating visual inputs as static premises for language-based inference(Zhang et al., 2023; Wang et al., 2025c). Although effective, such methods reason through discrete text tokens, which provide an indirect and lossy representation for fine-grained visual understanding. Recent *Thinking with Images* (Su et al., 2025) methods alleviate this limitation by using external visual tools to manipulate and inject intermediate visual evidence(Yang et al., 2023). However, their effectiveness is constrained by the availability, design, and granularity of predefined tools. This motivates internal visual reasoning, where models reason over visual evidence in continuous latent states rather than translating it into text or pixels.

Latent Reasoning. Recent studies have explored continuous latent spaces as an alternative to discrete token-level reasoning. Representative approaches include utilizing recursive hidden states for breadth-first search(Hao et al., 2024), self-distillation of explicit reasoning traces(Shen et al., 2025), and implicit reasoning via superposed latent chains(Deng et al., 2026). While these methods enhance reasoning efficiency, they remain constrained in textual space. Extending them to MLLMs is non-trivial: representing visual evidence by vocabulary embeddings or weakly supervised hidden states can distort fine-grained cues such as texture, color, and spatial layout. Effective visual latent

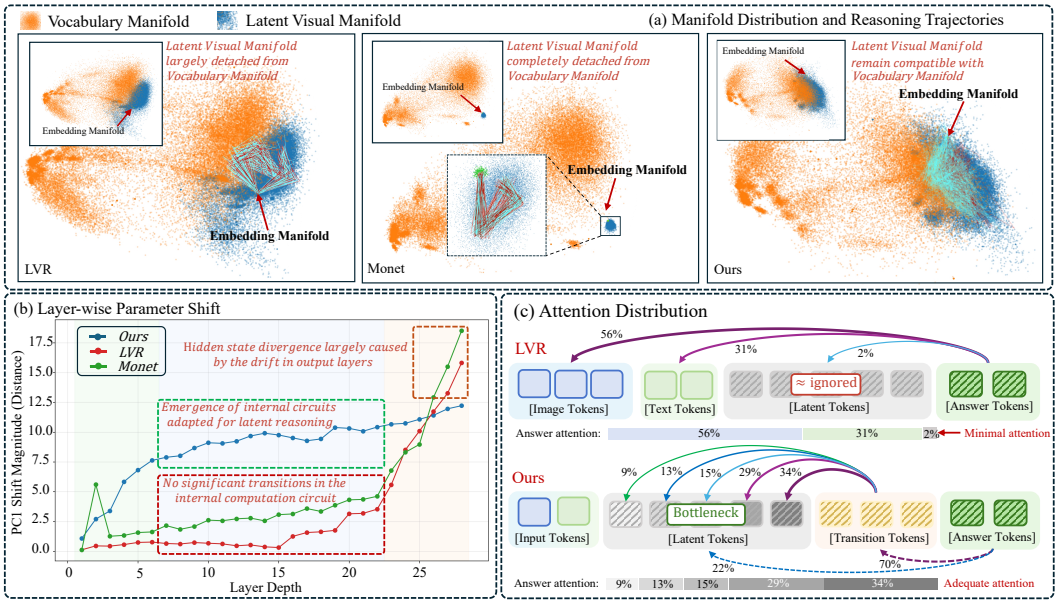


Figure 1: Geometric analysis of latent reasoning paradigms: (a) manifold distribution and trajectories, (b) layer-wise parameter shift relative to the base model, and (c) attention pattern of answer tokens.

reasoning requires a visual manifold that can preserve rich perceptual evidence while remaining compatible with language-grounded reasoning.

Latent Visual Reasoning. To move beyond static perception toward internal visual imagination, recent paradigms have explored performing logical deductions directly within the latent space. LVR(Li et al., 2025) performs autoregressive reasoning within the visual embedding space by reconstructing task-critical tokens from latent states. Monet generates continuous embeddings that serve as intermediate visual thoughts and aligns them with the visual semantic space through a distillation pipeline(Wang et al., 2025a). Mirage further treats hidden states as latent visual tokens to build multimodal reasoning trajectories without pixel-level image synthesis(Yang et al., 2025). Despite these advances, recent diagnostic studies reveal a persistent causality gap: *latent tokens are often weakly grounded in visual inputs and exert limited influence on final answers* (Li et al., 2026). Our analyses further reveal a fundamental *manifold divergence* in existing baselines, where latent trajectories drift into deep, uncalibrated regions far from the pre-trained semantic anchors. These limitations thus motivate our grounded latent reasoning framework.

3 Analysis on Reasoning Manifold

To characterize how latent reasoning interacts with pretrained representations, we analyze the trajectories of hidden states relative to the base-model and vocabulary embedding spaces. Given an evaluation set of N samples, each forward-decoding pass produces last-layer hidden states $\{\mathbf{h}_t^{(i)}\}_{t=1}^{T_i}$ from the latent and answer phases. We aggregate these states as $\mathcal{H}_{RIS}$ and collect hidden states from the frozen base model, $\mathcal{H}_{\text{base}}$, together with the vocabulary embedding matrix $\mathbf{E} \in \mathbb{R}^{V \times d}$. We jointly fit PCA on $\mathcal{H}_{\text{base}}$, $\mathcal{H}_{RIS}$, and $\mathbf{E}$, and project the resulting reasoning trajectories onto the leading two principal components.

3.1 Manifold Compatibility and Trajectory Dynamics

We use *manifold* to denote the empirical support of hidden-state or embedding representations. Figure 1(a) shows that the latent states of LVR and Monet are substantially separated from the pretrained vocabulary-aligned manifold, indicating a representational distribution shift that may reduce compatibility with pretrained reasoning and language decoding. In contrast, our latent trajectories remain more closely connected to this representation regime.

The trajectory view further shows that successful reasoning paths tend to remain connected to the vocabulary-aligned manifold, whereas failed paths are more often confined to detached latent regions. This does not require latent states to return explicitly to the vocabulary manifold at every step; rather, it suggests that effective latent reasoning should extend the pretrained representation regime while remaining compatible with it. In this sense, *latent visual reasoning should develop as a compatible extension of the pretrained reasoning manifold rather than a detached alternative.*

3.2 Layer-wise Adaptation Pattern

Figure 1(b) further compares layer-wise parameter shifts from the base model. LVR and Monet exhibit limited changes in middle layers but substantially larger shifts near the output layers, suggesting that adaptation is concentrated around the decoding interface. In contrast, ***RIS*** shows smoother and more sustained adaptation across middle layers, with milder shifts near the output, indicating that latent reasoning is incorporated more gradually into the model's internal computation while preserving the pretrained decoding regime.

3.3 Latent Bypassing and Trajectory Collapse

Figure 1(c) examines how answer tokens route information during decoding. In LVR, answer tokens attend predominantly to the original image and text inputs while assigning little attention to latent tokens, suggesting that the latent pathway is largely bypassed. This is consistent with prior observations that latent states can exert limited influence on final predictions (Li et al., 2026). We treat attention only as a diagnostic of information routing; the functional dependence on latent states is evaluated separately through inference-time interventions in Section 5.3 and Appendix E, with attention details provided in Appendix F.

Trajectories in Figure 1(a) reveal a complementary failure mode: latent trajectories from LVR and Monet are highly similar across samples, indicating limited instance-specific variation and trajectory collapse. Together, these observations suggest that learning a distinct latent manifold is insufficient if its states are weakly input-dependent or bypassed during answer generation.

Taken together, ***effective latent reasoning requires latent states that remain compatible with pretrained computation, preserve instance-specific grounded information, and form an effective pathway to language-grounded decoding.***

4 Method

In this section, we first elaborate on the construction of our step-wise *Grounded Latent Supervision Dataset (GLSD)*, which provides spatial-semantic supervision for intermediate visual reasoning. Built on this, we introduce ***RIS***, a grounded latent reasoning framework that uses last-layer hidden states as continuous latent tokens. Following common practice, ***RIS*** allocates a fixed number of latent tokens and feeds them forward as continuous embeddings. Figure 2 illustrates the overall training pipeline.

4.1 Dataset Construction

Since existing datasets rarely provide fine-grained supervision for intermediate reasoning, we convert GQA(Hudson & Manning, 2019) data into a unified step-wise grounded format through a two-stage pipeline. First, prompt an MLLM to decompose each question-answer pair into a 2–4 step trace, where each step contains an operation tag, a region-specific semantic description, visual information, and preliminary bounding-box coordinates, leading to a full answer (as transition tokens) and final answer as visualized in Appendix G. We then feed the generated descriptions and original image into Grounding DINO (Liu et al., 2023) to verify and calibrate the MLLM-proposed coordinates.

The verified trace is serialized as $\mathcal{T}$ and padded to the fixed latent budget K. Each supervised step is represented as `<step_start>` `<bbox>` b_i `</bbox>` v_i `<step_end>`, where b_i denotes the normalized bounding box coordinates and v_i denotes the corresponding visual information. Let $m_i \in \{0, 1\}$ indicate whether the i-th slot has explicit step-wise supervision. For unsupervised slots with $m_i = 0$, we insert `<step_start> [synthesize] <step_end>`, allowing them to learn evidence integration from downstream answer supervision. This yields $96k$ training samples.

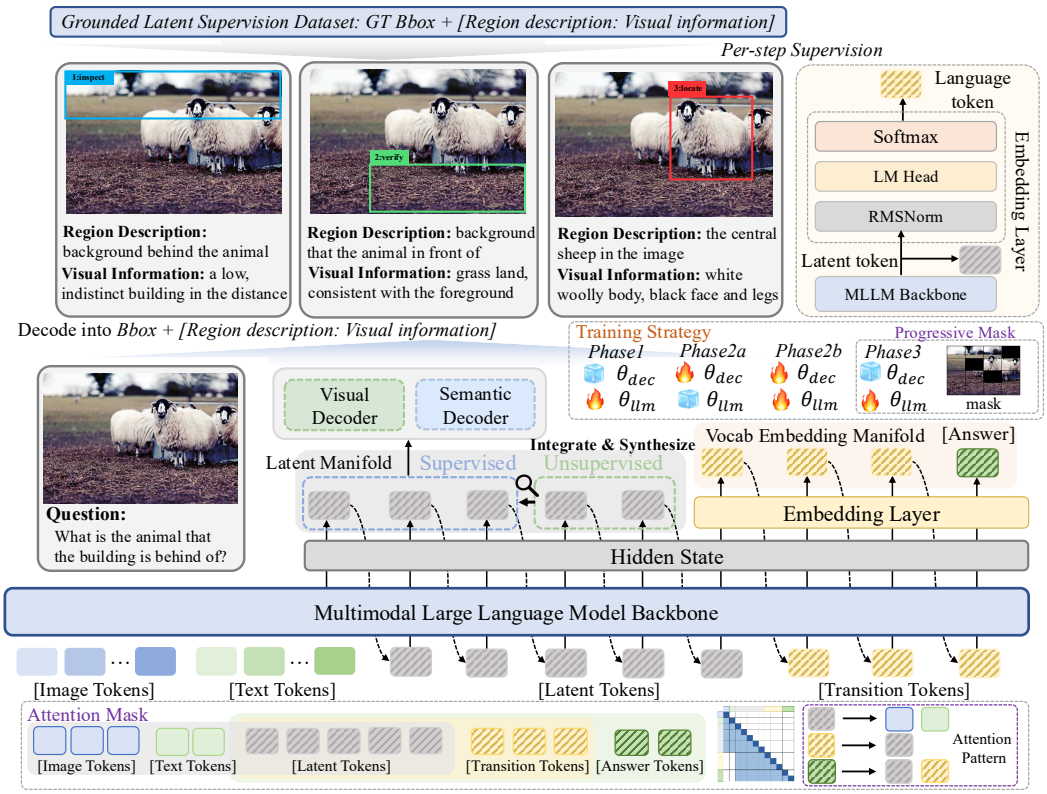


Figure 2: Overview of ***RIS*** framework. Visual and semantic decoders are only used for supervision in training and will be removed during inference. Full attention mask used during inference.

4.2 Training Phase 1: Explicit Grounded Reasoning Initialization

In Phase 1, we initialize the MLLM with explicit grounded visual reasoning before moving to continuous latent computation. Given image embeddings $\mathcal{X}_v$ and a tokenized query $\mathcal{X}_q$, the model is trained to generate the structured reasoning trace $\mathcal{T}$ followed by the *full answer* A_s and final answer A. Unlike generic textual rationales, $\mathcal{T}$ contains step-wise normalized bounding-box coordinates, fine-grained visual information, and boundary tokens, thereby teaching the model to retrieve regions, integrate visual evidence, and synthesize the answer in a discrete but explicitly grounded format.

We optimize the backbone $\mathcal{M}$ with the standard next-token prediction (NTP) objective:

$$\mathcal{L}_{\text{NTP}} = -\sum_{t=1}^{|\mathcal{T}\oplus A_s\oplus A|} \log P_{\mathcal{M}}\left(y_t \mid y_{<t}, \mathcal{X}_v, \mathcal{X}_q\right). \tag{1}$$

This stage equips the backbone with a reliable text-space grounded reasoning prior. After training, we use the Phase-1 trained model to encode the region-specific descriptions in each step. The resulting cached embeddings serve as stable semantic anchors for Phase 2 training, preventing the latent-space supervision targets from drifting during continuous-token training.

4.3 Training Phase 2a: Side-Head Grounding and Semantic Alignment

To transition from discrete textual reasoning to the continuous latent space, we introduce a set of fixed-capacity visual latent tokens $\mathcal{C} = \{c_1, c_2, \ldots, c_K\}$. In this phase, we freeze the MLLM backbone and calibrate two lightweight, task-specific side heads: *Visual Grounding Decoder* (f_{reg}) and *Semantic Alignment Decoder* (f_{desc}). For each reasoning step t where $m_t = 1$, we extract the last-layer hidden state z_t at the `<step_start>` token in the explicit reasoning trace, which summarizes the context before generating the step content. The two decoders then operate as explicit supervision signals:

Spatial Grounding: f_{reg} predicts the bounding box coordinates $\hat{b}_t = f_{\text{reg}}(z_t)$. We optimize this using a combination of ℓ_1 distance and Generalized IoU (GIoU) loss against the verified boxes b_t:

$$\mathcal{L}_{\text{reg}} = \frac{1}{\sum_{t=1}^{K} m_t} \sum_{t=1}^{K} m_t \Big(||\hat{b}_t - b_t||_1 + \lambda_{\text{GIoU}} \mathcal{L}_{\text{GIoU}}(\hat{b}_t, b_t) \Big) \tag{2}$$

Semantic Anchoring: To prevent the latent representation from drifting into uninterpretable regions, f_{desc} projects z_t into a semantic embedding space. We minimize the cosine distance between the projected vector and the pre-computed text embeddings e_t of the corresponding region description:

$$\mathcal{L}_{\text{desc}} = \frac{1}{\sum_{t=1}^{K} m_t} \sum_{t=1}^{K} m_t \Big(1 - \cos(f_{\text{desc}}(z_t), e_t) \Big) \tag{3}$$

The total loss for this warm-up phase is $\mathcal{L}_{\text{Phase2a}} = \lambda_r \mathcal{L}_{\text{reg}} + \lambda_d \mathcal{L}_{\text{desc}}$. This supervised calibration imposes spatial grounding to specify *where to look*, while semantic anchoring to specify *what is seen*. These calibrated side heads provide stable spatial-semantic constraints for converting explicit reasoning steps into grounded latent tokens in the subsequent phase.

4.4 Training Phase 2b: Progressive Latent Internalization

With the side heads calibrated, Phase 2b progressively converts explicit textual reasoning into continuous latent computation. Following the *Coconut* curriculum paradigm (Hao et al., 2024), we use a step-wise replacement schedule $s \in \{1, \ldots, K\}$: at stage s, the first s textual reasoning blocks are replaced by their corresponding latent states $\mathcal{C}_{\leq s}$, while the remaining textual steps $\mathcal{T}_{>s}$ and the *full answer* A_s and final answer A are still generated autoregressively as discrete text. Unlike Phase 2a, the MLLM backbone is unfrozen, allowing its internal computation to adapt to latent inputs.

The model is trained with next-token prediction on the remaining textual trace and final answer:

$$\mathcal{L}_{\text{cot}} = -\sum_{t=1}^{|\mathcal{T}_{>s}|} \log P_{\mathcal{M}}(y_t \mid y_{<t}, \mathcal{X}_v, \mathcal{X}_q, \mathcal{C}_{\leq s}), \quad \mathcal{L}_{\text{ans}} = -\sum_{t=1}^{|A_s \oplus A|} \log P_{\mathcal{M}}(a_t \mid a_{<t}, \mathcal{X}_v, \mathcal{X}_q, \mathcal{C}_{\leq s}, \mathcal{T}_{>s}) \tag{4}$$

To preserve grounding during internalization, the side-head losses are applied only to supervised latent slots that have been replaced, i.e., $\{t \leq s \mid m_t = 1\}$. The overall objective is

$$\mathcal{L}_{\text{Phase2b}} = \mathcal{L}_{\text{ans}} + \alpha \mathcal{L}_{\text{cot}} + \lambda_r \mathcal{L}_{\text{reg}} + \lambda_d \mathcal{L}_{\text{desc}} \tag{5}$$

As s increases, information previously expressed by text is gradually internalized into continuous latent states. By the end of this curriculum, the model learns to retrieve and transform grounded visual evidence in latent space with reduced dependence on explicit verbalization.

4.5 Training Phase 3: Bottlenecked Latent Integration

In Phase 3, all explicit reasoning steps are replaced by latent tokens, so the intermediate reasoning process is fully mediated by the latent tokens $\mathcal{C}$. To prevent answer decoding from bypassing these tokens and directly attending to the original image and query tokens $(\mathcal{X}_v, \mathcal{X}_q)$, we introduce a *Progressive Attention Mask*. Specifically, we anneal a masking probability $\rho(\tau)$ over training step τ and sample $M \sim \text{Bernoulli}(\rho(\tau))$ to modulate the causal attention matrix. As $\rho(\tau)$ increases, answer tokens are gradually forced to rely on $\mathcal{C}$, making latent states the information conduit for task-relevant visual and semantic evidence.

Since no textual reasoning steps remain in this phase, the textual CoT loss is removed. The model is trained with the final-answer objective, together with side-head constraints on supervised latent slots:

$$\mathcal{L}_{\text{Phase3}} = \mathcal{L}_{\text{ans}} + \lambda_r \mathcal{L}_{\text{reg}} + \lambda_d \mathcal{L}_{\text{desc}} \tag{6}$$

Under the fully masked condition, the answer loss is conditioned on the latent tokens:

$$\mathcal{L}_{\text{ans}} = -\sum_{t=1}^{|A_s \oplus A|} \log P_{\mathcal{M}}\left(u_t \mid u_{<t}, \mathcal{C}\right), \tag{7}$$

Model	V*			HRBench4K			HRBench8K			MMVP	BLINK
	Overall	Attribute	Spatial	Overall	FSP	FCP	Overall	FSP	FCP		
Proprietary Model											
GPT-4o	65.15	69.68	58.39	54.70	64.93	44.51	51.12	57.08	45.12	72.00	63.55
Open-Source Model											
Qwen2.5-VL-7B	76.65	77.12	74.35	68.30	80.60	56.03	64.33	74.42	54.24	63.49	54.94
Qwen2.5-VL-7B+***GLSD***	78.25	78.39	78.11	70.58	83.30	57.86	64.70	74.69	54.70	66.80	56.25
LVR(Li et al., 2025)	80.60	83.26	77.94	70.88	83.25	57.50	63.50	75.00	52.00	71.03	56.79
Monet(Wang et al., 2025a)	81.90	81.94	81.86	69.97	84.01	55.93	67.01	78.59	55.43	70.00	56.70
COVT(Qin et al., 2025)	79.10	81.05	77.15	71.90	85.50	58.30	68.40	**79.30**	57.50	58.70	57.40
Our Model											
RIS	**83.75**	**84.26**	**83.24**	**73.23**	**86.33**	60.12	**68.52**	79.05	**57.98**	73.55	60.60
RIS+VLPO**(Wang et al., 2025a)**	81.76	81.24	82.28	71.79	82.83	**60.75**	63.05	72.67	53.42	**73.76**	**60.95**
Relative Improvement	+7.10	+7.14	+8.89	+4.93	+5.73	+4.09	+4.19	+4.63	+3.74	+10.27	+5.66

Table 1: Main results on visual reasoning benchmarks across proprietary, open-source, and latent visual reasoning baselines (with Qwen2.5-VL-7B as the same backbone) with **5 latent tokens**.

where A_s and A denote the *full answer* bridge and the final answer, u_t indexes concated tokens.

Under this information bottleneck, the supervised latent tokens remain spatially and semantically grounded through the side-head losses, while the free tokens, which receive no direct spatial-semantic supervision, are optimized only through the answer objective and are therefore encouraged to aggregate and synthesize the evidence retrieved by earlier grounded tokens.

Finally, to facilitate a smooth return from the latent visual manifold to the discrete vocabulary manifold $\mathcal{M}_{vocab}$, we utilize the short *full answer* A_s (as defined in Phase 1) as a sequence of *Manifold Transition Tokens*. Instead of forcing an abrupt jump from synthesized latent states to final answer decoding, A_s provides an autoregressive intermediate path that gradually maps latent visual representations back toward the pretrained vocabulary-aligned manifold $\mathcal{M}_{vocab}$. Its dense next-token supervision stabilizes this transition and improves the compatibility between latent visual reasoning and language-grounded answer generation.

5 Experiments

5.1 Experiment Setup

Training and Evaluation Setup. We adopt Qwen2.5-VL-7B(Bai et al., 2025) as our base model. The training process follows our three-phase pipeline and all parameters are detailed in Appendix B.

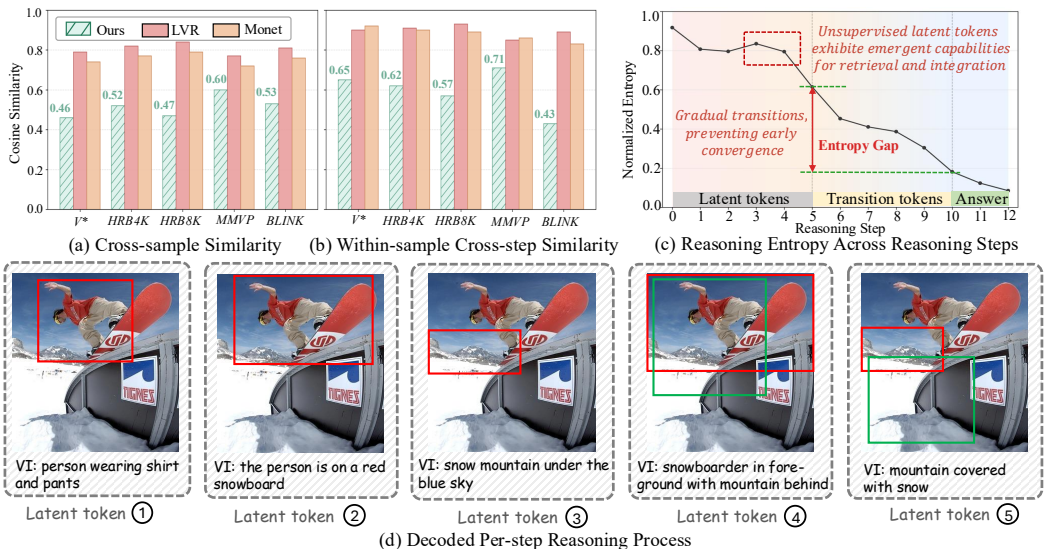
Evaluated Benchmarks. To comprehensively evaluate our method, we conduct experiments on a diverse set of perception and reasoning benchmarks: V*(Wu & Xie, 2024), HRBench4K(Wang et al., 2025b), HRBench8K(Wang et al., 2025b), MMVP(Tong et al., 2024), and BLINK(Fu et al., 2024).

Baselines. We compare ***RIS*** against a variety of baselines: (1) *Proprietary Models:* GPT-4o; (2) *Open-Source Base Model:* Qwen2.5-VL-7B; (3) *Vanilla SFT:* Qwen2.5-VL-7B+GLSD, which finetunes the base model on our curated *Grounded Latent Supervision Dataset (GLSD)* for same training steps as ***RIS***; (4) *Latent Visual Reasoning Methods:* LVR(Li et al., 2025), Monet(Wang et al., 2025a), and CoVT(Qin et al., 2025). Furthermore, to investigate the benefit of reinforcement learning on our method, we introduce ***RIS+VLPO***, a variant that further optimizes our model using *Visual-latent Policy Optimization (VLPO)* proposed by Monet.

5.2 Main Results

Table 1 summarizes the performance of ***RIS*** and baselines across five visual reasoning benchmarks. Overall, ***RIS*** consistently outperforms both the open-source backbone and existing latent visual reasoning baselines. The GLSD baseline, which retains explicit textual reasoning, yields much smaller gains. This indicates that the improvements are not only due to extra supervision but mainly stem from internalizing grounded visual evidence into latent states and performing latent reasoning. Although ***RIS+VLPO*** does not consistently yield further gains, it remains a promising direction for adaptive and stable latent computation, while a reliable latent variant is still underexplored.

The performance pattern further aligns with the characteristics of each benchmark. ***RIS*** achieves larger gains on benchmarks requiring precise localization, structured visual search, and multi-step perceptual reasoning, such as V* and BLINK. The detailed BLINK results in Table 2 show particularly great



(a) Cross-sample Similarity (b) Within-sample Cross-step Similarity (c) Reasoning Entropy Across Reasoning Steps

(d) Decoded Per-step Reasoning Process

Figure 3: Latent Behavior Analysis of ***RIS***: Diversity, Reasoning Entropy, and Interpretability.

improvements on tasks closely tied to visual grounding and sequential evidence retrieval, including *Spatial Reasoning*, *Object Localization*, and *Relational Reasoning*. In contrast, on MMVP, where performance is more constrained by the visual encoder's ability, the gains are more moderate.

5.3 Analysis on Latent Behaviors

Impact of Latent Token Budget. Figure 4(c) studies the effect of latent token budget. Performance does not improve monotonically with more latent tokens, as most training samples contain only three to four supervised reasoning steps; additional slots receive no spatial-semantic supervision and are optimized only through the final-answer objective. Their utility is therefore task-dependent: a few extra tokens benefit multi-step visual exploration on V* and BLINK by integrating retrieved evidence, whereas larger budgets introduce more unsupervised states and reduce grounding reliability. The degradation is more pronounced on HRBench8K and MMVP, which depend respectively on precise local evidence and visual encoder capability. Overall, the latent budget should align with the available step-wise supervision.

Model	S.R.	O.L.	R.R.	Counting	R.D.
Base	86.81	43.62	39.87	68.30	67.26
Base+***GLSD***	87.03	49.57	41.13	67.59	65.66
LVR	86.01	50.82	43.28	69.17	76.61
Monet	85.31	45.08	39.55	70.83	75.81
CVOT	87.41	53.20	38.06	65.83	**77.65**
RIS	**89.60**	**54.63**	**44.25**	**72.02**	76.50
Improvement	+2.79	+11.01	+4.38	+3.72	+9.24

Table 2: Performance on *BLINK*. *(S.R.: Spatial Reasoning, O.L.: Object Localization, R.R.: Relative Reflectance, R.D.: Relative Depth).*

Entropy Dynamics of Latent Reasoning. Figure 3(c) shows the normalized entropy along the reasoning trajectory, with details in Appendix D. Supervised latent tokens maintain high entropy, indicating that grounded reasoning preserves multiple plausible visual hypotheses rather than collapsing prematurely. Entropy slightly increases in unsupervised slots, consistent with their role in integrating evidence from earlier grounded states. Transition tokens then gradually reduce entropy toward answer decoding, bridging high-entropy latent reasoning and the low-entropy vocabulary space. This supports the intended computation flow: visual evidence is explored and integrated in latent space before being mapped back to language decoding.

Latent Token Diversity and Interpretability. Figure 3(a)(b) examines whether latent representations collapse into generic patterns. Across both cross-sample and within-sample comparisons, ***RIS*** exhibits substantially lower similarity than LVR and Monet, indicating more instance-specific and step-dependent latent states. Although within-sample similarity is naturally higher due to shared visual context, the latent trajectory remains diverse, particularly on BLINK, where repeated visual search is required. Figure 3(d) provides complementary qualitative evidence: decoded regions form

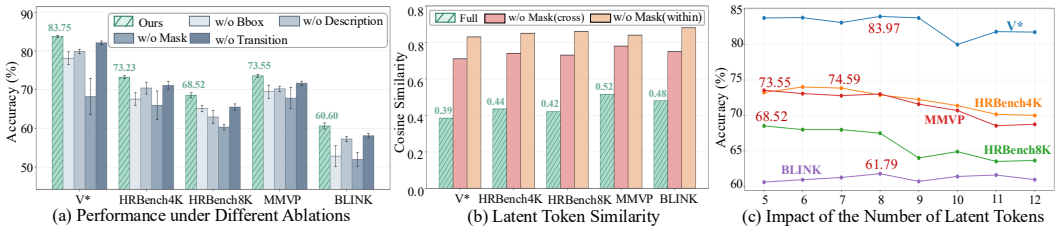


Figure 4: Design Ablations. (Error bars denote accuracy standard deviation across repeated training runs.)

a progressive reasoning trajectory, while later synthesis tokens cover multiple relevant regions and integrate their visual information. Together, these results suggest that ***RIS*** organizes grounded evidence progressively rather than repeatedly encoding similar features.

Causal Dependence on Latent States. We further test whether latent states are functionally used for answer through inference-time interventions that perturb only the latent pathway. Removing latent states, blocking latent-to-answer access, shuffling states across or within examples, or replacing them with position-wise means or moment-matched noise consistently reduces average accuracy from 71.93 to 64.40–67.13. In particular, removing latent states and blocking their access reduce accuracy to 66.29 and 66.67, while cross- and within-example shuffling decrease it to 64.40 and 64.85. These results show that ***RIS*** depends not only on additional latent computation, but also on instance-specific latent content and its sequential organization. Full results are provided in Appendix E.

5.4 Ablation Study

To assess the contribution of each component in ***RIS***, we conduct systematic ablations, with results summarized in Figure 4 leading to three critical takeaways regarding latent visual reasoning.

Spatial-semantic supervision stabilizes grounded latent reasoning. Removing either side-head supervision degrades performance and increases variance, confirming the importance of grounding. Bounding-box supervision is especially important on localization-heavy benchmarks such as V* and BLINK, showing that spatial anchoring is crucial for directing latent tokens to task-relevant visual evidence. Description supervision has a milder but consistent impact, suggesting that semantic alignment regularizes the latent space and prevents drift from the vocabulary reasoning manifold.

The progressive attention mask prevents latent bypassing and collapse. Removing the progressive attention mask causes the largest performance drop. Without this bottleneck, answer tokens can directly attend to the raw image and question, weakening the role of latent tokens. The degradation further suggests that uncalibrated latent tokens are not merely ignored placeholders; they can interfere with prediction even when answer tokens still access the original inputs. This is further supported by Figure 4(b), removing the mask also sharply increases both cross-sample and within-sample latent-token similarity, revealing collapse toward generic representations. Thus, the mask is essential for routing visual evidence through the latent trajectory and maintaining step-specific latent computation.

Transition tokens facilitate language-grounded decoding. Removing transition tokens also consistently hurts performance, indicating that the short-answer bridge is important for mapping synthesized latent evidence back to language-grounded decoding. This proves that these tokens reduce representation mismatch between the latent visual manifold and the vocabulary-aligned output space.

6 Conclusion

In this work, we propose ***RIS***, a spatial-semantic grounded latent visual reasoning framework that enables MLLMs to reason over visual evidence within continuous latent states. We show that existing latent visual reasoning methods suffer from weak grounding, trajectory collapse, and answer shortcuts. To address these issues, ***RIS*** anchors latent tokens with spatial-semantic supervision, enforces their use through a progressive attention bottleneck, and uses transition tokens to bridge latent reasoning to language decoding. Extensive experiments demonstrate consistent improvements over strong baselines. Further analyses verify that ***RIS*** learns more diverse, interpretable, and causally effective latent trajectories, suggesting a practical path to faithful internal visual reasoning in MLLMs.

7 AI USE STATEMENT

We used LLMs solely for two purposes: (1) to assist with and polish the writing of the manuscript, including improving clarity, grammar, and readability; and (2) to support retrieval and literature discovery, such as identifying potentially relevant prior work. (3) Generate our training dataset. All AI-assisted text and retrieved references were reviewed and verified. We take full responsibility for the final content of the paper.

8 REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide detailed descriptions of the proposed method and experimental setup in the main paper and appendix. We have also submitted the source code as supplementary material, including the implementation of our method and the necessary scripts for reproducing the main experimental results. Together, these materials are intended to provide sufficient information for reproducing and verifying our findings.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Jingcheng Deng, Liang Pang, Zihao Wei, Shicheng Xu, Zenghao Duan, Kun Xu, Yang Song, Huawei Shen, and Xueqi Cheng. Llm latent reasoning as chain of superposition, 2026. URL https://arxiv.org/abs/2510.15522.

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

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14953–14962, 2023.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

Yushi Hu, Otilia Stretcu, Chun-Ta Lu, Krishnamurthy Viswanathan, Kenji Hata, Enming Luo, Ranjay Krishna, and Ariel Fuxman. Visual program distillation: Distilling tools and programmatic reasoning into vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9590–9601, 2024.

Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6693–6702, 2019. URL https://api.semanticscholar.org/CorpusID:152282269.

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

Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Hao Chen, Emad Barsoum, Muhao Chen, and Zicheng Liu. Latent visual reasoning. *arXiv preprint arXiv:2509.24251*, 2025.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

You Li, Chi Chen, Yanghao Li, Fanhu Zeng, Kaiyu Huang, Jinan Xu, and Maosong Sun. Imagination helps visual reasoning, but not yet in latent space. *arXiv preprint arXiv:2602.22766*, 2026.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, et al. Llava-plus: Learning to use tools for creating multimodal agents. In *European conference on computer vision*, pp. 126–142. Springer, 2024.

Yiming Qin, Bomin Wei, Jiaxin Ge, Konstantinos Kallidromitis, Stephanie Fu, Trevor Darrell, and XuDong Wang. Chain-of-visual-thought: Teaching vlms to see and think better with continuous visual tokens. *arXiv preprint arXiv:2511.19418*, 2025.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. Codi: Compressing chain-of-thought into continuous space via self-distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 677–693, 2025.

Zhaochen Su, Peng Xia, Hangyu Guo, Zhenhua Liu, Yan Ma, Xiaoye Qu, Jiaqi Liu, Yanshu Li, Kaide Zeng, Zhengyuan Yang, et al. Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers. *arXiv preprint arXiv:2506.23918*, 2025.

Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11888–11898, 2023.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9568–9578, 2024.

Qixun Wang, Yang Shi, Yifei Wang, Yuanxing Zhang, Pengfei Wan, Kun Gai, Xianghua Ying, and Yisen Wang. Monet: Reasoning in latent visual space beyond images and language. *arXiv preprint arXiv:2511.21395*, 2025a.

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7907–7915, 2025b.

Yaoting Wang, Shengqiong Wu, Yuecheng Zhang, Shuicheng Yan, Ziwei Liu, Jiebo Luo, and Hao Fei. Multimodal chain-of-thought reasoning: A comprehensive survey. *arXiv preprint arXiv:2503.12605*, 2025c.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13084–13094, 2024. doi: 10.1109/CVPR52733.2024.01243.

Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. *arXiv preprint arXiv:2506.17218*, 2025.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023.

Xinlei Yu, Zhangquan Chen, Yongbo He, Tianyu Fu, Cheng Yang, Chengming Xu, Yue Ma, Xiaobin Hu, Zhe Cao, Jie Xu, et al. The latent space: Foundation, evolution, mechanism, ability, and outlook. *arXiv preprint arXiv:2604.02029*, 2026.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023.

A LIMITATIONS

Although ***RIS*** provides an effective framework for grounded latent visual reasoning, it still has several limitations.

First, our framework involves multiple interacting components. While each component is empirically essential, the overall framework introduces several hyperparameters and schedules whose optimal settings may depend on the backbone model, task type, and data distribution. A more principled and automatic strategy for configuring these components would further improve the robustness and usability of our framework.

Second, ***RIS*** still fixes the latent token budget, which is a common setting of most existing latent reasoning methods. In our framework, the supervised latent tokens and unsupervised latent tokens provide an implicit form of adaptive computation, and our analyses suggest that different latent slots can spontaneously specialize into retrieval, integration, and synthesis roles. Nevertheless, the total number of latent slots is still predefined rather than dynamically determined according to the complexity of each input. Developing latent reasoning models with truly adaptive token allocation remains an important direction of our future work.

Third, our experiments mainly focus on standard image-based VQA and visual reasoning benchmarks commonly used for latent visual reasoning. Although these benchmarks cover localization, structured visual search, and multi-step perceptual reasoning, they do not fully reflect the diversity of real-world multimodal reasoning scenarios. Future work would further evaluate and extend ***RIS*** to broader settings, such as open-ended long-form visual question answering, multi-image reasoning, video reasoning, and interactive embodied tasks.

Finally, GLSD relies on grounded reasoning traces generated by a single strong teacher model. Although Grounding DINO helps calibrate the spatial annotations, errors in the generated descriptions or reasoning steps may still affect supervision quality. The traces may also inherit teacher-specific preferences in reasoning decomposition and evidence ordering, limiting the diversity of visual reasoning strategies represented in the dataset. Exploring multiple teachers or grounding supervision less dependent on teacher-generated traces is a promising direction for future work.

B TRAINING HYPERPARAMETERS

Table 3: Detailed Hyperparameters for ***RIS*** Training.

Training Phase	**Hyperparameter**	**Value**
Architecture	Base Model	Qwen2.5-VL-7B
	Latent Tokens (K)	5
Phase 1	Epochs	1 – 2
	Learning Rate	1×10^{-5}
Phase 2	Sub-stage 2.A LR (f_{reg})	5×10^{-4}
	Sub-stage 2.A LR (f_{desc})	1×10^{-4}
	Sub-stage 2.B Epochs per Curriculum Stage	1 – 2
	Sub-stage 2.B LR	5×10^{-6}
Phase 3	Epochs	1 – 2
	Learning Rate	3×10^{-6}
	Mask Ratio Annealing	$0.3 \rightarrow 1.0$
Loss Weights	λ_r (Region Grounding)	1.0
	λ_{giou} (GIoU within Region Grounding)	2.0
	λ_d (Region Description Alignment)	0.5
	α (Textual CoT)	1.0

Table 4: Additional hyperparameters for the ***RIS***+VLPO reinforcement learning stage.

Hyperparameter	Value
Initialization	Phase-3 ***RIS*** checkpoint
RL data	3.2k/**6.4k** samples from GLSD
Latent action	***RIS*** latent slot hidden state c_t
Latent tokens	$K = 5$
Attention mask	Phase-3 bottleneck mask, fixed at $\rho = 1.0$
Reference model	Frozen Phase-3 ***RIS*** checkpoint
Rollouts per prompt	$G = 4$
Policy epochs per batch	1
RL epochs	1
Global prompt batch size	32
Learning rate	1×10^{-6}
KL coefficient	$\beta = 0.02$
Latent likelihood scale	$\sigma = 1.0$
Max response length	128 tokens
Sampling temperature / top-p	1.0 / 0.95
Reward	$r_{\text{acc}} + 0.1 r_{\text{fmt}}$
Accuracy reward r_{acc}	1 if normalized answer matches ground truth, else 0
Format reward r_{fmt}	1 if final answer is extractable, else 0
Latent-use reward	None
Grounding regularization	$0.1\mathcal{L}_{\text{reg}} + 0.05\mathcal{L}_{\text{desc}}$
Trainable modules	MLLM backbone; side heads frozen

Table 5: Recorded training time of different stages on 4 NVIDIA A100 80GB GPUs.

Training Stage	**Trainable Components**	**Update Steps**	**Wall-clock Time**
Phase 1	Backbone	2.5K	2.5–3.0 h
Phase 2a	Side heads	2.5K	2.0–2.5 h
Phase 2b	Backbone + side heads	12.5K	16–20 h
Phase 3	Backbone + side heads	2.5K	6.0–7.5 h
Total	–	20.0K	26.5–33 h

C Compute Resources

All experiments were conducted on a single server equipped with an Intel Xeon Platinum 8383C CPU, 512GB DDR5 RAM, and 4 NVIDIA A100 80GB GPUs connected with NVLink. We trained on an 80k-sample *Grounded Latent Supervision Dataset* (GLSD) with a global batch size of 32, corresponding to 2,500 optimization steps for each full data pass. Table 5 reports the estimated wall-clock training time for each stage of ***RIS***. For Phase 2b, we assume a latent budget of $K = 5$ and one full data pass for each curriculum stage; its cost scales approximately linearly with K.

D Details of Reasoning Entropy Estimation

We provide a detailed definition and implementation of *Reasoning Entropy* in this section. For latent tokens, we consider two possible estimators. The first estimator, *Semantic Region Similarity as Reasoning Entropy*, offers a direct and strict way to quantify how much grounded visual evidence is encoded in each latent token by measuring its compatibility with multiple semantic evidence regions. However, it depends on the semantic decoder f_{desc} calibrated for latent states and cannot be directly applied to language tokens due to their next-token prediction nature. Therefore, we use it only as a reference analysis for latent tokens.

The second estimator, *Intervention-based Visual Evidence Importance as Reasoning Entropy*, places latent tokens and language tokens in the same space. It therefore enables a unified computation and comparison of *Reasoning Entropy* across latent tokens, transition tokens, and answer tokens. In practice, we found that the entropy of latent tokens computed by the two estimators exhibits highly

consistent trends. Therefore, unless otherwise specified, we adopt the intervention-based estimator as the unified definition of *Reasoning Entropy* throughout our analysis.

D.1 Dataset Preparation

For each image–question pair (I, q), we construct an analysis-only bank of diverse grounded reasoning traces. Following the same data construction protocol used for our step-wise grounded supervision, we prompt a strong MLLM to sample multiple visual reasoning traces. Each trace contains a sequence of grounded evidence steps, and each step consists of a bounding box, a region-specific semantic description, and the corresponding visual information. We retain only traces whose final answers match the ground-truth answer and whose grounded boxes pass the verification procedure. This yields an image-question-specific trace bank:

$$\mathcal{T}(I,q) = \left\{ \mathcal{T}_m = \{(b_{m,s}, d_{m,s}, v_{m,s})\}_{s=1}^{S_m} \right\}_{m=1}^{M},$$

where m indexes a valid reasoning trace, s indexes an evidence step, $b_{m,s}$ is the verified bounding box, $d_{m,s}$ is the region description, and $v_{m,s}$ is the extracted visual information.

We encode each evidence step into the same semantic space used by the semantic alignment decoder:

$$e_{m,s} = g\left(d_{m,s} \oplus v_{m,s}\right),$$

where $g(\cdot)$ denotes the frozen semantic encoder used to build stable region-level semantic anchors, and $\oplus$ denotes textual concatenation. The resulting set

$$\mathcal{B}(I,q) = \{e_{m,s}\}_{m=1,s=1}^{M,S_m}$$

serves as an image-question-specific bank of answer-relevant visual evidence.

D.2 Semantic Region Similarity as Reasoning Entropy

Probing latent token states in the semantic space. For a latent visual reasoning model, we collect the last-layer hidden states along the latent trajectory, including both supervised and unsupervised latent tokens. For a hidden state h_t, we use the calibrated semantic alignment decoder as a probing interface:

$$d_t = f_{\text{desc}}(h_t).$$

We compute the similarity between d_t and each evidence step in $\mathcal{B}(I,q)$, and normalize the similarities into an evidence-step distribution:

$$p_{m,s}^{(t)} = \frac{\exp\left(\cos(d_t, e_{m,s})/\tau\right)}{\sum_{m'=1}^{M}\sum_{s'=1}^{S_{m'}} \exp\left(\cos(d_t, e_{m',s'})/\tau\right)},$$

where τ is a temperature parameter.

Normalized reasoning entropy. Our goal is to measure whether a latent state remains compatible with multiple plausible grounded reasoning traces, we aggregate the evidence-step distribution into a trace-level distribution:

$$P_m^{(t)} = \sum_{s=1}^{S_m} p_{m,s}^{(t)}.$$

The reasoning entropy of token t is then defined as

$$H_{\text{reason}}(h_t) = -\sum_{m=1}^{M} P_m^{(t)} \log P_m^{(t)}.$$

To make values comparable across samples with different numbers of retained valid traces, we report the normalized reasoning entropy:

$$\widetilde{H}_{\text{reason}}(h_t) = \frac{H_{\text{reason}}(h_t)}{\log M}.$$

Thus, $\widetilde{H}_{\text{reason}}(h_t) \in [0, 1]$. A higher value indicates that the hidden state remains semantically compatible with multiple valid grounded reasoning traces, while a lower value indicates that the state has concentrated on a smaller set of reasoning possibilities.

This probing analysis should not be interpreted as decoding multiple bounding boxes from one latent token. The visual grounding decoder f_{reg} still predicts a single supervised bounding box for each grounded latent token. The entropy is instead estimated through the semantic alignment decoder f_{desc}, which probes whether the hidden state is semantically close to multiple valid grounded reasoning traces beyond its explicitly decoded spatial output.

D.3 Intervention-based Visual Evidence Importance as Reasoning Entropy

The semantic probing entropy above measures whether a hidden state is compatible with multiple grounded reasoning traces in the learned semantic evidence space. Although more intuitive, this probing interface relies on the calibrated semantic decoder f_{desc}, which is trained on latent states and is therefore not directly comparable for ordinary language states. Another possible choice is to compute entropy from the language-modeling distribution, i.e.,

$$H_{\text{vocab}}(t) = -\sum_{v \in \mathcal{V}} p_\theta(v \mid u_t) \log p_\theta(v \mid u_t),$$

where $p_\theta(v \mid u_t)$ is obtained by applying the LM head and softmax to the hidden state. However, this quantity measures next-token prediction uncertainty over the vocabulary rather than the amount of visual evidence represented by the current state. For example, a transition token may have low vocabulary entropy simply because the next word is syntactically predictable, even if its hidden state still depends on multiple visual regions. Conversely, function words or ambiguous lexical continuations may yield high vocabulary entropy without indicating broad visual grounding. Moreover, vocabulary entropy is not naturally defined for non-linguistic latent slots, making it unsuitable for comparing latent tokens and language tokens in a shared representational space.

To fairly compare latent tokens, transition tokens, and answer tokens, we therefore estimate entropy through an intervention-based visual evidence distribution, which does not require any additional projection head or vocabulary-level decoding. This estimate evaluates all analyzed states in the same image-question-specific evidence space by measuring how their representations change when each grounded visual evidence region is removed.

For each image–question pair (I, q), we use the grounded evidence bank $\mathcal{B}(I, q)$ defined above. Each evidence node corresponds to a verified region b_k from the retained grounded traces, where $k \in \{1, \ldots, K_{I,q}\}$. To avoid artificially increasing entropy due to repeated boxes across different traces, we merge highly overlapping evidence regions using non-maximum suppression and keep the merged regions as the intervention units. Let u_t denote the last-layer hidden state at trajectory position t, where u_t can be a supervised latent token, an unsupervised latent token, or to generate a transition token or an answer token. The model first inferences on the original image and record the full-state representation u_t^{full}.

We then construct one counterfactual input for each evidence node by masking the corresponding image region b_k, producing $I^{(-k)}$. The same question, latent-token layout, transition tokens, and answer tokens are used under teacher forcing, so that token positions are aligned across the original and counterfactual forward passes. This yields a counterfactual representation $u_t^{(-k)}$ for every analyzed state, and the difference between u_t^{full} and $u_t^{(-k)}$ reflects the effect of removing visual evidence region b_k, rather than changes in the generated token sequence. The visual sensitivity of state u_t to evidence node k is defined as

$$s_{t,k} = \max\left(0,\, 1 - \cos\left(\text{LN}(u_t^{\text{full}}), \text{LN}(u_t^{(-k)})\right)\right),$$

where $\text{LN}(\cdot)$ denotes the same final hidden-state normalization used before the language modeling head. A larger $s_{t,k}$ indicates that removing region b_k induces a larger change in the token state, suggesting that the state depends more strongly on this visual evidence node.

We normalize the sensitivities over all evidence nodes to obtain an interventional visual evidence distribution:

$$q_{t,k} = \frac{s_{t,k} + \epsilon}{\sum_{r=1}^{K_{I,q}} (s_{t,r} + \epsilon)},$$

where ϵ is a small constant for numerical stability. The corresponding evidence entropy is

$$H_{\mathrm{IVE}}(u_t) = -\sum_{k=1}^{K_{I,q}} q_{t,k} \log q_{t,k}.$$

Since different samples may contain different numbers of retained evidence nodes, we report normalized entropy:

$$\widetilde{H}_{\mathrm{IVE}}(u_t) = \frac{H_{\mathrm{IVE}}(u_t)}{\log K_{I,q}}.$$

A potential issue is that visually inactive tokens, such as function words or punctuation, may have uniformly small sensitivities to all evidence regions. Such tokens can obtain spuriously high normalized entropy after normalization. We therefore compute the total visual sensitivity mass

$$M_t = \sum_{k=1}^{K_{I,q}} s_{t,k},$$

and use a mass-aware entropy score:

$$\widehat{H}_{\mathrm{IVE}}(u_t) = \frac{M_t}{M_t + \alpha} \cdot \widetilde{H}_{\mathrm{IVE}}(u_t),$$

where α is set to the median visual sensitivity mass over all analyzed states on the validation subset. This weighting preserves high entropy only when the token state is both visually grounded and broadly influenced by multiple evidence nodes.

Trajectory-level aggregation. We compute $\widehat{H}_{\mathrm{IVE}}$ for all states along the latent-to-answer trajectory. Fixed latent slots are averaged by slot index across samples. Since the number of transition and answer tokens may vary across examples, we align textual positions by their normalized phase position and average them into fixed-width bins. Special tokens and padding tokens are excluded. The plotted curve reports the mean entropy over samples, with the latent phase, transition-token phase, and answer-token phase shown in order.

This intervention-based estimate places latent tokens and language tokens in the same evidence space: both are evaluated by how their hidden states causally respond to removing each answer-relevant visual region. Therefore, the entropy gap should be interpreted as evidence-dispersion rather than next-token uncertainty. High values indicate that a state remains sensitive to multiple grounded visual evidence nodes, while low values indicate that the state has concentrated on a narrower set of evidence required for final vocabulary-aligned decoding.

E Inference-time Latent-State Interventions

To complement the observational analyses in Section 5.3, we conduct inference-time interventions to examine whether the learned latent states contribute to answer generation. These interventions target three properties of the latent pathway: its availability, its instance-specific content, and its sequential organization.

Experimental Setup. We use the trained RIS model with five latent states and evaluate it on the same five benchmarks as in Section 5.2. Model parameters, visual and textual inputs, and decoding settings remain unchanged across conditions. Each intervention modifies only the latent pathway at inference time, without additional training. We compare the following six interventions against the full RIS model:

- **Remove latent states.** We remove all continuous latent states, so that the model generates answers without these intermediate reasoning states.
- **Latent-path masking.** We retain the latent states but block information transmission from them to answer tokens, testing the contribution of the latent-to-answer pathway.

Table 6: Inference-time latent-state interventions on RIS. All scores are accuracy (%). Average denotes the unweighted mean across the five benchmarks. All interventions use the same trained model without further optimization.

Intervention	V*	HRBench4K	HRBench8K	MMVP	BLINK	Average
RIS (full model)	**83.75**	**73.23**	**68.52**	**73.55**	**60.60**	**71.93**
Remove latent states	77.84	67.92	63.08	69.03	53.57	66.29
Latent-path masking	78.31	68.45	63.49	69.58	53.51	66.67
Cross-example shuffling	75.26	65.83	60.95	67.29	52.66	64.40
Within-example shuffling	76.04	66.37	61.62	67.95	52.26	64.85
Position-wise mean replacement	77.12	67.31	62.67	68.68	52.44	65.64
Moment-matched noise replacement	78.02	68.84	63.58	69.89	55.31	67.13

- **Cross-example shuffling.** We replace an example's latent states with states from other examples at the corresponding latent positions. This disrupts the correspondence between latent content and the current input while preserving slot alignment.
- **Within-example shuffling.** We randomly permute the latent states within each example, preserving the set of states while disrupting their original order.
- **Position-wise mean replacement.** We replace each latent state with the mean representation at the corresponding latent position, computed over the training set. This retains a position-specific representation while removing instance-specific variation.
- **Moment-matched noise replacement.** We replace the latent states with random noise matched to their mean and variance, preserving these low-order statistics while disrupting the encoded content.

Dependence on Latent Availability. As shown in Table 6, every intervention reduces performance on all five benchmarks. Removing the latent states decreases average accuracy from 71.93 to 66.29, while latent-path masking reduces it to 66.67. These results support the functional contribution of the latent pathway to answer generation, complementing the representation and attention analyses.

Dependence on Instance-specific Content. Cross-example shuffling reduces average accuracy to 64.40, the lowest among the evaluated interventions. Position-wise mean replacement and moment-matched noise replacement also degrade performance, yielding average accuracies of 65.64 and 67.13, respectively. Thus, retaining latent positions, position-specific mean representations, or matched low-order statistics does not preserve the performance of the full model. These findings indicate that RIS uses latent content associated with the current input. The larger degradation under cross-example shuffling than under removal further suggests that mismatched latent content can interfere with answer generation.

Dependence on Sequential Organization. Within-example shuffling reduces average accuracy to 64.85 despite retaining the same set of latent states for each input. This suggests that answer generation is sensitive to how latent content is organized across positions, consistent with the step-specific structure encouraged by grounded supervision.

Together, these interventions provide evidence that RIS relies on the availability, instance-specific content, and sequential organization of its latent states. Since the perturbations may also introduce distribution shift, we interpret the results as evidence of functional dependence on the learned latent pathway, rather than proof that each latent state is individually necessary.

F Attention Analysis Details and Robustness

We provide the attention definitions, aggregation procedure, and supplementary robustness checks for Figure 1(c) and Section 3.3. The analysis characterizes the allocation of attention during final-answer decoding. The functional contribution of latent-state content is evaluated separately through the inference-time interventions in Appendix E.

Query Positions and Token Groups. We measure post-softmax causal self-attention from all final-answer token queries to historical key positions. Final-answer queries exclude the transition tokens that precede the final answer. For each query, historical positions are partitioned into six mutually exclusive groups:

- **Image:** Original image tokens.
- **Question/Text:** Original question and textual input tokens.
- **Continuous Latent:** The five recurrent continuous hidden-state positions, $c_1, \ldots, c_5$, excluding discrete boundary and control tokens.
- **Transition:** The intermediate language tokens that bridge latent reasoning and final-answer generation.
- **Boundary/Control:** Discrete tokens used to delimit stages or switch generation modes.
- **Preceding Answer:** Previously generated final-answer tokens, excluding transition tokens.

These groups exhaust the historical key positions used in the analysis. Preceding Answer depends on the query position: it is empty for the first final-answer query and contains earlier final-answer tokens for subsequent queries. We apply the same grouping to LVR and RIS; LVR has no transition tokens.

Layer and Head Selection. The main analysis uses the final decoder layer of Qwen2.5-VL-7B and averages over all 28 query attention heads. This layer is adjacent to the final normalization and language-model head, making it a natural location to examine attention allocation near the output interface. Averaging over all query heads avoids selecting heads after inspecting their patterns. We additionally average over the last four decoder layers to assess sensitivity to layer selection.

Aggregation Procedure. Let $A_{qk}^{(i,\ell,h)}$ denote the post-softmax attention weight from query position q to key position k, for example i, decoder layer ℓ, and query head h. Let Q_i contain all final-answer query positions, and let $G_i(q)$ contain the historical key positions assigned to group G for query q. We first sum the attention weights over keys within each group:

$$m_{q,h}^{(i,\ell)}(G) = \sum_{k \in G_i(q)} A_{qk}^{(i,\ell,h)}.$$

We then average over final-answer queries and all $H = 28$ query heads within each example, followed by an equal-weight average over the N examples:

$$M_i^{(\ell)}(G) = \frac{1}{|Q_i|H} \sum_{q \in Q_i} \sum_{h=1}^{H} m_{q,h}^{(i,\ell)}(G),$$

$$M^{(\ell)}(G) = \frac{1}{N} \sum_{i=1}^{N} M_i^{(\ell)}(G).$$

Group-wise summation measures the total attention allocated to an information source. This is useful when groups differ substantially in size: individually small weights over many image tokens can collectively constitute a major route for answer decoding. Averaging within examples before averaging across examples prevents longer answers from receiving greater weight.

For the last-four-layer analysis, we compute $M^{(\ell)}(G)$ separately for each of the four layers and take their arithmetic mean. For the first-answer-token analysis, we restrict Q_i to the first final-answer query while retaining the same head and example aggregation.

Group-level Attention and Layer Robustness. Table 7 reports the complete attention distribution for the final layer and the last-four-layer average. In the final layer, LVR assigns 56% to Image and 31% to Question/Text, compared with 2% to Continuous Latent. The remaining 11% is allocated to Boundary/Control and Preceding Answer. Under the full inference-time bottleneck, RIS assigns 22% to Continuous Latent and 70% to Transition, with the remaining 8% allocated to Boundary/Control and Preceding Answer. Its zero attention to the original inputs follows from the bottleneck mask.

The last-four-layer average preserves the same pattern. LVR allocates 85.2% to the original image and question/text tokens and only 2.6% to continuous latent states. RIS allocates 23.1% to continuous latent states and 68.4% to transition tokens. Thus, the observed routing pattern is consistent across the two layer aggregations.

Table 7: Group-level attention mass (%) using all final-answer queries. The final-layer rows correspond to Figure 1(c); the last-four-layer rows provide a robustness check. Each row sums to 100%.

Method	Layers	Image	Question/Text	Continuous Latent	Transition	Boundary/Control	Preceding Answer
LVR	Final	56.0	31.0	2.0	0.0	4.0	7.0
LVR	Last four	54.8	30.4	2.6	0.0	4.5	7.7
RIS	Final	0.0	0.0	22.0	70.0	3.0	5.0
RIS	Last four	0.0	0.0	23.1	68.4	3.2	5.3

Table 8: Group-level attention mass (%) for the first final-answer query in the final decoder layer. The corresponding all-query results are reported in Table 7.

Method	Image	Question/Text	Continuous Latent	Transition	Boundary/Control	Preceding Answer
LVR	59.4	32.1	2.3	0.0	6.2	0.0
RIS	0.0	0.0	25.6	70.8	3.6	0.0

Robustness to Query Selection. Later final-answer queries can attend to previously generated answer tokens. To examine whether this affects the observed routing pattern, we repeat the final-layer analysis using only the first final-answer query. In this setting, Preceding Answer is empty and receives zero attention.

As shown in Table 8, LVR still allocates 91.5% to the original inputs and only 2.3% to continuous latent states. RIS allocates 96.4% to continuous latent states and transition tokens combined. The persistence of this pattern at the first query indicates that it is not driven by attention to preceding answer tokens.

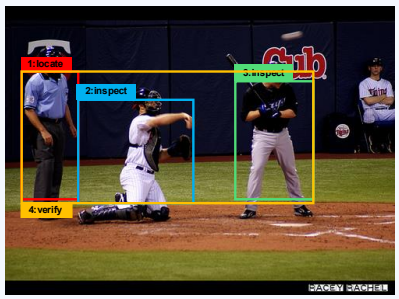
Slot-level Attention. The lower statistics in Figure 1(c) describe a separate, locally normalized view of RIS attention. We retain the final-answer queries but restrict the key positions to the latent-stage context: the five continuous latent positions and their associated boundary/control positions. Attention is renormalized within this context, with the same averaging over heads, queries, and examples. Transition and preceding-answer positions are excluded.

Consequently, the slot-level percentages describe relative attention allocation within the latent-stage context; they are not percentages of the full decoding context or a direct decomposition of the 22% group-level latent attention. The five displayed slot values sum to 97%, with the remaining 3% assigned to the associated boundary/control positions.

As shown in Table 9, the later latent slots receive a larger share of local attention under both layer aggregations. This is consistent with their proposed role in integrating evidence from earlier states, although attention allocation alone does not establish the computation performed by an individual slot.

G Dataset Statistics and Visualization

Tables 10–13 summarize the overall scale, reasoning-step distribution, JSONL sample schema, and per-step annotation format of GLSD. Figure 5–7 provide visualization of GLSD.



Question

Who is standing?

Full Answer

The umpire is standing.

Final Answer

umpire

Reasoning Chain

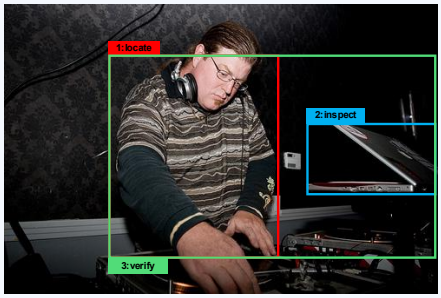
Operation: [locate]
Bbox: [0.04,0.23,0.18,0.67]
Region Description: the umpire on the left side of the image.
Visual Information: wearing a blue shirt and black pants, standing upright.

Operation: [inspect]
Bbox: [0.18,0.33,0.48,0.68]
Region Description: the catcher in the center.
Visual Information: in a crouched position, knees on the ground.

Operation: [inspect]
Bbox: [0.59,0.26,0.79,0.67]
Region Description: the batter on the right.
Visual Information: standing with bat raised, feet on ground.

Operation: [verify]
Bbox: [0.04,0.23,0.79,0.68]
Region Description: all individuals in the scene.
Visual Information: only the umpire and batter are standing; catcher is crouching; question asks 'who is standing?', and umpire is explicitly standing.

Figure 5: Example of 4-steps Supervision Sample.



Question

Who is the laptop in front of?

Full Answer

The laptop is in front of the man.

Final Answer

man

Reasoning Chain

Operation: [locate]
Bbox: [0.25,0.18,0.63,0.87]
Region Description: a man wearing glasses and a striped sweater, leaning over DJ equipment.
Visual Information: man is positioned behind the laptop, facing downward toward controls.

Operation: [inspect]
Bbox: [0.7,0.42,1.0,0.62]
Region Description: the open laptop on the right side of the DJ setup
Visual Information: laptop screen is angled upward, placed in front of the man's torso and hands

Operation: [verify]
Bbox: [0.25,0.18,1.0,0.87]
Region Description: spatial relationship between man and laptop
Visual Information: man's upper body and hands are behind the laptop, confirming laptop is in front of him

Figure 6: Example of 3-steps Supervision Sample.



Question

What is the menu on?

Full Answer

The menu is on the food truck.

Final Answer

food truck.

Reasoning Chain

Operation: [locate]
Bbox: [0.0,0.25,0.83,0.82]
Region Description: the white food truck on the left side of the image.
Visual Information: a large blue menu panel is mounted on its side.

Operation: [read]
Bbox: [0.12,0.39,0.31,0.64]
Region Description: the blue menu panel on the food truck's side.
Visual Information: text includes 'TACOS' and other items.

Figure 7: Example of 2-steps Supervision Sample.

Table 9: Locally normalized attention mass (%) within the RIS latent-stage context. Both rows use all final-answer queries. Boundary/Control includes only positions associated with the latent stage.

Layers	c_1	c_2	c_3	c_4	c_5	Boundary/Control
Final	9.0	13.0	12.0	29.0	34.0	3.0
Last four	9.4	13.2	12.5	28.6	33.1	3.2

Table 10: Statistics of the *Grounded Latent Supervision Dataset* (GLSD).

Item	Value
Source dataset	GQA train split
Storage format	JSONL
Parseable samples	96,000
Reasoning steps per sample	2–4
Total grounded reasoning steps	256,444
Average reasoning steps	2.67
Spatial supervision	Normalized and pixel-level bounding boxes
Semantic supervision	Region descriptions and visual information
Answer fields	full answer and final answer

Table 11: Distribution of reasoning-chain lengths in GLSD.

Reasoning-chain length	Number of samples	Percentage
2 steps	47,410	49.39%
3 steps	32,736	34.10%
4 steps	15,854	16.51%
Total	96,000	100.00%

Table 12: Top-level fields in each GLSD JSONL sample.

Field	Type	Description
`question`	string	Question text
`answer`	string	final answer phrase
`full_answer`	string	elaborated answer (transition)
`image`	string	GQA image filename
`width`, `height`	int	Image resolution
`dataset`, `split`	string	Source dataset and split
`reasoning_chain`	array	Step-wise grounded reasoning trace
`annotation_mask`	array[int]	Valid-step mask padded to budget K
`K`	int	Latent-slot budget in the stored sample
`reasoning_chain_viz_file`	string	Optional visualization filename

Table 13: Fields of each reasoning step in `reasoning_chain`.

Field	Type	Description
`step`	int	Step index starting from 1
`operation`	string	Step type, e.g., `locate`, `inspect`, `verify`
`bbox_01`	array[float]	Normalized box $[x_1, y_1, x_2, y_2]$ in $[0, 1]$
`bbox_pixels`	array[int]	Pixel-space bounding box
`region_description`	string	Description of the attended region
`visual_information`	string	Visual evidence extracted from the region