Beyond Visual Memory: Mechanistic Diagnostics of Latent Visual Reasoning

Anonymous ACL submission

Abstract

Recent latent visual reasoning methods achieve substantial gains by inserting continuous latent tokens into multimodal language models. These gains are commonly attributed to the tokens encoding visual evidence; recent analyses, however, reveal a paradox: the tokens are loosely tied to the image and contribute little to the answer. Critically, these analyses treat latent tokens as a single unit, obscuring the true source of the gains. We therefore decompose latent tokens into three testable components: latent slots, boundary markers, and format, and develop a state-of-the-art method as a probe under favorable conditions. Across six method-stage settings and four perception-heavy benchmarks, latent slots fail every prediction of the visual-memory account. Strikingly, retaining only the boundary markers preserves 78 to 100% of the gain in several settings, while the model attends to the image more narrowly at latent positions than at answer positions. The gain therefore comes from boundary markers, format, and this attention pattern, not from latent slots. How each method engages this mechanism depends on its training supervision: at matched accuracy, mechanisms can still differ markedly. Latent visual reasoning thus needs evaluation not only by accuracy but by what the model actually relies on.

1 Introduction

Visual reasoning often relies on signals that resist textual encoding: fine-grained spatial relations, localized evidence, and intermediate visual hypotheses lose their structure once verbalized into a textual chain of thought (Hu et al., 2024; Man et al., 2025; Wei et al., 2022). Recent methods therefore insert latent tokens into multimodal language models, conditioning final answers on intermediate hidden states rather than text alone (Bigverdi et al., 2025), with a common design wrapping a span of latent tokens with two learned delimiters:

$$\langle \texttt{lat_start} \rangle, s_1, \ldots, s_K, \langle \texttt{lat_end} \rangle. \quad (1)$$

The learned delimiters are defined as *boundary markers*, while the intervals between them are defined as *latent slots*. The latent slots condition the following answer and are commonly assumed to encode visual evidence beyond text, working like an internal sketchpad. We term this theoretical framework the *visual-memory account*.

Recent analyses challenge this account, finding latent tokens only loosely tied to the image and only weakly causal for the answer (Zhang et al., 2026; Li et al., 2026b; Zhang et al., 2025c). Yet they treat latent tokens as a single unit, leaving open which component is at fault and whether any remains functional. Final accuracy cannot decide: the same gain could come from latent slots, boundary markers, format, or attention shifts during latent generation.

We therefore decompose latent tokens into three independently testable components: latent slots,

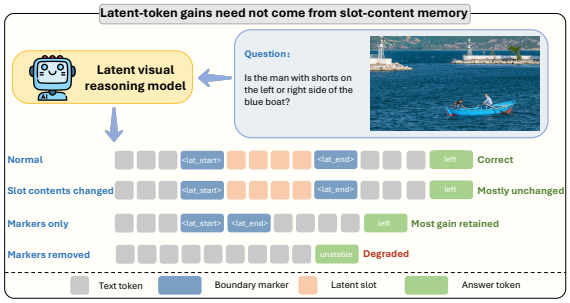

Figure 1: **Latent-token gains need not come from slot-content memory.** Changing slot contents has little effect, while keeping only the learned boundary markers preserves 78–100% of the latent-token gain in most MVH and ILVR settings. Corrupting the boundary markers can instead cause degenerate generation, suggesting that latent-token gains can arise from boundary markers and format rather than from recoverable slot contents.

boundary markers, and format, and intervene on each in isolation. To test the visual-memory account under conditions favorable to it, we develop **MVH** (Mesoscale Visual Hypotheses), a *content-favorable probe* that supervises slot contents with broad image coverage and question-relevant anchors. Under matched data, backbone, and hyperparameters, MVH reaches state-of-the-art accuracy among the latent-visual methods we compare (Table 1): if slot contents ever encode recoverable visual evidence, they should encode it here.

Across six method-stage settings and four perception-heavy benchmarks, slot contents fail every prediction of the visual-memory account: changing them moves accuracy by at most 0.027, image-conditioned slot states recover only 7.8% of the missing-image gap, and cross-example swaps transfer donor answers at 3.5–12.3%, near chance. Strikingly, retaining only the boundary markers preserves 78–100% of the latent-token gain in several settings, while at the same positions the model attends to the image more narrowly than during answer generation. The gain therefore comes from boundary markers, format, and this accompanying attention pattern, not from slot contents (Figure 1).

Settings with near-identical accuracy further separate along two diagnostic axes: dependence on boundary markers and format, and proximity between slot contents and projected visual-token representations. Methods that directly anchor slots to visual-token representations develop strong marker dependence, while methods that distill from teacher hidden states do not. Two methods can match in accuracy yet rely on latent tokens through markedly different mechanisms.

Contributions.

1. We decompose latent tokens into three independently testable components: latent slots, boundary markers, and format, moving the analysis from a single-unit view to a component-level one.

2. We develop MVH, a content-favorable probe that reaches state-of-the-art accuracy under matched settings, and use it to give the visual-memory account its strongest test.

3. Beyond negative findings, we locate where the gain comes from: boundary markers, format, and a focused attention pattern at latent positions. We further show that how each method engages these sources varies with training supervision, so latent visual reasoning should be evaluated by mechanism-level probes rather than accuracy alone.

2 Related Work

Latent visual reasoning methods. Recent methods insert continuous visual states into multimodal LLM decoding. Mirage grounds latent tokens via image-embedding supervision and aligns the trajectory with task objectives (Yang et al., 2026); LVR interleaves latent and text generation to reconstruct query-relevant visual tokens (Li et al., 2026a); ILVR adds selective perceptual modeling under a teacher model (Dong et al., 2026); and Monet uses teacher-state distillation refined by a visual-latent policy objective (Wang et al., 2026a). Other methods enrich the supervision signal in different ways (Qin et al., 2025; Chen et al., 2026; Wang et al., 2026b; Ding et al., 2026; Sun et al., 2025), while a separate line externalizes visual reasoning through sketches, pixel-level renderings, or tool-assisted inspection (Zhang et al., 2025a; Li et al., 2025; Zheng et al., 2026). Across continuous-latent designs, slots are typically described as carrying task-relevant visual evidence; we test whether these hidden states actually behave as recoverable visual memory rather than propose yet another such design.

Analyses and critiques of latent visual reasoning. Recent analyses challenge this picture. CapImagine (Li et al., 2026b) reports that inputs have little effect on latent states and that latent states have little effect on the answer, and shows explicit text-based imagination matches or exceeds latent-space methods on vision-centric benchmarks. Viveiros et al. (2026) find that uninformative dummy tokens leave accuracy nearly unchanged, tracing this to oracle latents carrying little information and inference-time latents collapsing to a narrow region. Zhang et al. (2026) identify a "silenced visual latents" phenomenon and propose inference-time optimization to recover the suppressed contributions. Our work differs in two ways: we build MVH, which reaches state-of-the-art accuracy under matched settings with structured visual targets, so the visual-memory account is tested under favorable conditions; and by intervening on slot contents, boundary markers, and format separately, paired with visual-subspace and image-attention probes, we move from diagnosing failure to characterizing how latent tokens are used when slot-content mem-

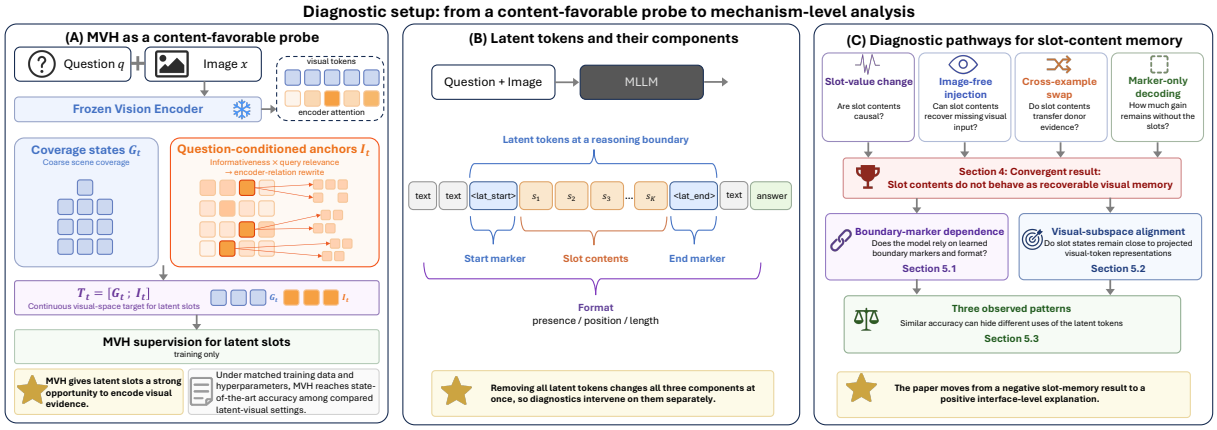

Figure 2: **Diagnostic setup.** (A) MVH trains latent slots with visual targets that combine broad image coverage and question-relevant anchors. (B) Latent tokens have three components that can be intervened on separately: slot contents, boundary markers, and format. (C) We first test whether slot contents behave as recoverable visual memory, then examine how the latent interface is used through boundary-marker dependence and visual-subspace alignment.

ory fails. This is complementary to CapImagine: high-accuracy latent methods do not necessarily achieve their gains through the content storage their design suggests.

Latent reasoning and structural tokens. Our results connect to broader work on tokens that carry no textual content (Zhu et al., 2025): continuous thoughts (Hao et al., 2025), pause tokens (Goyal et al., 2024), filler tokens (Pfau et al., 2024), and register tokens in vision transformers (Darcet et al., 2024) all show that non-content tokens can shape behavior. Latent tokens in a multimodal model, however, combine slot contents, boundary markers, format, and a geometry tied to the visual encoder. The decomposition into these components, and the separation of marker-driven control from slot-driven content, have no direct analogue in the text-only setting; we examine each component in Section 4.

3 Diagnostic Setup

We first describe MVH as a favorable probe (§3.1), then specify the settings, benchmarks, and tests used in the analysis (§3.2). Figure 2 gives an overview.

3.1 MVH: A Content-Favorable Probe

Earlier latent-visual methods supervise slot states at different levels of detail. Patch-level alignment ties each slot to a specific visual region, while distillation from teacher states provides a flexible but less explicit visual signal. MVH sits between these two extremes: each slot is trained toward a target that combines broad image coverage with question-relevant anchors (Figure 2A). The design intentionally gives the visual-memory account a structured, explicitly visual target. If slot contents can ever encode recoverable visual evidence, MVH should be one of the easier settings in which to observe it.

Target structure. At each reasoning step t, the frozen vision encoder produces visual tokens $V_t = \{v_{t,1}, \ldots, v_{t,M_t}\}$ on an $H_t \times W_t$ grid, with self-attention maps $\{A_t^{(h)}\}$. The teacher target is

$$T_t = [G_t; I_t], \qquad |T_t| = K_g + K_i = N, \quad (2)$$

where G_t contains K_g coverage states (the global mean of V_t plus nine regional means over a 3×3 partition) and I_t contains K_i question-relevant anchors. We use $K_g = K_i = 10$ and latent tokens of length $N = 20$ at training time.

Selecting anchors. Anchors are picked by a fused score

$$p_{t,j} \propto r_j^{\text{base}} \cdot u_{t,j}, \quad (3)$$

where r_j^{base} measures the average attention each visual token receives from the encoder, and $u_{t,j}$ measures the relevance of that token to the current decoder context. The top-K_i tokens are then rewritten into compositional anchors that mix each selected token with its encoder-attention neighbors, so anchors reflect contextual co-evidence rather than purely appearance-based redundancy. Full equations for the score, the rewriting kernel, the

Setting	V*	HR-4K	HR-8K	MME-RW	Avg.
Proprietary Model					
GPT-4o	67.5	59.0	55.5	52.0	58.5
Open-Source Baselines					
Qwen2.5-VL-7B	76.4	68.0	63.8	45.8	63.5
+ vanilla SFT	81.7	68.4	61.6	51.3	65.8
+ SFT + GRPO	78.5	70.0	66.8	52.4	66.9
ILVR-Stage1	80.6	70.0	66.5	49.2	66.6
ILVR-Stage2	80.6	70.3	66.6	50.4	67.0
Monet-SFT	79.6	70.4	63.0	52.9	66.5
Monet-RL	82.2	71.0	66.0	51.2	67.6
Our Model					
MVH-SFT	82.2	71.5	67.1	50.4	67.8
MVH-RL	**83.2**	**72.4**	**69.5**	**53.8**	**69.7**

Table 1: **Accuracy of the analyzed settings.** SFT-stage settings share the same backbone, SFT data, and hyperparameters. For RL-stage settings, Monet's RL data is not public; we train MVH-RL using data built following Monet's construction pipeline. MVH reaches the strongest accuracy among the latent-visual settings we analyze, supporting its use as a favorable probe. The **bold numbers** indicate the best performance achieved by each setting.

EMA teacher, and norm-aligned injection are in Appendix A.

SFT stage. The SFT stage supervises each latent slot toward its target via next-token-shifted cosine alignment:

$$\mathcal{L}_{\text{lat}} = \frac{1}{|\mathcal{S}_{\text{lat}}|} \sum_{(t,i)\in\mathcal{S}_{\text{lat}}} \big(1 - \cos(z_{t,i-1},\, \tau_{t,i}^{\text{aligned}})\big), \tag{4}$$

where $z_{t,i-1}$ is the student's last-layer hidden state at the position immediately before the i-th slot, and $\tau_{t,i}^{\text{aligned}}$ is the norm-rescaled teacher target. Cosine alignment is used because norm-aligned injection already matches the scale of the teacher to the embedding space, leaving direction as the only degree of freedom. The full SFT objective combines this loss with the standard text cross-entropy under a linear warm-up.

RL stage. The RL stage refines the policy with an outcome reward while preserving the slot geometry learned during SFT. We sample G rollouts per prompt and compute GRPO (Shao et al., 2024) advantages $\hat{A}_g$ from a reward mixing answer correctness and format. Text tokens follow the standard clipped PPO objective (Schulman et al., 2017). Slot states are continuous and have no token-level log-probability, so instead of fitting a parametric density over them as in VLPO (Wang et al., 2026a), we propagate each rollout's advantage through cosine alignment between the on-policy slot state $z_{t,i}^{\theta}$ and the rollout-time state $z_{t,i}^{(g)}$:

$$\mathcal{L}_{\text{lat}}^{\text{RL}} = -\,\mathbb{E}_{g,(t,i)}\big[\hat{A}_g \cdot \cos(z_{t,i}^{\theta},\, z_{t,i}^{(g)})\big]. \tag{5}$$

A latent trust region anchored to the frozen SFT checkpoint, the continuous analogue of PPO's reference-KL, prevents drift from the SFT subspace. Full objective and weights are in Appendix A.

Inference. No teacher is needed at inference: the model generates the latent tokens autoregressively from image and question. We use $K_{\text{eval}} = 8$ latent slots per step in all evaluations.

Why MVH is a favorable probe. Two pieces of evidence support this role. First, under matched data, backbone, and hyperparameters, MVH-RL reaches state-of-the-art accuracy among the latent-visual settings we compare (Table 1). Second, the latent slots produced by MVH respond to the input: their cross-sample cosine similarity averages 0.68, well below the 0.96–0.97 observed for ILVR and Monet (Figure A1A), showing that they vary with the input rather than collapsing across examples. Within a sample, coverage and anchor slots separate into distinct sub-blocks (BSR $= 1.13$), and on the same image with different questions, the anchor slots shift more than the coverage slots (RSR $= 1.12$, against ≈ 1.00 for baselines). Accuracy alone would be a circular justification for the probe; the structural diagnostics above are independent of accuracy, and they show that MVH's slots respond to the input and to question variation in ways the visual-memory account would predict.

3.2 Settings, Benchmarks, and Tests

Latent tokens (Figure 2B) have three components that can be intervened on separately: the slot contents, the boundary markers, and the latent format (its presence, position, and length). We exploit this separability throughout the analysis. Each test in Sections 4–5 changes one component at a time and looks at how accuracy or generation behavior responds.

Settings. We analyze six method-stage settings across three supervision strategies. MVH-SFT and MVH-RL correspond to the two stages of MVH described above. The other two methods are chosen for their representativeness: ILVR adopts

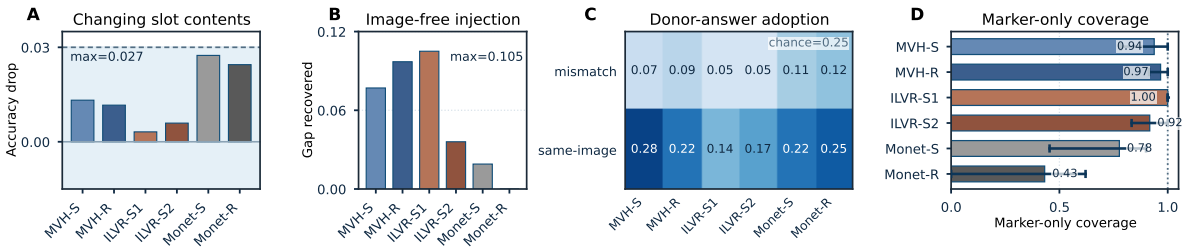

Figure 3: **Four tests of slot-content memory.** (A) Changing slot contents (zero, random, or fixed substitutes) produces only small accuracy changes. (B) Injecting image-conditioned slot states into image-free runs recovers little of the missing-image gap. (C) Swapping real slot states across examples does not reliably transfer donor-specific answers; donor-answer adoption stays near or below chance, including in same-image cross-question controls. (D) Marker-only decoding recovers much of the unique latent-token gain over no-latent decoding. Together, these tests do not support the view that slot contents behave as recoverable visual memory, even under content-favorable MVH supervision.

direct visual-token alignment and Monet adopts teacher-state distillation, the two dominant supervision strategies in latent visual reasoning, and both achieve strong performance among published methods. ILVR-Stage1 and ILVR-Stage2 differ in that Stage2 relaxes the alignment target; Monet-SFT and Monet-RL differ in that the RL stage applies reward-based optimization on answer correctness. All six settings share the same latent-token structure but differ in how the slot states are supervised.

Training data. Monet checkpoints come from the original authors. We re-implement and train the other four settings: MVH-SFT, ILVR-Stage1, and ILVR-Stage2 share the SFT corpus released by Monet, making the SFT comparison fully matched in data. Monet's RL corpus is not released; we follow its pipeline to build one from DeepEyes-v2 and use it to train MVH-RL. The RL comparison is therefore not data-matched, but the SFT results alone already establish MVH's accuracy advantage. Reproduction details appear in Appendix B.

Benchmarks. We evaluate on V* (Wu and Xie, 2024), HRBench-4K, HRBench-8K (Wang et al., 2025), and MME-RealWorld-Lite (Zhang et al., 2025b). This suite stresses fine-grained recognition, high-resolution inspection, spatial reasoning, and real-world visual understanding, and is well suited for testing whether latent states carry task-relevant visual evidence.

4 Slot Contents Do Not Behave as Recoverable Visual Memory

We now test the visual-memory account on the probe established in Section 3. If slot contents act as recoverable visual memory, they should be causal for the answer, recoverable when the image is removed, portable across samples, and responsible for the unique latent-token gain. We test these four predictions with four complementary tests (Figure 3).

4.1 Four Tests of Slot-Content Memory

Changing slot contents. The first test replaces the slot contents with substitutes (zero, random, or fixed vectors) while keeping the boundary markers in place. If slot contents carry the evidence used by the decoder, these substitutions should hurt accuracy. They do not. Benchmark-averaged accuracy moves by at most 0.027 across the six settings (Figure 3A), even though the same models respond strongly to other interventions on the latent tokens (Section 5). Slot contents are weakly causal under direct perturbation.

Injecting latents into image-free runs. A natural follow-up asks whether slot contents can recover visual information when the image itself is removed. We run this test across the method-stage settings in Figure 3B. For each example, we blank the image and inject the slot states that the same model produced on the same example with the image present. If latent tokens carry recoverable visual content, these states should close a substantial fraction of the gap between blind and full-image decoding. Across settings, injected image-conditioned states recover only a small fraction of the missing-image gap. For example, MVH-SFT reaches 0.845 with the image and 0.228 without it; injected states raise blind accuracy only to 0.276, recovering 7.8% of the gap (Figure 3B). The task

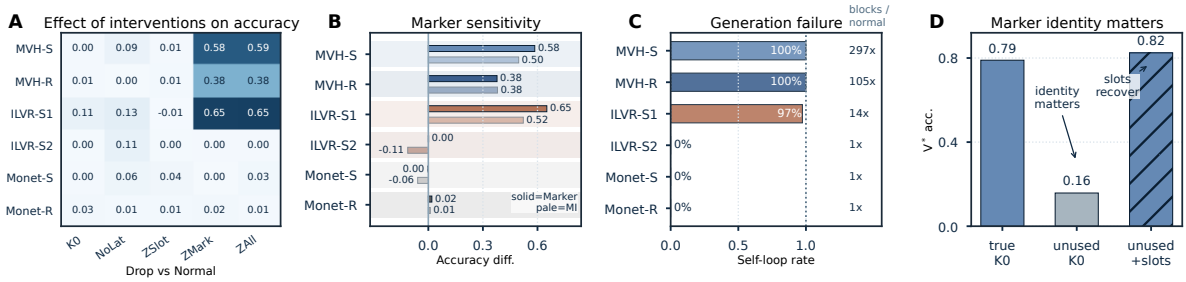

Figure 4: **Marker-dependence diagnostics.** Changing slot contents has little effect on accuracy, but zeroing the boundary-marker embeddings disrupts the marker-dependent settings, especially MVH-SFT and ILVR-Stage1, causing large accuracy drops and self-looping latent tokens. Forced-prefix controls show that trained markers are not replaceable by arbitrary special tokens. K0 keeps markers but removes slots; NoLat removes all latent tokens; ZSlot zeros slot states; ZMark zeros marker embeddings; ZAll zeros both. Full condition definitions in Table A6.

clearly requires visual input, but the input is not substantially recoverable from the slot states alone.

Swapping slot states across examples. Zero, random, and fixed substitutes are out-of-distribution. This test removes that concern: slot states are captured from real model runs on donor examples and injected at the corresponding decoder layer of a target run, with the layer swept across three depths (Appendix F.4). If slot states behave as recoverable visual memory, a real donor state should transfer some of the donor's content, especially when donor and target share the same image but ask different questions. This same-image cross-question control is the most favorable to the visual-memory account: donor and target share the visual scene and differ mainly in the question-dependent evidence required to answer. Across the six settings and three layers, swap accuracy stays close to self-swap, and donor-answer adoption ranges from 3.5% to 12.3% (Figure 3C), at or below the 25% chance level for four-way multiple choice, including in the same-image control. Slot states transfer little donor-specific information.

Marker-only decoding. The final test moves from aggregate accuracy to the examples on which latent tokens are most clearly useful: those answered correctly by normal latent decoding, but missed when we drop the latent tokens entirely. We ask, of these examples, how many remain solved if we keep only the boundary markers and drop the slots between them. If slot contents carry the unique latent-token contribution, this fraction should be low. Instead, marker-only decoding recovers 78–100% of these gained examples in most MVH and ILVR settings (Figure 3D).[1] The unique latent-token contribution is therefore largely recoverable without the slot states that sit between the markers.

4.2 Evidence Across Tests

The four tests make different assumptions but agree (Figure 3): perturbation tests causal sensitivity, image-free injection tests recoverability, swaps test portability, and marker-only decoding tests responsibility for the unique latent-token gain. Together, slot contents barely affect the answer, do not recover missing visual input, do not transfer across examples, and are not needed for most of the unique latent-token gain. These results do not support the visual-memory account, even when latent slots are trained with content-favorable MVH supervision.

5 How Latent Tokens Are Used Without Slot-Content Memory

Section 4 establishes two anchors for what follows: slot contents do not function as visual memory, while marker-only decoding preserves much of the latent-token gain. Two questions follow. Are boundary markers interchangeable placeholders or learned control signals, and does the answer hold across methods? Beyond the markers, does latent generation itself carry any role? Section 5.1 characterizes the markers, Section 5.2 examines visual behavior during latent generation, and Section 5.3 combines the two axes.

[1] Monet-RL is an exception; per-setting values and small-denominator cases are in Appendix F.5.

5.1 Boundary Markers Are Control Signals, Not Placeholders

Boundary markers act as learned control signals for entering and leaving latent generation, not as interchangeable placeholders. Whether they take this role depends on training supervision. Two observations support this claim, both from perturbing the markers while leaving slot states and format intact.

Observation 1: marker perturbation produces a sharply polarized effect. Zeroing marker embeddings causes MVH-SFT and ILVR-Stage1 to collapse, with benchmark-averaged drops of 0.585 and 0.651; MVH-RL loses 0.378, while ILVR-Stage2 and Monet remain essentially unaffected (≤ 0.02). The same polarization appears when the trained markers are replaced by unused special tokens or <bos>/<eos> (Appendix G), indicating that the trained markers carry information that arbitrary special tokens do not reproduce. The same latent format therefore supports markedly different modes of use across training settings; pure placeholders would not yield such polarization.

Observation 2: zeroing markers induces self-looping generation. In marker-dependent settings, zeroing the markers leads MVH-SFT and ILVR-Stage1 to enter self-looping latent generation in nearly all V* runs, repeatedly emitting latent insertions without terminating. This is not attributable to a generic loss of computation, as the latent slots remain present, and is consistent with disrupted control over how the decoder enters and exits latent mode.

Monet, where markers carry no such role, is not a counterexample but a prediction of this account: its training never anchors slot states to visual targets, so the markers are never recruited as mode-switching signals. We return to this supervision-mechanism link in Section 5.3. The observation that slot contents are inert while another component of the latent token carries the load parallels findings from *Let's Think Dot by Dot* (Pfau et al., 2024), which shows that content-free filler tokens can support hidden computation in text-only Transformers. Our results extend this picture along two multimodal-specific dimensions: markers and slot contents play asymmetric roles within the same token, and latent generation exhibits a visual behavior without a text-only analogue, which we examine next.

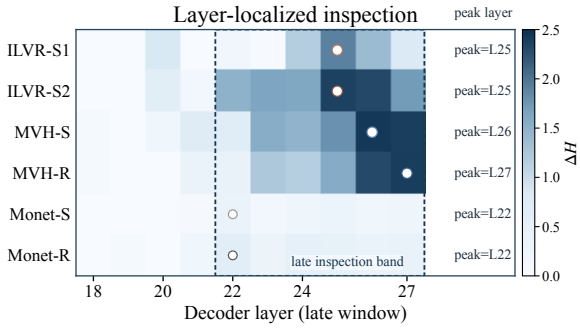

Figure 5: **Layer-localized visual inspection.** $\Delta H_l = H_l^{\text{answer}} - H_l^{\text{latent}}$; positive values mean image attention is more concentrated at latent than at answer positions. MVH and ILVR show clear late-layer peaks; Monet does not. An attention-based signature, not a causal test of visual content use.

5.2 Latent Generation Engages Vision Without Leaving a Recoverable Trace

During latent generation the model engages with the image, yet this engagement leaves no recoverable trace in slot states (Section 4). Two probes provide convergent positive evidence; both are diagnostic rather than causal.

Probe 1 (geometric): slot states lie close to visual-token representations in a method-dependent order. For each slot position we compute the cosine similarity between its final-layer hidden state and each projected visual-token representation, retain the top-5 values, and average across slots. The settings form a clear order: ILVR-Stage1 remains closest to visual-token representations, ILVR-Stage2 relaxes this alignment, MVH occupies a more compressed region of visual space, and Monet sits near the low-similarity end. A slot state can therefore be visually aligned without functioning as portable memory.

Probe 2 (dynamic): the model attends to the image more narrowly during latent generation than during answer generation. At decoder layer l, H_l^{latent} and H_l^{answer} measure the entropy of decoder attention to image tokens, averaged over slot positions and answer-token positions respectively (Appendix H). A positive $\Delta H_l = H_l^{\text{answer}} - H_l^{\text{latent}}$ indicates attention is more concentrated during latent generation. MVH and ILVR exhibit clear late-layer peaks, with peak ΔH ranging from 1.90 to 2.43 between decoder layers 25 and 27 (Figure 5). Monet's nominal peak appears at layer 22 but remains below $\Delta H = 1.0$.

Latent steps in MVH and ILVR are thus accompanied by a layer-localized visual-inspection sig-

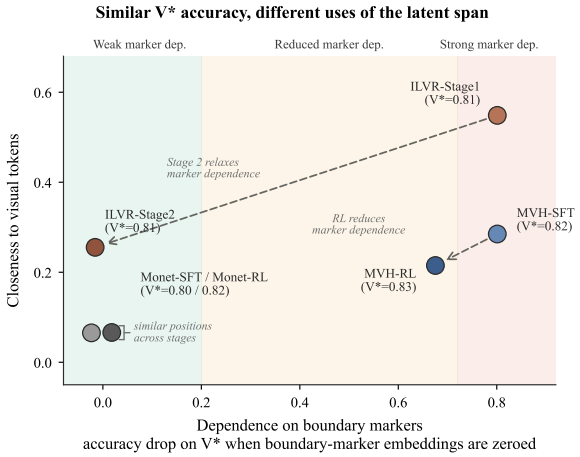

Figure 6: **Similar V* accuracy, different uses of latent tokens.** The six settings reach similar V* accuracy but separate along boundary-marker dependence and closeness to visual-token representations. Shaded regions are coarse descriptive ranges, not method categories. Arrows show how later training stages move settings along these axes.

Metric	Strong	Reduced	Weak
Settings	MVH-S/ILVR-S1	MVH-RL	ILVR-S2/Monet
Marker sens.	0.59–0.65	0.38	$\leq$0.02
MOC	78–100%	91–100%	0–90%
Top-5 sim.	0.13–0.55	0.10	0.07–0.27
Peak ΔH	$\geq$1.90	2.39	Monet $<$1.0

Table 2: **Measurements behind the mechanism map.** Columns group settings by marker-dependence range (Figure 6). Marker sensitivity and MOC track marker reliance; top-5 visual similarity and peak ΔH track visual-subspace and attention signatures. MOC: fraction of latent-uniquely-solved examples still solved with markers only. Full per-setting values in Appendix I.

nature, while Monet shows no such signature. The visual engagement that does occur is observable in attention behavior but not extractable from the hidden states the slots produce.

5.3 Same Accuracy, Different Mechanisms

At matched accuracy, methods rely on latent tokens through markedly different mechanisms, and these mechanisms are shaped by training supervision. The two diagnostic axes vary sharply across the six settings: marker drop ranges from ≤ 0.02 to 0.651, and top-5 visual similarity from below 0.10 to above 0.50, even though V* accuracy remains within 0.80–0.83 (Figure 6, Table 2).

Finding 1: training supervision places methods on both axes. MVH-SFT and ILVR-Stage1 anchor slot states directly to visual-token representations. The model learns to switch into a visual processing mode at the markers, producing strong marker dependence (0.585 and 0.651); slot states are pulled into the visual subspace, so latent generation is accompanied by focused image attention ($\Delta H \geq 1.90$). Monet distills teacher hidden states instead, so markers are never recruited as control signals and latent generation is never directed toward the image; both axes consequently remain low (marker dependence ≤ 0.02, $\Delta H < 1.0$). RL acts as a relaxer rather than a builder: in MVH-RL, refinement on top of MVH-SFT reduces marker dependence from 0.585 to 0.378 without driving it to zero.

Finding 2: the two axes can vary independently. The clearest evidence comes from ILVR's Stage1-to-Stage2 transition: relaxing the visual-alignment target collapses marker dependence from 0.651 to ≤ 0.02, yet the late-layer visual-inspection signature is preserved ($\Delta H \geq 1.90$). A setting can therefore lose its reliance on markers while retaining focused image attention.

Two methods can thus reach near-identical accuracy while using latent tokens in mechanistically different ways. Probing how latent tokens are actually used, through marker interventions, visual-subspace geometry, and attention behavior during latent generation, is required to distinguish mechanisms that accuracy alone obscures.

6 Conclusion

Latent visual reasoning is often described as if the slots between boundary markers carried portable visual evidence. We tested this account on a favorable probe, MVH, which reaches state-of-the-art accuracy among the latent-visual settings we compare. Across four tests, slot contents prove weakly causal, do not recover missing visual input, do not transfer donor content, and are not required for most of the unique latent-token gain. Latent tokens nonetheless contribute, through their boundary markers, the format they impose, and the focused image attention that accompanies latent generation. Two diagnostic axes, dependence on boundary markers and proximity between slot states and visual-token representations, further separate settings with similar V* accuracy into mechanistically distinct uses of the latent tokens, shaped by training supervision. Latent visual reasoning should therefore be evaluated not only by accuracy but by what the model actually relies on.

Limitations

Our analysis focuses on latent visual reasoning methods that use a full-image VLM interface and insert latent slots into the decoding sequence. Architectures that enforce a stronger visual bottleneck, restrict image access during answer generation, or use explicit external visual workspaces may show different balances between slot contents, boundary markers, and format. We also study a controlled set of method-stage settings built around Qwen-family backbones and perception-heavy benchmarks. Broader model families, video inputs, embodied tasks, or long-horizon planning may introduce additional uses of latent states that are not covered here. Finally, our diagnostics are designed to test whether slot contents act as recoverable visual memory; they do not rule out other roles for latent slots, such as providing computation, timing, formatting, or routing capacity during decoding.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025a. Qwen3-vl technical report. *Preprint*, arXiv:2511.21631.

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

Mahtab Bigverdi, Zelun Luo, Cheng-Yu Hsieh, Ethan Shen, Dongping Chen, Linda G. Shapiro, and Ranjay Krishna. 2025. Perception tokens enhance visual reasoning in multimodal language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3836–3845.

Zhangquan Chen, Manyuan Zhang, Xinlei Yu, Xufang Luo, Mingze Sun, Zihao Pan, Xiang An, Yan Feng, Peng Pei, Xunliang Cai, and Ruqi Huang. 2026. Think with 3d: Geometric imagination grounded spatial reasoning from limited views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Zihui Cheng, Qiguang Chen, Jin Zhang, Hao Fei, Xiaocheng Feng, Wanxiang Che, Min Li, and Libo Qin. 2025. Comt: A novel benchmark for chain of multimodal thought on large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23678–23686.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. 2024. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*.

Ziyang Ding, Linjian Meng, Yiming Wu, Yuhan Li, Yuhao Liu, and Zhen Zhao. 2026. Colvr: Enhancing exploratory latent visual reasoning via contrastive optimization. *Preprint*, arXiv:2605.08802.

Shuai Dong, Siyuan Wang, Xingyu Liu, Chenglin Li, Haowen Hou, and Zhongyu Wei. 2026. Interleaved latent visual reasoning with selective perceptual modeling. *Preprint*, arXiv:2512.05665.

Haodong Duan, Xinyu Fang, Junming Yang, Xiangyu Zhao, Yuxuan Qiao, Mo Li, Amit Agarwal, Zhe Chen, Lin Chen, Yuan Liu, Yubo Ma, Hailong Sun, Yifan Zhang, Shiyin Lu, Tack Hwa Wong, Weiyun Wang, Peiheng Zhou, Xiaozhe Li, Chaoyou Fu, and 13 others. 2025. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. *Preprint*, arXiv:2407.11691.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. 2024. Think before you speak: Training language models with pause tokens. In *International Conference on Learning Representations*, volume 2024, pages 27896–27923.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2025. Training large language models to reason in a continuous latent space. *Preprint*, arXiv:2412.06769.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah Smith, and Ranjay Krishna. 2024. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 139348–139379. Curran Associates, Inc.

Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Emad Barsoum, Muhao Chen, and Zicheng Liu. 2026a. Latent visual reasoning. In *The Fourteenth International Conference on Learning Representations*.

Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. 2025. Imagine while reasoning in space: Multimodal visualization-of-thought. In *Forty-second International Conference on Machine Learning*.

You Li, Chi Chen, Yanghao Li, Fanhu Zeng, Kaiyu Huang, Jinan Xu, and Maosong Sun. 2026b. Imagination helps visual reasoning, but not yet in latent space. In *Proceedings of the 43rd International Conference on Machine Learning*.

Yunze Man, De-An Huang, Guilin Liu, Shiwei Sheng, Shilong Liu, Liang-Yan Gui, Jan Kautz, Yu-Xiong Wang, and Zhiding Yu. 2025. Argus: Vision-centric reasoning with grounded chain-of-thought. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14268–14280.

Jacob Pfau, William Merrill, and Samuel R. Bowman. 2024. Let's think dot by dot: Hidden computation in transformer language models. In *First Conference on Language Modeling*.

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

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. 2025. Pixel reasoner: Incentivizing pixel space reasoning via curiosity-driven reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, pages 8222–8251. Curran Associates, Inc.

Guohao Sun, Hang Hua, Jian Wang, Jiebo Luo, Sohail Dianat, MAJID RABBANI, Raghuveer Rao, and Zhiqiang Tao. 2025. Latent chain-of-thought for visual reasoning. In *Advances in Neural Information Processing Systems*, volume 38, pages 103739–103762. Curran Associates, Inc.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

André G. Viveiros, Nuno Gonçalves, André F. T. Martins, and Matthias Lindemann. 2026. What's holding back latent visual reasoning? *Preprint*, arXiv:2605.18445.

Qixun Wang, Yang Shi, Yifei Wang, Yuanxing Zhang, Pengfei Wan, Kun Gai, Xianghua Ying, and Yisen Wang. 2026a. Monet: Reasoning in latent visual space beyond images and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. 2025. 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, pages 7907–7915.

Yubo Wang, Juntian Zhang, Yichen Wu, Yankai Lin, Nils Lukas, and Yuhan Liu. 2026b. Forest before trees: Latent superposition for efficient visual reasoning. *Preprint*, arXiv:2601.06803.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Penghao Wu and Saining Xie. 2024. V*: Guided visual search as a core mechanism in multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13084–13094.

Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. 2026. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Huanyu Zhang, Wenshan Wu, Chengzu Li, Ning Shang, Yan Xia, Yangyu Huang, Yifan Zhang, Li Dong, Zhang Zhang, Liang Wang, Tieniu Tan, and Furu Wei. 2025a. Latent sketchpad: Sketching visual thoughts to elicit multimodal reasoning in mllms. *Preprint*, arXiv:2510.24514.

Xin Zhang, Qiqi Tao, Jiawei Du, Moyun Liu, and Joey Tianyi Zhou. 2026. Visual latents know more than they say: Unsilencing latent reasoning in mllms. *Preprint*, arXiv:2605.02735.

YiFan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, Liang Wang, and Rong Jin. 2025b. MME-realworld: Could your multimodal LLM challenge high-resolution real-world scenarios that are difficult for humans? In *The Thirteenth International Conference on Learning Representations*.

Yuyi Zhang, Boyu Tang, Tianjie Ju, Sufeng Duan, and Gongshen Liu. 2025c. Do latent tokens think? a causal and adversarial analysis of chain-of-continuous-thought. *Preprint*, arXiv:2512.21711.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and XingYu. 2026. Deepeyes: Incentivizing "thinking with images" via reinforcement learning. In *The Fourteenth International Conference on Learning Representations*.

Rui-Jie Zhu, Tianhao Peng, Tianhao Cheng, Xingwei Qu, Jinfa Huang, Dawei Zhu, Hao Wang, Kaiwen

Xue, Xuanliang Zhang, Yong Shan, Tianle Cai, Taylor Kergan, Assel Kembay, Andrew Smith, Chenghua Lin, Binh Nguyen, Yuqi Pan, Yuhong Chou, Zefan Cai, and 14 others. 2025. A survey on latent reasoning. *Preprint*, arXiv:2507.06203.

A MVH: Full Method Details

This appendix gives the equations, training-time assembly, and hyperparameters elided in Section 3.1.

A.1 Problem setup

At each reasoning step t, the frozen vision encoder processes a helper image x_t^{hlp} (a crop, an annotated view, or a simulated next state drawn from the training data) and produces a set of post-merger visual tokens

$$V_t = \{v_{t,1}, \ldots, v_{t,M_t}\}, \qquad v_{t,j} \in \mathbb{R}^{d_h},$$

arranged on an $H_t \times W_t$ spatial grid with $H_t \cdot W_t = M_t$. From the last window-attention block of the encoder we additionally extract per-head self-attention maps $\{A_t^{(h)}\}_{h=1}^{H}$, where $A_t^{(h)}(i,j)$ is the weight from token i to token j in head h. Both V_t and $\{A_t^{(h)}\}$ are fixed once the helper image is fixed. We further denote by $h_t \in \mathbb{R}^{d_h}$ the student's last-layer hidden state at the text token immediately preceding the latent tokens at step t; this is the only quantity that changes step-to-step.

A.2 Teacher targets: design principle

At each step, MVH constructs a structured teacher hypothesis set

$$T_t = [G_t;\, I_t], \qquad |T_t| = K_g + K_i = N,$$

organized along two complementary axes. The coverage block G_t is question-independent and preserves global scene structure (§A.3); the informative block I_t is question-dependent and rewrites high-information-density anchor tokens with question-relevant evidence (§A.4). The assembly and injection procedures are given in §A.5–§A.7, and the training objectives that align student slots to T_t in §A.9–§A.10.

A.3 Coverage states

The global mean over visual tokens is

$$g_0 = \frac{1}{M_t} \sum_{j=1}^{M_t} v_{t,j}, \tag{6}$$

and $g_1, \ldots, g_9$ are the means of the tokens falling in each cell of a fixed 3×3 partition of the $H_t \times W_t$ grid. When a side is not divisible by three, the remainder is assigned to the last row/column; empty cells fall back to g_0. We take

$$G_t = \{g_0, g_1, \ldots, g_9\}, \qquad |G_t| = K_g = 10.$$

The 3×3 partition is the coarsest grid that can still express layouts such as a top-right quadrant or a middle-row band, without the combinatorial blow-up of denser partitions; the global token g_0 provides a grid-agnostic summary.

A.4 Informative hypotheses: selection and rewriting

Base informativeness. The encoder-side informativeness of token j is the per-head average of incoming attention,

$$r_j^{\text{base}} = \frac{1}{HM_t} \sum_{h=1}^{H} \sum_{i=1}^{M_t} A_t^{(h)}(i,j), \tag{7}$$

followed by an ℓ_1 renormalization. Tokens with high r_j^{base} act as information hubs that absorb and redistribute compressed global information during encoding.

Query relevance. We project the current step's text context h_t against the visual tokens with a scaled dot product and normalize,

$$q_{t,j} = (v_{t,j})^\top h_t \,/\, \sqrt{d_h}, \tag{8}$$

$$u_{t,j} = \frac{\exp(q_{t,j})}{\sum_k \exp(q_{t,k})}. \tag{9}$$

Because V_t and h_t already share the decoder's hidden dimension, no extra projection matrix is required.

Fused score and top-K_i selection. The two signals are combined multiplicatively with equal weight and renormalized,

$$p_{t,j} = \frac{r_j^{\text{base}} \cdot u_{t,j}}{\sum_k r_k^{\text{base}} \cdot u_{t,k}}, \tag{10}$$

and we take $I_t^{\text{idx}} = \text{Top-}K_i(p_t)$ with $K_i = 10$. Equal multiplicative weighting avoids a free mixing hyperparameter and prevents either signal from dominating.

Token-to-token relation. For anchor rewriting we use the encoder's own symmetrized self-attention,

$$s_{j,k} = \frac{1}{H} \sum_{h=1}^{H} \frac{A_t^{(h)}(j,k) + A_t^{(h)}(k,j)}{2}, \tag{11}$$

which captures *functional* relatedness inside the encoder rather than mere visual look-alikeness between patches.

Anchor rewriting. For each anchor $j \in I_t^{\text{idx}}$, we form a row-wise softmax over $s_{j,\cdot}$ with the self-edge masked,

$$w_{j,k} = \begin{cases} 0 & k = j, \\ \dfrac{\exp(s_{j,k})}{\sum_{m \neq j} \exp(s_{j,m})} & k \neq j, \end{cases} \tag{12}$$

and produce the rewritten anchor by a self/neighbor blend,

$$\tilde{v}_{t,j} = \rho\, v_{t,j} + (1-\rho) \sum_{k \neq j} w_{j,k}\, v_{t,k}, \qquad \rho = 0.6. \tag{13}$$

The blend is taken over the whole image rather than a local neighborhood, and $w_{j,k}$ is deliberately kept task-free: injecting $\log p_t$ or q_t into the softmax was found to collapse all K_i anchors toward the same high-score tokens. The informative set is

$$I_t = \{\tilde{v}_{t,j} \mid j \in I_t^{\text{idx}}\}, \qquad |I_t| = K_i.$$

A.5 Teacher sequence

The teacher hypothesis set for step t is serialized as

$$T_t = [G_t;\, I_t] \in \mathbb{R}^{N \times d_h}, \qquad N = K_g + K_i = 20.$$

The K_g coverage tokens are placed before the K_i informative ones, a coarse-to-specific ordering that matches a causal decoder's left-to-right view of the latent tokens.

A.6 Latent-placeholder assembly

At every reasoning-step position, we insert $N = 20$ special latent placeholder tokens between the learned boundary markers,

$$\langle \texttt{lat_start} \rangle,\ \texttt{PAD}_1, \ldots, \texttt{PAD}_N,\ \langle \texttt{lat_end} \rangle.$$

The number of placeholders is fixed at training time, so all K_g coverage targets and all K_i informative targets are realized teacher targets; the latent objective reduces to a deterministic cosine over a known target set without two-level masking. At inference and at the RL stage we use $K_{\text{eval}} = 8$ slots per step, matching the deployment configuration on which downstream accuracy peaks.

A.7 Norm-aligned teacher injection

Teacher targets $\tau_{t,i} \in T_t$ live in the vision-encoder post-merger space, which shares dimensionality with the LLM input embedding space but not its scale. Let $\mu_e = \mathbb{E}_w[\|e_w\|_2]$ denote the mean L_2 norm of the LLM input embedding matrix, computed once at initialization. Each teacher vector is rescaled in a direction-preserving way,

$$\tau_{t,i}^{\text{aligned}} = \mu_e \cdot \frac{\tau_{t,i}}{\|\tau_{t,i}\|_2 + \varepsilon}, \tag{14}$$

and replaces the input embedding at the corresponding latent placeholder. The boundary-marker embeddings are kept at their learned values.

A.8 EMA teacher

Teacher construction requires h_t, which itself co-evolves with the student during training. To decouple the supervision signal from high-frequency student updates, we maintain an exponential moving average of the student parameters, following the mean-teacher framework (Tarvainen and Valpola, 2017)

$$\theta_{\text{tea}} \leftarrow \tau\, \theta_{\text{tea}} + (1-\tau)\, \theta_{\text{stu}}, \qquad \tau = 0.999. \tag{15}$$

At every step a single EMA forward (with the visual encoder frozen) produces h_t, V_t, and $\{A_t^{(h)}\}$ in one pass; teacher targets T_t are then constructed from these.

A.9 Stage-1 objective

Let $\mathcal{S}_{\text{text}}$ and $\mathcal{S}_{\text{marker}}$ denote the sequence positions of ordinary text tokens and boundary markers, respectively. The text objective is

$$\mathcal{L}_{\text{text}} = -\sum_{\ell \in \mathcal{S}_{\text{text}} \cup \mathcal{S}_{\text{marker}}} \log p_\theta(y_\ell \mid y_{<\ell}, x, q), \tag{16}$$

with labels at latent and image positions set to the ignore index; the boundary markers remain supervised so that the model learns when to open and close the latent tokens.

Let $\mathcal{S}_{\text{lat}}$ collect the (t, i) pairs indexing every latent slot in the sequence, and let $z_{t,i-1}$ denote the student's last-layer hidden state at the token immediately preceding the i-th slot of step t (for $i = 1$ this is the position of $\langle \texttt{lat_start} \rangle$). The shifted-cosine latent objective is

$$\mathcal{L}_{\text{lat}} = \frac{1}{|\mathcal{S}_{\text{lat}}|} \sum_{(t,i) \in \mathcal{S}_{\text{lat}}} \big(1 - \cos(z_{t,i-1},\, \tau_{t,i}^{\text{aligned}})\big). \tag{17}$$

Direction alignment is the only degree of freedom because (14) already matches norms.

The Stage-1 loss combines the two with a linear warm-up on the latent weight,

$$\mathcal{L}_{\text{Stage1}} = \mathcal{L}_{\text{text}} + \lambda_{\text{lat}}(s)\, \mathcal{L}_{\text{lat}}, \tag{18}$$

$$\lambda_{\text{lat}}(s) = \min\big(s/0.1\, S_{\text{total}},\ 1\big)\, \lambda^{\star}_{\text{lat}}, \tag{19}$$

where s is the optimization step. The warm-up prevents early-training cosine gradients from overwhelming $\mathcal{L}_{\text{text}}$ while the student has not yet learned to use the latent placeholders. In implementation, the alignment in (17) corresponds to comparing `outputs.hidden_states[..., :-1, :]` with the assembled `inputs_embeds[..., 1:, :]` under the latent-placeholder mask.

A.10 Stage-2 objective

Rollout. For each prompt (x, q) we draw G rollouts from the Stage-1 policy $\pi_{\theta_{\text{old}}}$. Each rollout g consists of a token sequence $\mathbf{y}^{(g)}$ (interleaving text and ⟨`lat_pad`⟩ placeholders) and the latent hidden states $\mathbf{z}^{(g)} = \{z^{(g)}_{t,i}\}_{(t,i)\in\mathcal{S}_{\text{lat}}}$ emitted at every latent slot. Latent states are captured directly from the decoder loop.

Reward and advantage. Each rollout receives a scalar reward

$$R_g = \alpha_o \mathbb{1}[\text{answer}_g{=}\text{gold}] + \alpha_f \mathbb{1}[\text{format}_g \text{ valid}], \tag{20}$$

combining outcome correctness with a light format term. The advantage is the group-relative normalization

$$\hat{A}_g = \frac{R_g - \text{mean}(R_{1:G})}{\text{std}(R_{1:G}) + \epsilon}. \tag{21}$$

Text policy term. On text positions we apply the standard clipped PPO objective to token log-probabilities,

$$\begin{aligned}\mathcal{L}^{\text{RL}}_{\text{text}} = -\,\mathbb{E}_{g,\ell}\big[\min\big(&r^{(g)}_{\ell}\,\hat{A}_g,\\ &\text{clip}(r^{(g)}_{\ell}, 1{-}\varepsilon, 1{+}\varepsilon)\,\hat{A}_g\big)\big],\end{aligned} \tag{22}$$

with $r^{(g)}_{\ell} = \pi_\theta(y^{(g)}_{\ell} \mid y^{(g)}_{<\ell}) / \pi_{\theta_{\text{old}}}(y^{(g)}_{\ell} \mid y^{(g)}_{<\ell})$. MVH introduces no modification to text-token optimization.

Advantage-signed latent alignment. Latent slots have no discrete log-probability. We re-inject each rollout's captured latent states $z^{(g)}_{t,i}$ (detached, no gradient) at the slot positions of $\mathbf{y}^{(g)}$ and run a training forward pass under θ to obtain on-policy slot states $z^{\theta}_{t,i}$. Advantage is then propagated through cosine alignment,

$$\mathcal{L}^{\text{RL}}_{\text{lat}} = -\,\mathbb{E}_{g,(t,i)}\big[\hat{A}_g \cdot \cos\big(z^{\theta}_{t,i},\ z^{(g)}_{t,i}\big)\big], \tag{23}$$

so that positive-advantage rollouts pull the policy toward the rollout trajectory and negative-advantage rollouts push it away, mirroring the reinforce/suppress semantics of the text-token PPO term.

Latent trust region. Equation (23) alone is direction-unconstrained for negative advantages. We anchor the policy to the frozen Stage-1 checkpoint θ_{SFT}: a reference forward pass under θ_{SFT} on the same rollout sequence yields anchor states $z^{\text{SFT}}_{t,i}$, and we add

$$\mathcal{L}^{\text{RL}}_{\text{anchor}} = \mathbb{E}_{g,(t,i)}\big[1 - \cos\big(z^{\theta}_{t,i},\ z^{\text{SFT}}_{t,i}\big)\big]. \tag{24}$$

This plays the continuous-space role that PPO's reference-KL plays on the text distribution: latent updates are bounded to a direction-only neighborhood of the MVH-aligned Stage-1 manifold.

Full Stage-2 loss.

$$\mathcal{L}_{\text{Stage2}} = \mathcal{L}^{\text{RL}}_{\text{text}} + \lambda^{\text{RL}}_{\text{lat}}\,\mathcal{L}^{\text{RL}}_{\text{lat}} + \beta_{\text{anchor}}\,\mathcal{L}^{\text{RL}}_{\text{anchor}}. \tag{25}$$

The three terms act on disjoint aspects of the policy: $\mathcal{L}^{\text{RL}}_{\text{text}}$ keeps text-token selection inside the standard PPO framework; $\mathcal{L}^{\text{RL}}_{\text{lat}}$ steers the continuous latent representation under the same GRPO advantage; and $\mathcal{L}^{\text{RL}}_{\text{anchor}}$ bounds the latent update within a trust region around the Stage-1 checkpoint. All weights and clip parameters are listed in Table A1.

B Reproduction Details and Full Accuracy

Training data. The SFT corpus used by MVH-SFT, ILVR-Stage1, and ILVR-Stage2 is taken from Monet's released SFT data, which keeps the SFT comparison fully matched. Each sample interleaves text reasoning steps with helper images such as cropped regions or annotated views; during training we replace each helper image with $N = 20$ latent placeholder tokens and build the teacher targets from the original image, as described in Appendix A. Monet's RL corpus is not public, so we follow its construction pipeline on DeepEyes-v2 to assemble an RL set for MVH-RL. The RL-stage comparison between MVH-RL and Monet-RL is therefore matched in pipeline rather than in exact data, while the SFT comparison alone already establishes the accuracy ordering reported in the main text. Monet-SFT and Monet-RL checkpoints come from the original authors and are used without modification. Detailed hyperparameter settings are summarized in Table A1.

Quantity	Value / note
Shared	
Backbones	Qwen2.5-VL-7B (Bai et al., 2025b), Qwen3-VL-8B (Bai et al., 2025a)
Coverage tokens	$K_g = 10$
Informative hypotheses	$K_i = 10$
Stage-1 latent budget	$N = K_g + K_i = 20$
RL / inference latent budget	$K_{\text{eval}} = 8$
Rewrite blend	$\rho = 0.6$
EMA teacher decay	$\tau = 0.999$
Stage-1	
Latent-loss target weight	$\lambda^{\star}_{\text{lat}} = 1.0$
Latent-loss warm-up	first 10% of steps
Text-loss weight	1.0
Stage-2	
GRPO group size	$G = 8$
PPO clip range	$\varepsilon = 0.2$
PPO epochs per rollout	$E = 2$
Reward weights	$\alpha_o = 0.9, \alpha_f = 0.1$
Latent-loss weight	$\lambda^{\text{RL}}_{\text{lat}} = 0.5$
Anchor (trust-region) weight	$\beta_{\text{anchor}} = 0.1$

Table A1: Core MVH hyperparameters used in the experiments. The query temperature $\tau_q = d_h^{-1/2}$ and the rescaling target μ_e are design constants rather than tunable hyperparameters.

Evaluation protocol. Evaluation runs through VLMEvalKit (Duan et al., 2025) on V*, HRBench-4K, HRBench-8K, and MME-RealWorld-Lite. All six method-stage settings share an inference latent budget of $K_{\text{eval}} = 8$ slots per step, which matches the budget that gave the highest downstream accuracy in our preliminary sweep. Scoring follows each benchmark's official multiple-choice protocol, and per-sample correctness is the unit used by the marker-only coverage (Appendix F.5) and the cross-example swap (Appendix F.4) analyses below. The benchmark-averaged numbers cited in the main text are unweighted means over the four benchmarks. All reported results are averaged over multiple independent runs to reduce evaluation variance.

Full per-benchmark accuracy. Table A2 gives the full version of Table 1 from the main text, adding sub-category scores (Attribute/Spatial on V*, FSP/FCP on HRBench, Reasoning/Perception on MME-RealWorld-Lite) and comparisons against GPT-4o, DeepEyes (Zheng et al., 2026), Pixel Reasoner (Su et al., 2025), and LVR. We follow Monet's training setup for this table: MVH-SFT runs for 3 epochs and MVH-RL for 2 epochs, and ILVR-Stage1 and ILVR-Stage2 each run for 3 epochs.

Fine-grained perception (CoMT). Table A3 reports results on CoMT (Cheng et al., 2025), a fine-grained perception benchmark with four sub-tasks: creation, deletion, selection, and update. We train MVH on the CoMT training data released by ILVR, and the MVH-SFT stage follows the same training setup as ILVR-Stage1 to keep the two SFT-stage methods directly comparable. Following ILVR's training setup, MVH-SFT and ILVR-Stage1 each run for 15 epochs, and MVH-RL and ILVR-Stage2 each run for 2 epochs. Results are reported on both Qwen2.5-VL-7B and Qwen3-VL-8B backbones.

C Ablation Studies

We perform two ablations on MVH's design choices. The first targets the rewriting blend weight ρ used in anchor rewriting, with all variants trained from a Qwen2.5-VL-7B initialization. $\rho = 0.6$ achieves the best benchmark-averaged accuracy (Table A4). The second targets the latent RL design, with all variants initialized from the MVH-SFT checkpoint. Our advantage-signed latent alignment outperforms applying vanilla GRPO directly to the latent positions (Table A5).

D Structural Sanity Diagnostics

These diagnostics support the probe-legitimacy claim in §3.1: before running causal interventions, we verify that MVH latents are nontrivial enough for the slot-content memory hypothesis to be worth testing. They are not intended to prove content causality.

Cross-sample latent diversity. For each setting we dump the latent hidden states on 100 V* samples. At every latent slot index k, we then compute the 100×100 cosine-similarity matrix of the slot-k states across samples. A latent that varies with the input produces visible off-diagonal structure and contrast between diagonal and off-diagonal entries; a latent that collapses to a near-constant representation produces a uniformly high matrix. MVH-SFT, MVH-RL and ILVR-Stage1 show strong cross-sample diversity at every slot index, while ILVR-Stage2, Monet-SFT, and Monet-RL stay close to a uniform matrix, consistent with cross-sample collapse.

Within-sample block structure. For each sample we compute the $K \times K$ cosine matrix between its K latent slots and average this matrix across 100

Model	V*			HRBench4K			HRBench8K			MME-RealWorld-Lite		
	Overall	Attribute	Spatial	Overall	FSP	FCP	Overall	FSP	FCP	Overall	Reasoning	Perception
Proprietary Model												
GPT-4o	67.5*	72.2*	60.5*	59.0*	70.0*	48.0*	55.5*	62.0*	49.0*	52.0*	48.3*	54.4*
Open-Source Model												
Qwen2.5-VL-7B	76.4	77.4	75.0	68.0	80.3	55.8	63.8	73.8	53.8	45.8	39.7	49.6
+ vanilla SFT	81.7*	83.5*	79.0*	68.4*	78.3*	**58.5***	61.6*	70.8*	52.5*	51.3*	46.4*	54.4*
+ vanilla SFT + GRPO	78.5*	78.3*	79.0*	70.0*	83.3*	56.8*	66.8*	78.0*	55.5*	52.4*	48.1*	55.2*
PixelReasoner	80.6*	83.5*	76.3*	72.9*	86.0*	60.3*	66.9*	80.0*	54.3*	49.7*	44.5*	53.1*
Deepeyes	83.3*	84.4*	81.6*	71.3*	83.8*	58.8*	65.1*	77.0*	53.3*	54.3*	50.5*	56.6*
LVR	81.7*	**84.4***	77.6*	70.8*	83.8*	57.8*	63.8*	74.5*	51.5*	50.6*	42.7*	55.7*
ILVR-Stage1	80.6	84.3	75	70.0	87.5	52.5	66.5	80.3	52.8	49.2	44.0	52.5
ILVR-Stage2	80.6	84.3	75	70.3	87.8	52.8	66.6	81.8	51.5	50.4	45.3	53.6
Monet-SFT	79.6	81.7	76.3	70.4	84.0	57.0	63.0	78.3	49.5	52.9	48.4	55.8
Monet-RL	82.2	83.5	80.3	71.0	85.8	56.3	66.0	80.0	52.0	51.2	46.5	54.2
Our Model												
MVH-SFT	82.2	82.6	81.6	71.5	**88.5**	54.5	67.1	**82.5**	51.8	50.4	45.3	53.7
MVH-RL	**83.2**	82.6	**84.2**	**72.4**	88.3	56.5	**69.5**	**82.5**	**56.5**	**53.8**	**49.6**	**56.5**

Table A2: **Full per-benchmark accuracy including sub-category scores and additional comparisons.** Settings analyzed in this paper are in the lower block. Gray rows mark reasoning-with-tool methods. The **bold numbers** indicate the best performance achieved by each setting, and the underline numbers are the second best. PixelReasoner and DeepEyes are excluded from this ranking. Stars (*) indicate values reproduced from prior reports; dashes indicate scores not reported on the corresponding benchmark.

samples. Under the MVH design $K{=}20$ slots are split into a coverage half and an informative half, and a 2×2 block pattern is expected in this matrix (within-half similarity higher than cross-half similarity). The Block Sensitivity Ratio (BSR) is the ratio of mean within-block similarity to mean cross-block similarity; values above one indicate that the two halves have been learned as functionally distinct subsets. MVH obtains BSR ≈ 1.13, while ILVR and Monet do not have a designed block structure and their first/second-half split is reported only as a positional reference.

Role separation across questions. The within-sample analysis cannot distinguish coverage from informative slots by function; it only checks that the two halves are not identical. To target the designed role separation, we take 20 images with three to five different questions each, dump latents for every (image, question) pair, and compute the cross-question similarity inside the coverage half and inside the informative half separately. The coverage half is expected to be query-independent (high cross-question similarity) and the informative half query-dependent (lower cross-question similarity); the Role Separation Ratio (RSR) is their ratio. MVH reaches RSR ≈ 1.12 on this set, while ILVR and Monet remain near 1.00 since they have no designed role split.

E Intervention Suite

Intervention conditions. Table A6 summarizes the latent-interface interventions used throughout the paper. The conditions are designed to separate four factors: the presence of the latent format, the precise contents of latent slots, the identity of learned boundary markers, and malformed execution when a corrupted span remains in the sequence.

Implementation of interventions. All interventions are applied at the input-embedding level, after token assembly but before the first decoder block. $K{=}0$ keeps the boundary-marker positions and drops the latent placeholders entirely, so the model decodes from ⟨`lat_start`⟩ immediately followed by ⟨`lat_end`⟩. NoLatent additionally bans the boundary markers through a logits processor, producing a clean removal of the latent format. ZeroSlot, RandomSlot, and FixedSlot replace the slot input embeddings with zeros, Gaussian noise at the matching scale, or a position-conditioned fixed vector while leaving the marker positions and sequence layout untouched. ZeroMarker zeros the marker input embeddings while keeping the marker positions in place, which lets the corrupted span propagate through generation. ZeroAll combines slot and marker zeroing. The forced-pad-without-markers condition inserts a bare slot span with no enclosing markers and is treated as an out-of-distribution stress test rather

Methods	COMT					COMT				
	Creation	Deletion	Selection	Update	**Avg.**	Creation	Deletion	Selection	Update	**Avg.**
Backbones	*Qwen2.5-VL-7B*					*Qwen3-VL-8B*				
Standard Baselines										
Zero-shot	68.0	38.0	35.0	14.0	38.8	**89.0**	32.0	15.0	24.0	40.0
Direct-SFT	52.0*	60.0*	51.0*	49.0*	53.0*	**89.0***	67.0*	49.0*	53.0*	64.5*
CoT-SFT	**80.0***	52.0*	45.0*	46.0*	55.8*	83.0*	62.0*	49.0*	44.0*	59.5*
Latent Reasoning										
Mirage-Stage1	53.0*	54.0*	45.0*	42.0*	48.5*	81.0*	58.0*	43.0*	50.0*	58.0*
Mirage-Stage2	65.0*	62.0*	47.0*	50.0*	56.0*	84.0*	66.0*	54.0*	57.0*	65.3*
ILVR-Stage1	74.0	62.0	39.0	42.0	54.3	83.0	65.0	43.0	53.0	61.0
ILVR-Stage2	68.0	**68.0**	48.0	44.0	57.0	86.0	68.0	54.0	57.0	66.3
Our Model										
MVH-SFT	71.0	65.0	48.0	46.0	57.5	88.0	70.0	59.0	61.0	69.5
MVH-RL	73.0	67.0	**53.0**	**53.0**	**61.5**	**89.0**	**72.0**	**63.0**	**64.0**	**72.0**

Table A3: **CoMT benchmark accuracy across Qwen2.5-VL-7B and Qwen3-VL-8B backbones.** Four sub-tasks (creation, deletion, selection, update) are evaluated separately and as an unweighted average. The **bold numbers** indicate the best performance achieved by each setting, and the underline numbers are the second best.

ρ	V*	HR-4K	HR-8K	MME-RW	Avg.
0.2	**82.4**	71.3	67.1	50.2	**67.8**
0.4	**82.4**	71.3	66.8	**50.4**	67.7
0.6 (MVH-SFT)	82.2	**71.5**	67.1	**50.4**	**67.8**
0.8	81.0	71.2	**67.2**	49.9	67.3
1.0	80.8	70.9	66.9	50.1	67.2

Table A4: Sensitivity to the rewriting blend weight ρ. Benchmark-averaged accuracy on V*, HRBench-4K, HRBench-8K, and MME-RealWorld-Lite. The highlighted row is the MVH-SFT setting. The **bold numbers** indicate the best performance achieved by each MLLM, and the underline numbers are the second best.

Method	V*	HR-4K	HR-8K	MME-RW	Avg.
Vanilla GRPO	80.6	**73**	66.25	52.9	68.2
Ours (MVH-RL)	**83.2**	72.4	**69.5**	**53.8**	**69.7**

Table A5: Comparison between vanilla GRPO and our advantage-signed latent alignment on the same SFT checkpoint. Benchmark-averaged accuracy on V*, HRBench-4K, HRBench-8K, and MME-RealWorld-Lite. The **bold numbers** indicate the best performance achieved by each setting.

than a clean comparison.

Sensitivity metrics. We derive four scalar sensitivities from the seven principal conditions, each of which condenses a row of the intervention matrix into a one-row-per-setting summary. Slot-content sensitivity is Normal − avg(ZeroSlot, RandomSlot, FixedSlot), which is close to zero in every setting. Marker sensitivity is Normal − ZeroMarker, and it separates the three marker-dependence ranges discussed in §5.3. Latent-format dependence is Normal − NoLatent, which captures how much accuracy depends on the latent tokens as a whole. Malformed interference is NoLatent − ZeroMarker, which compares the cost of keeping a corrupted span in the sequence against the cost of removing the span cleanly; a positive value indicates that the model has a learned protocol for the latent tokens and is hurt more by a corrupted version than by a clean removal.

Full intervention matrix. Figure A2 expands Figure 4 with raw benchmark-averaged accuracy, accuracy drops relative to Normal, and the derived sensitivity metrics used to define the marker-dependence ranges.

F Slot-Content Memory Tests: Detailed Results

This appendix expands the four tests of slot-content memory reported in §4, following the order of tests in §4.1. Each test uses a different probing assumption: a static perturbation test on slot contents, an image-free injection test, a cross-example swap test, and a sample-level marker-only coverage test. The subsections below give the protocol and the data behind the corresponding row of the consolidated content-tests table.

F.1 Overview

Table A7 consolidates the four tests into a single predicted-versus-observed view.

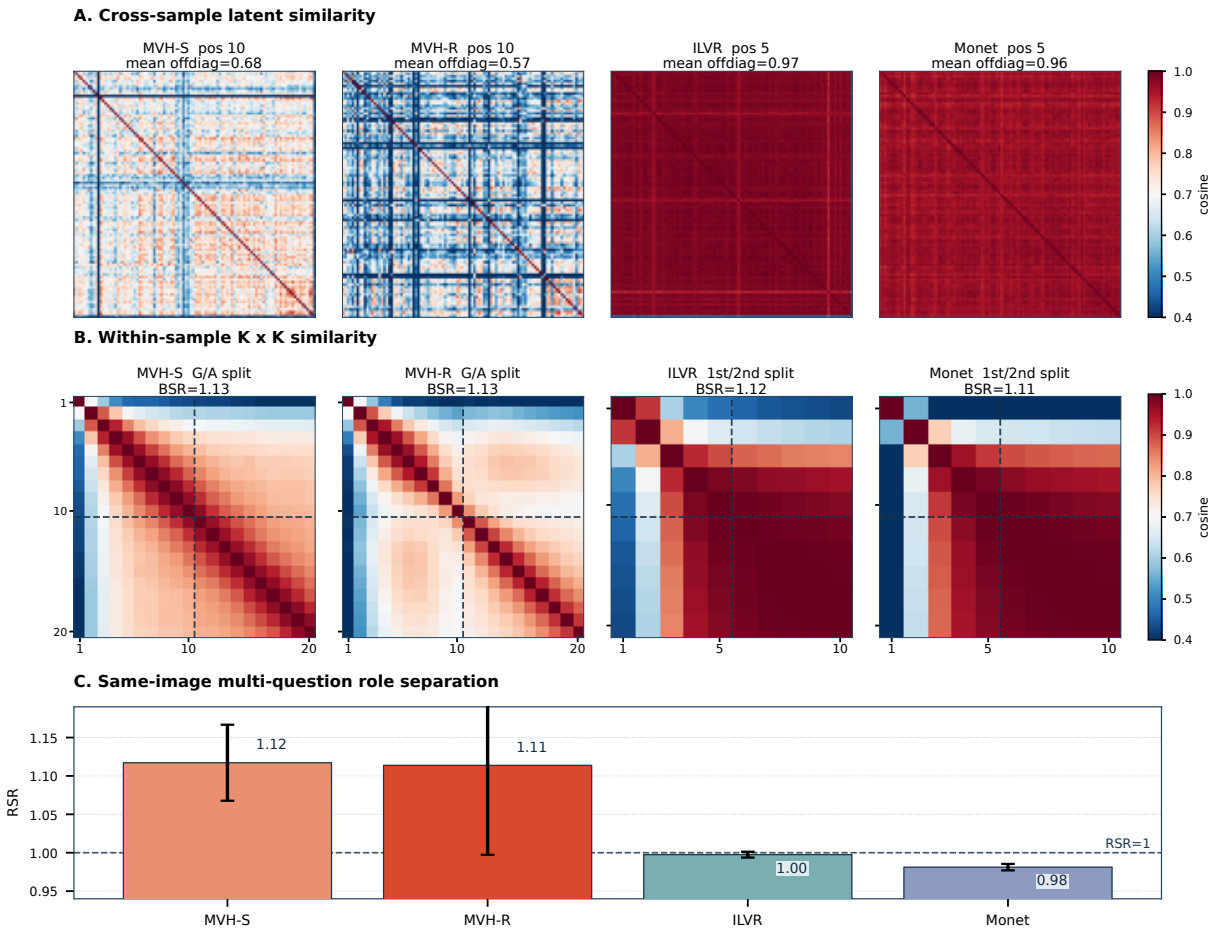

Figure A1: **Structural diagnostics for latent organization.** (A) Cross-sample latent similarity shows that MVH latents vary more across inputs than ILVR and Monet, which are closer to cross-sample collapse. (B) Within-sample $K \times K$ similarity recovers the intended coverage/anchor split in MVH; for ILVR and Monet, the first/second-half split is only a positional reference, not a designed role split. (C) Same-image multi-question analysis quantifies role separation with RSR, the ratio between coverage-state and anchor-state similarity across questions. MVH has RSR above 1.1, while ILVR and Monet remain near 1.0.

F.2 Slot-Value Perturbation

This test asks whether the contents of the slots are causally responsible for the latent tokens' gain by replacing them with substitutes that retain the sequence layout. ZEROSLOT sets the slot input embeddings to zero, RANDOMSLOT samples Gaussian noise at the matching scale, and FIXEDSLOT reuses a position-conditioned vector that is independent of the input. The benchmark-averaged accuracy under these three conditions is reported in Figure A2(A) as part of the full intervention matrix. Across the six settings, perturbing slot contents moves accuracy by at most 0.027, while the same models respond strongly to interventions on the boundary markers (§G). The slot-content sensitivity metric defined in §E collapses this into a single number per setting, which stays near zero in every case.

F.3 Image-Free Injection

As a stronger portability test, we remove the image and ask whether latent states extracted from an image-conditioned run can restore the missing visual information. The test is targeted at MVH-SFT, where the Stage-1 teacher provides direct visual-space supervision for the slot states.

The probe runs as a paired three-condition test on the V* subset that emits valid latent tokens. We first run MVH-SFT with the full image and capture the final-layer hidden state at every emitted latent slot ($n = 58$ samples). We then re-run the same prompt with the image replaced by zeros, both with the captured slot states injected at the corresponding positions ($n = 58$) and without any injection ($n = 57$ after a prompt-formatting filter). If the captured states encode the visual evidence that the answer needs, blind accuracy with injection should rise

Condition	Markers	Latent slots	Tested factor	Interpretation
NORMAL	kept	kept	Reference latent-span inference	Reference condition
$K = 0$	kept	removed	Boundary markers without latent slots	Marker-only contribution
NOLATENT	disabled	disabled	Clean removal of the latent interface	Latent-format dependence
ZEROSLOT	kept	zeroed	Slot-state values	Slot-content sensitivity
RANDOMSLOT	kept	random	Meaningful slot contents	OOD slot-content perturbation
FIXEDSLOT	kept	fixed by position	Input-dependent slot variation	Content-variation control
ZEROMARKER	zeroed	kept	Boundary-marker input embeddings	Malformed span with latent slots retained
ZEROALL	zeroed	zeroed	Joint marker and slot corruption	Malformed span without usable contents
FORCEDSLOTS w/o MARKERS	disabled	forced	Bare latent slots without boundary markers	OOD stress test

Table A6: **Latent-interface intervention conditions.** The intervention suite separates clean latent removal, slot-content perturbation, boundary-marker corruption, and bare-slot stress tests. $K = 0$ keeps learned boundary markers while removing latent slots; NOLATENT removes the span cleanly; ZEROMARKER zeroes marker input embeddings while keeping the corrupted marker positions in the sequence. Forced pads without markers are treated as an out-of-distribution stress test.

Test	Slot-content prediction	Observed	Verdict
Slot-content perturbation	Zero/random/fixed slot contents should cause a large drop.	Benchmark-averaged slot-content sensitivity is near zero across MVH, ILVR, and Monet.	Weak slot-content causality.
Latent injection	Injected latents should recover image-conditioned accuracy.	MVH-SFT recovers only 7.8% of the missing-image gap	Not recoverable memory.
Image-quality bottleneck	Latents should compensate when visual input is degraded.	With-latent and blind curves show no consistent mid-quality kick-in; the latent contribution is broadly stable across quality levels rather than selectively activated.	No standalone visual buffer.
In-distribution swap	Real donor slot states should transfer donor content or answers.	Max swap deviation is small; donor adoption is near/below chance.	No in-distribution content transfer.
Marker-only coverage	Slot contents should explain the incremental latent gain.	K=0 recovers most Normal-vs-NO_LATENT unique solves.	Format/protocol dominates.

Table A7: Tests of the visual-memory account. Even under content-favorable MVH supervision, slot contents do not behave like recoverable visual memories.

toward the full-image level.

Table A8 shows that injection recovers only a small fraction of the full-image gap. The result does not imply that the latent tokens are useless; rather, it shows that the image-conditioned slot states are not portable enough to restore visual performance when the image signal is removed.

Image-quality bottleneck. As a continuous-degradation version of the same test, we degrade the image gradually and ask whether the latent tokens compensate for missing visual evidence. A standalone latent visual buffer would predict a selective separation between with-latent and no-latent runs at intermediate image quality. We sweep image quality across six levels (blanked at 0.0, then 0.2, 0.3, 0.5, 0.8, and the original image at 1.0), apply matched downsampling and blur at each level, and run paired with-latent and no-latent decodes on V*, HRBench-4K, HRBench-8K, and MME-RealWorld-Lite for every setting (Figure A3).

F.4 Cross-Example Slot Swap

Zero, random, and fixed slot-content perturbations may be criticized as out-of-distribution corruptions. We therefore run an in-distribution swap test that uses real model executions as donors. We force a fixed latent prefix at the end of each prompt (⟨lat_start⟩, K slot placeholders, ⟨lat_end⟩), capture the slot hidden states at a chosen decoder

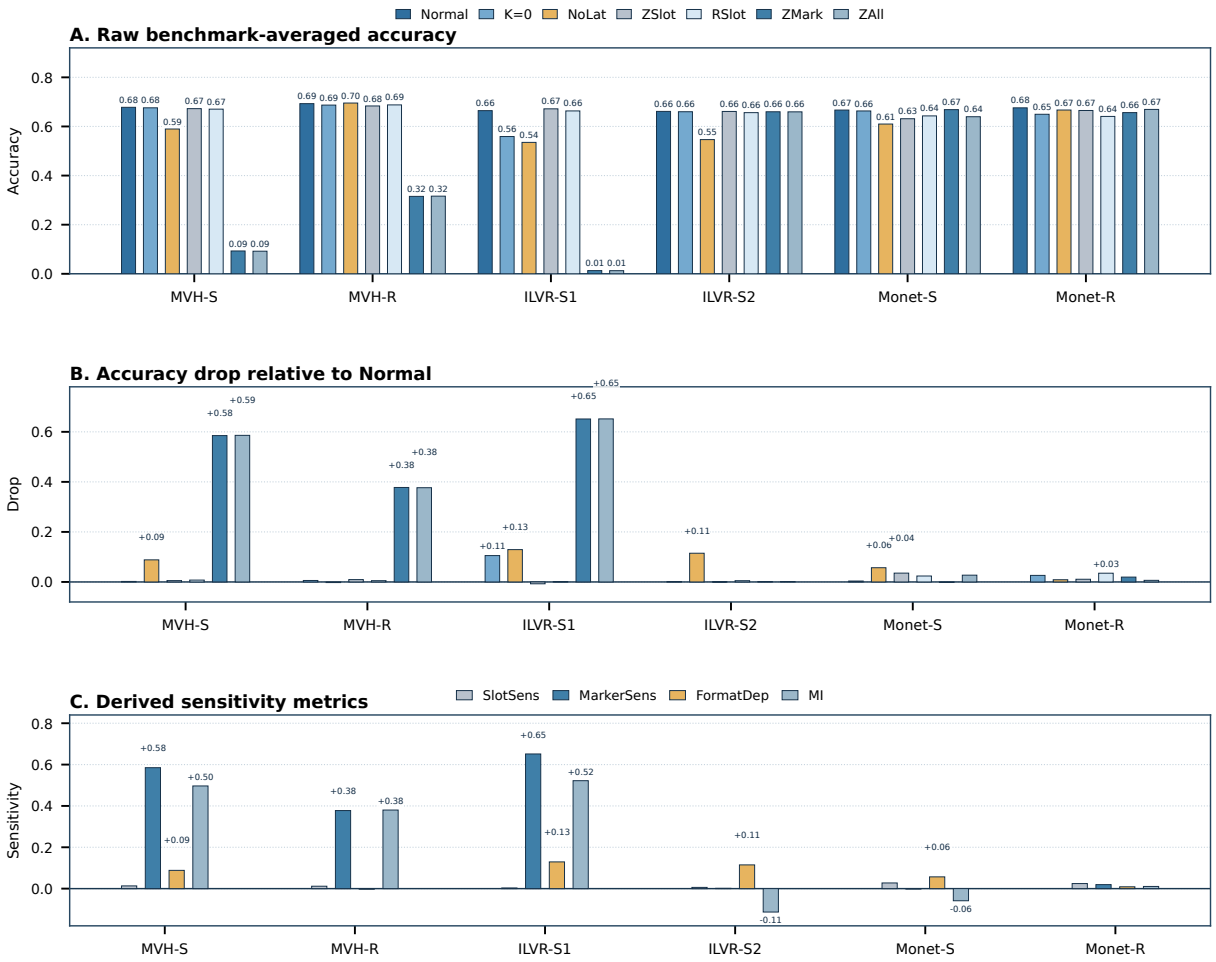

Figure A2: **Full latent-interface intervention matrices.** (A) Benchmark-averaged raw accuracy under each intervention condition. (B) Accuracy drop relative to the Normal condition. (C) Derived sensitivity metrics used to classify marker-dependence ranges. Slot-content perturbations remain close to Normal, whereas ZeroMarker and ZeroAll selectively collapse MVH-SFT, MVH-RL, and ILVR-Stage1.

layer during a clean forward pass, and re-inject the captured states at the same positions of a target sample. The protocol keeps marker positions, slot positions, and decoder layer identical across donor and target, so any change observed after the swap is attributable to the slot contents rather than to the layout. We test three decoder depths: an early layer (L5), a mid layer (L14), and a late layer (L27). Each (setting, layer) combination is evaluated under four donor conditions: a self-swap sanity check (target equals donor), a label-matched cross-sample swap (different image but the same multiple-choice answer letter), a label-mismatched cross-sample swap (different image and different answer), and a random donor baseline. The resulting grid is 6 settings $\times$ 3 layers $\times$ 4 conditions $=$ 72 sub-jobs on the same 57-sample V* subset used by the marker controls.

Table A9 verifies that self-swap leaves forced-prefix accuracy unchanged, validating the hook protocol. Figure A4 then shows that cross-sample donor states neither substantially move target accuracy nor transfer donor answers. The same-image cross-question control is particularly diagnostic: even when donor and target share the same visual context, replacing slot states does not cause the decoder to adopt the donor question's answer.

F.5 Marker-Only Coverage by Benchmark

Aggregate accuracy does not distinguish whether the same examples are solved across intervention conditions. We therefore report marker-only coverage at the sample level. Let S_{Normal} be the set of examples solved by normal latent decoding, $S_{K=0}$ the set solved when only the boundary markers are kept and the slots are dropped, and S_{NoLatent} the set solved when the entire latent tokens are removed. The latent tokens' unique solved set is

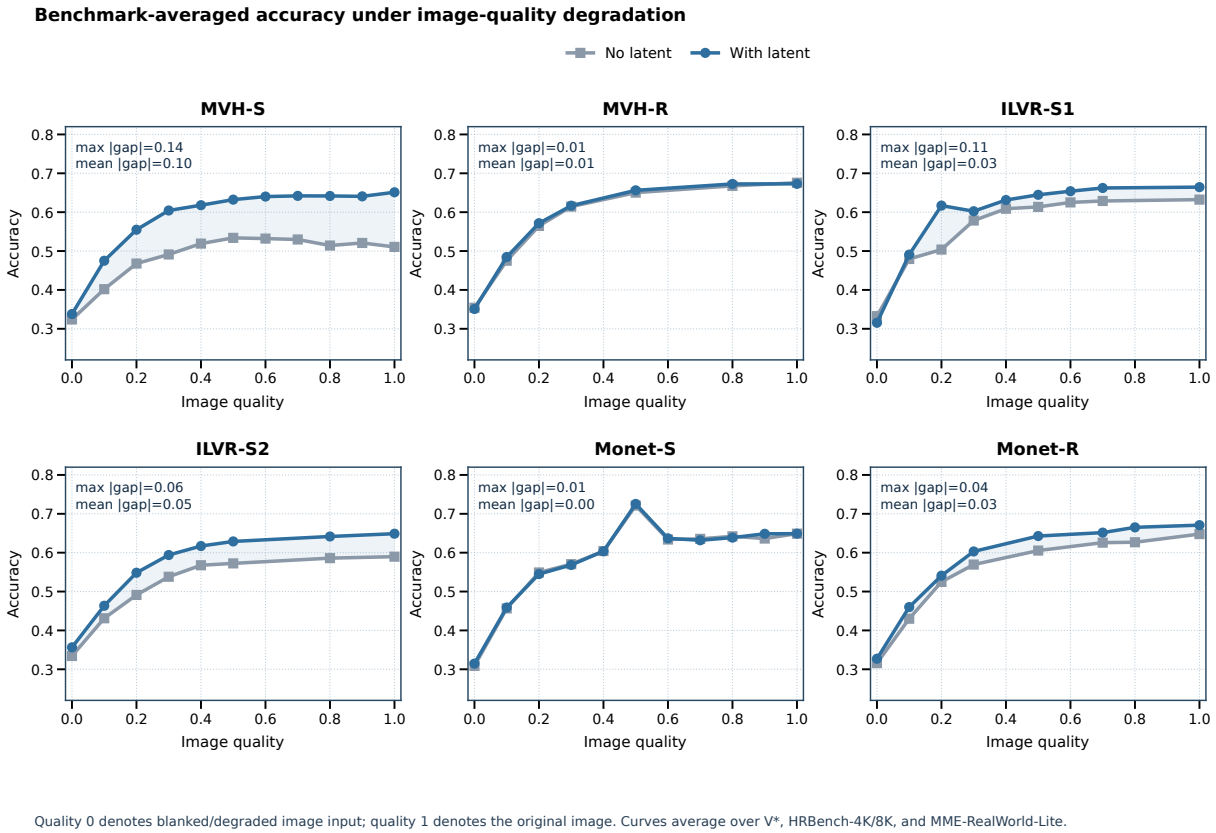

Figure A3: **Image-quality bottleneck curves.** Accuracy is measured as image quality is progressively degraded from blanked input (0.0) to the original image (1.0). Curves report paired benchmark averages over V*, HRBench-4K/8K, and MME-RealWorld-Lite for with-latent and no-latent decoding. The curves show no consistent mid-quality kick-in pattern: some settings have a persistent offset, while others nearly overlap, but the latent tokens do not behave like an independent visual buffer that selectively compensates for missing image information.

$S_{\text{Normal}} \setminus S_{\text{NoLatent}}$, and marker-only coverage is the fraction of that set that is also solved by marker-only decoding:

$$\text{MOC} = \frac{|S_{K=0} \cap (S_{\text{Normal}} \setminus S_{\text{NoLatent}})|}{|S_{\text{Normal}} \setminus S_{\text{NoLatent}}|}. \quad (26)$$

MOC counts examples rather than averaging accuracy, so a high value means that the same examples uniquely solved by the latent tokens are also solved by a marker-only decode. A value close to one identifies settings in which the latent tokens' unique contribution is largely covered by the boundary markers; a value near zero would identify settings in which slot contents matter for the uniquely solved subset.

Table A10 expands the aggregate MOC values reported in Figure 3. ILVR-Stage1 reaches complete marker-only coverage on all four benchmarks, and MVH remains high except for a lower but still substantial MME-RealWorld-Lite value for MVH-SFT. Monet-SFT is weaker, especially on MME-RealWorld-Lite, while Monet-RL has small incremental sets in V* and MME-RealWorld-Lite. High MOC does not imply that latent slots are functionally irrelevant; it shows that the examples uniquely gained by the latent tokens are usually not dependent on precise slot contents.

G Boundary-Marker Controls

The K=0 and marker-only coverage results show that learned boundary markers cover much of the latent-token gain. We next ask whether this is a generic effect of adding any special or pause-like tokens. To avoid ambiguity from free latent generation, we use forced-prefix decoding: a fixed prefix is appended to the prompt, and the model decodes only after that prefix.

Nine prefix variants are compared on the same 57-sample V* subset used by the swap analysis. The reference prefix uses the trained boundary markers without latent slots in between. From this reference, we test prefixes that replace the trained markers with two unused special tokens, with the model's ⟨bos⟩/⟨eos⟩ pair, with a textual delimiter

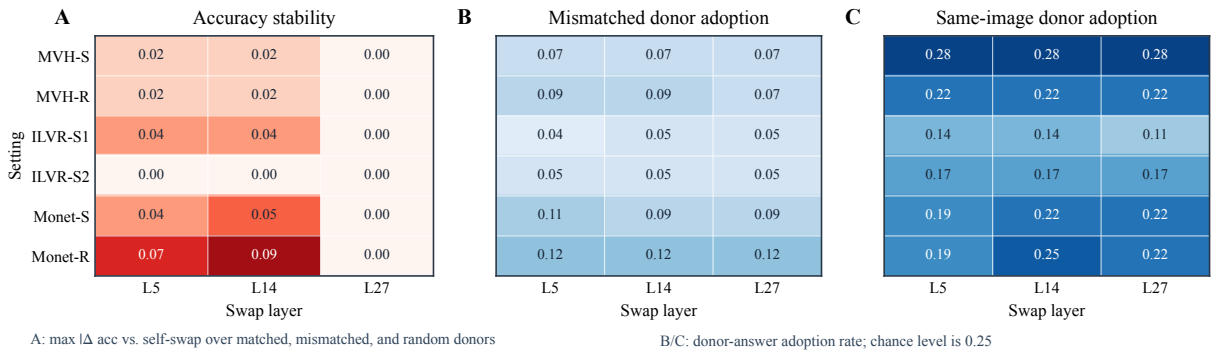

Figure A4: **Full in-distribution latent-swap results.** We capture slot states from forced-prefix runs and replace the target slot states at the same decoder layer, using layers L5, L14, and L27. (A) Accuracy deviations from self-swap remain small across the main V* swap grid, indicating that real donor slot states do not substantially rewrite target predictions. (B) For label-mismatched donors, donor-answer adoption stays well below the 25% chance level across settings and layers. (C) The same-image cross-question control asks whether slot states carry question-specific anchor information when the visual context is identical; donor-answer adoption remains near chance on the multi-question V* subset. Together, these results address the concern that slot-content perturbations are merely out-of-distribution corruptions: even in-distribution donor slot states do not behave as portable visual evidence.

Condition	Image / latent source	n	V* acc.
Full-image run	original image + generated latent span	58	0.845
Blind run	blank/zero image; latent disabled	57	0.228
Blind + injected latent	blank/zero image + image-conditioned latent states	58	0.276
Injected − Blind	–	–	+0.048
Recovery rate	gap fraction	–	0.078

Table A8: **Blind-run latent injection test for MVH-SFT on V*.** Image-conditioned latent states are extracted from a full-image run and injected into a blind run. If slot states acted as portable visual memories, injection should recover a large fraction of the full-image performance gap. Instead, injected latents improve accuracy only from 0.228 to 0.276, yielding a recovery rate of $(A_{\text{inj}} - A_{\text{blind}})/(A_{\text{full}} - A_{\text{blind}}) = 0.078$.

Setting	Self-swap@L14	TrueMk+Slot	Δ
MVH-SFT	0.842	0.842	+0.000
MVH-RL	0.842	0.842	+0.000
ILVR-Stage1	0.667	0.667	+0.000
ILVR-Stage2	0.825	0.842	-0.018
Monet-SFT	0.719	0.737	-0.018
Monet-RL	0.544	0.544	+0.000

Table A9: **Self-swap sanity check.** TrueMk+Slot denotes forced-prefix decoding with the trained boundary markers and the original slot states (no swap). Δ reports Self-swap@L14 minus the TrueMk+Slot accuracy. Replacing a target's slot states with its own captured states leaves forced-prefix accuracy unchanged up to small evaluation noise, validating the hook protocol before cross-sample donor swaps are interpreted.

such as [LATENT]/[/LATENT], with K period or newline tokens that act as pause-like fillers, or with K bare slots and no markers at all. Each marker variant is also tested with and without a length-matched run of latent slots between the markers, which separates the effect of marker identity from the effect of slot-level capacity at fixed length.

Figure A5 shows that marker identity matters most when the prefix contains only boundary-like tokens. For MVH-SFT, replacing learned markers with unused special tokens or <bos>/<eos> causes large drops relative to the true-marker K=0 condition. However, adding latent slots can recover much of the behavior even under dummy markers. This distinguishes two mechanisms: a learned routing role tied to marker identity, and a slot-level capacity role provided by latent slots. We therefore avoid the stronger claim that latent slots are useless; the causal claim concerns slot contents as portable visual memories.

H Visual-Subspace Alignment and Layer-wise Attention

The main text characterizes the relation between slot states and projected visual-token representations along two complementary axes. The first is a static geometric probe. For each V* sample with emitted latent tokens, we take the final-layer hidden state at every slot position and compare it, in co-

Setting	V*	HRBench-4K	HRBench-8K	MME-RW-Lite	Mean
MVH-SFT	1.00 (18)	0.98 (125)	0.98 (125)	0.78 (308)	0.94
MVH-RL	1.00 (7)†	0.97 (34)	0.98 (62)	0.91 (58)	0.97
ILVR-Stage1	1.00 (48)	1.00 (162)	1.00 (168)	1.00 (346)	1.00
ILVR-Stage2	1.00 (35)	0.92 (120)	0.91 (133)	0.83 (390)	0.92
Monet-SFT	0.90 (30)	0.88 (43)	0.87 (61)	0.45 (77)	0.78
Monet-RL	0.50 (2)†	0.61 (28)	0.62 (29)	0.00 (2)†	0.43

Table A10: **Per-benchmark marker-only coverage.** Each cell reports MOC with $n_{\text{inc}} = |S_{\text{Normal}} \setminus S_{\text{NoLatent}}|$ in parentheses. MOC is the fraction of the latent tokens' unique solved set that is also solved by marker-only decoding. The mean column is the unweighted average over benchmarks. † marks cells with $n_{\text{inc}} < 10$.

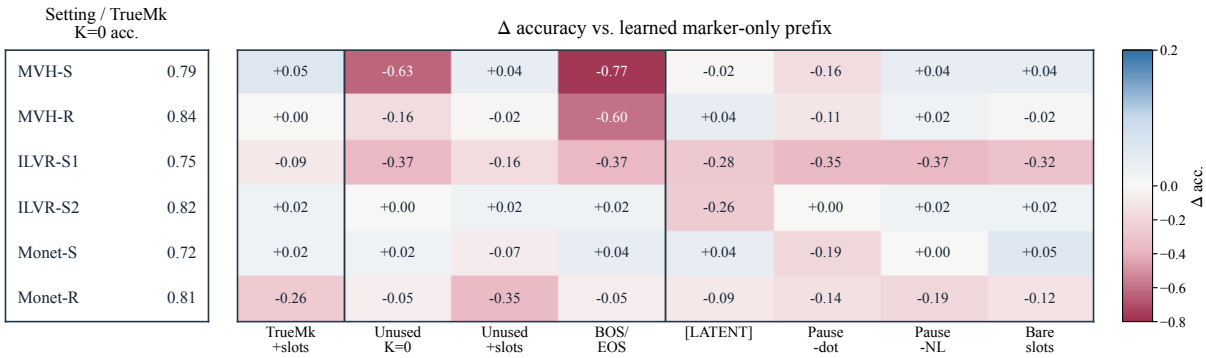

Figure A5: **Dummy-marker forced-prefix controls.** We force a latent-prefix at the end of the prompt and decode from that prefix on V* ($n = 57$). The left column reports accuracy under the learned marker-only prefix; heatmap cells report accuracy differences relative to that condition. Replacing learned markers with unused special tokens or <bos>/<eos> sharply degrades marker-dependent settings, especially MVH-SFT, showing that learned marker identity is not replaceable by arbitrary special tokens. Adding latent slots or using length-matched bare-slot prefixes can recover some behavior, separating marker-identity routing from latent-slot compute, timing, or formatting capacity.

sine similarity, with every post-merger visual token of the same image after projection into the LLM hidden space. The top-5 mean similarity averages over slots and across the V* subset (between 14 and 29 valid samples per setting depending on the emission rate), giving a single number per setting. ILVR-Stage1 reaches 0.58, ILVR-Stage2 relaxes to 0.27, MVH-SFT and MVH-RL sit between 0.10 and 0.15, and Monet stays near 0.07, which is indistinguishable from chance-level alignment. The per-setting values for both the coverage and informative halves appear in the concordance table (Appendix I).

The second probe is dynamic and decomposes attention to image patches at every decoder layer. For each decoder layer l, we compute the entropy gap

$$\Delta H_l = H_l^{\text{answer}} - H_l^{\text{latent}}, \tag{27}$$

where H_l^{latent} is the per-layer entropy of decoder attention to image tokens averaged over latent slot positions, and H_l^{answer} is the analogous quantity averaged over answer-token positions. Low entropy means attention concentrates on a small subset of image patches; positive ΔH_l means latent-step image attention is more focused than answer-step image attention at layer l. The main text reports the peak gap and peak layer for each setting; here we show the full layer-wise curve (Figure A6).

The full curves confirm that the visual-inspection mode is localized rather than uniformly distributed across layers. MVH and ILVR exhibit late-layer peaks, whereas Monet remains below $\Delta H = 1.0$. This supports the main-text claim that weak slot-content causality does not imply functional irrelevance: some latent tokens induce a focused visual-inspection mode even when their slot contents do not transfer as recoverable visual memory. All layer-wise attention analyses use the V* subset used for attention replay; full-sequence HRBench-8K replay was omitted due to memory cost (per-setting values in Table A11).

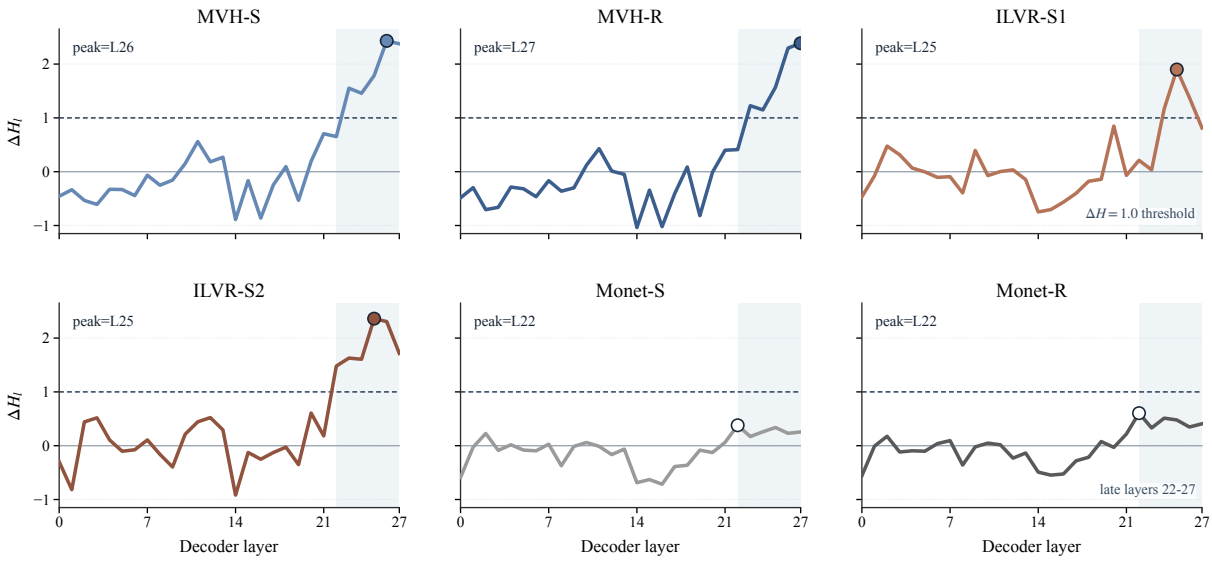

Figure A6: **Full layer-wise attention-concentration curves.** For each method-stage setting, we plot the per-layer image-attention entropy gap $\Delta H_l = H_l^{\text{answer}} - H_l^{\text{latent}}$, where larger values indicate that latent-step attention is more concentrated on image tokens than answer-step attention. The dashed horizontal line marks $\Delta H = 1.0$. MVH-SFT, MVH-RL, ILVR-Stage1, and ILVR-Stage2 show localized late-layer peaks, with peak layers in L25–L27 and peak gaps between 1.90 and 2.43. Monet-SFT and Monet-RL stay below $\Delta H = 1.0$. This expands the compact summary in Figure 5.

I Per-Setting Concordance Across Diagnostics

Table A11 expands the compact regime summary in Table 2 with the underlying per-setting values across all eleven diagnostics used to assign regime labels. The rows group into intervention-based diagnostics (marker sensitivity, malformed interference, marker-only coverage, marker-identity substitution under forced-prefix decoding, swap deviation), generation-behavior diagnostics under marker corruption (self-loop rate, latent blocks emitted), visual-subspace diagnostics (top-5 latent-to-visual similarity for the coverage and informative halves), and attention diagnostics (latent-step image-attention entropy, peak entropy gap). The bottom row reports the final V* and benchmark-averaged accuracy of each setting. Each column corresponds to one method-stage setting, and the columns are ordered by the marker-dependence ranges defined in §5.3.

The bottom-row accuracies are similar across columns, while the upper rows separate the three ranges along multiple axes. This is the form of evidence that supports the two-axis description in §5.3: final accuracy alone does not identify how a given latent tokens are being used.

J AI Assistance Statement

We used ChatGPT solely for light language polishing (grammar, word choice, and phrasing) on portions of the manuscript. All research ideas, experimental design, implementation, analysis, and scientific claims are the authors' own. The authors verified all AI-assisted text edits for accuracy and take full responsibility for the content of the paper.

	Strong marker dep.		Reduced marker dep.	Weak marker dep.		
Metric	MVH-SFT	ILVR-S1	MVH-RL	ILVR-S2	Monet-SFT	Monet-RL
Marker Sensitivity	0.585	0.651	0.378	0.001	-0.002	0.019
Malformed Interference	0.497	0.522	0.380	-0.113	-0.059	0.011
Marker-only coverage	78–100%	100%	91–100%	83–100%	45–90%	0–62%
Marker identity Δ	-0.77	-0.37	-0.60	+0.02	+0.04	-0.05
Self-loop under marker zero	100%	97%	100%	0%	0%	0%
Latent blocks under marker zero	$300\times$	$14\times$	$105\times$	$1\times$	$1\times$	$1\times$
Latent–visual top-5 sim, coverage	0.15	0.58	0.10	0.27	0.07	0.07
Latent–visual top-5 sim, anchor	0.11	0.52	0.10	0.24	0.07	0.06
Latent-step attention entropy	4.31	5.21	4.27	4.57	6.59	n=2
Peak ΔH	2.43	1.90	2.39	2.36	0.38	0.61
Swap max $\lvert\Delta_{self}\rvert$	0.018	0.035	0.018	0.000	0.053	0.088
Final accuracy, V^*/avg	.82/.68	.81/.67	.83/.70	.81/.67	.80/.67	.82/.68

Table A11: Full per-setting concordance across independent diagnostics. Slot-content tests, marker interventions, generation behavior, visual-subspace alignment, and attention diagnostics separate the six method-stage settings into three marker-dependence ranges. The bottom row shows that final accuracies are similar; the other rows show that the settings use the latent interface differently. Compact summary appears in Table 2.