ABLATE-TO-VALIDATE: ARE VISION-LANGUAGE MODELS REALLY USING CONTINUOUS THOUGHT TOKENS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models (VLMs) are increasingly augmented with continuous or latent non-textual tokens intended to support "visual reasoning." Improved task accuracy alone does not show that models use these tokens: gains may instead arise from added context length, special-token anchoring, or training-time regularization. We formalize a diagnostic principle, Ablate-to-Validate, and instantiate it as the Token Replacement Test (TRT), a standardized suite of content-replacement ablations that measures how sensitive task accuracy is to a latent span's content while holding span placement, token budget, prompt, image and decoding fixed. We apply TRT to three off-the-shelf visual reasoning methods, Mirage, Mull-Tokens and CoVT. As a controlled testbed, we also train our own LLaVA-13B and Qwen2.5-VL-3B models to estimate relative depth by predicting and consuming continuous or discrete depth tokens. Our results show that accuracy gains can be a misleading proxy for latent-token reasoning: across all methods with continuous visual tokens, accuracy changes little when latent-token content is replaced by distribution-matched noise or shuffled across slots, and injecting an oracle yields no measurable gain, revealing a gap between "having a latent channel" and relying on its content. This gap is bolder for continuous spans: under the same test, accuracy with discrete depth tokens is more sensitive to their content on both backbones. TRT therefore shows how much and where the content of visual reasoning tokens matters for accuracy, and we recommend reporting such diagnostics as standard practice. *Code will be released upon publication.*

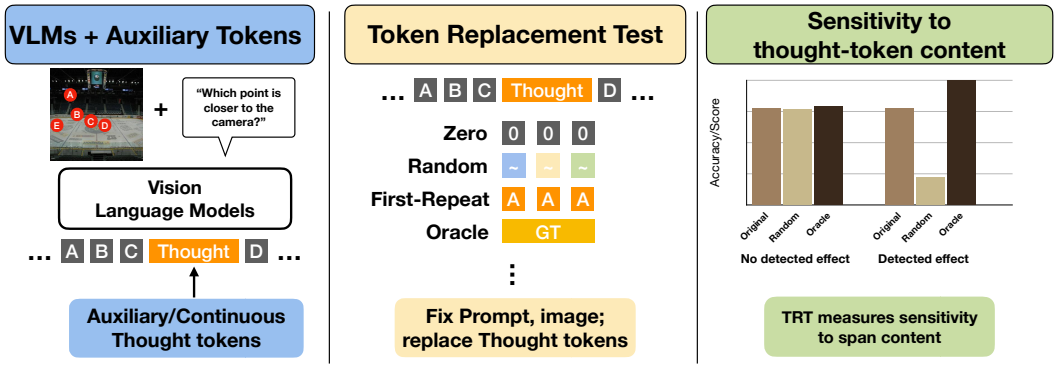


Figure 1: Overview of the Token Replacement Test (TRT). TRT replaces the intermediate thought-token span while fixing the prompt, image, token budget, and decoding procedure. Comparing identity, random, first-repeat, and oracle replacements separates sensitivity to span content from span-position and token-budget effects.

1 INTRODUCTION

Vision-language models (VLMs) first rose to prominence on tasks where success largely reduces to conditioning on visual features and producing short-form text: captioning, visual question answer-

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.

ing, and referring expression comprehension (Vinyals et al., 2015; Antol et al., 2015; Kazemzadeh et al., 2014). In these regimes, the dominant bottlenecks are perceptual coverage and alignment. For such tasks, scaling data, architectures, and instruction tuning tends to yield steady improvements.

Recent benchmarks increasingly target settings where perception is necessary but not sufficient, and *multi-step visual reasoning* becomes the limiting factor (Yue et al., 2024; Lu et al., 2024; Yu et al., 2024). In parallel, explicitly eliciting intermediate structure, most notably via *chain-of-thought*, can substantially improve language models on hard problems (Wei et al., 2022; Kojima et al., 2022). Together, these trends raise a basic question for multimodal systems: what should intermediate structure look like when plain text is an awkward or lossy interface for visual computation?

A natural answer is to augment VLMs with the ability to reason beyond language, using intermediate visual representations. Ideally, they would emit and later consume such intermediate representations as part of their reasoning process. One family of such representations uses *discrete* visual tokens; for instance, "perception tokens" encode intermediate artifacts like depth maps or bounding boxes, and condition reasoning on those tokens (Bigverdi et al., 2025). A second family uses latent non-textual tokens, often framed as *latent* visual reasoning tokens, that the model produces and later reads (Tong et al., 2025; Ray et al., 2026; Qin et al., 2025; Yang et al., 2026b). Across both families, these visual token mechanisms routinely report accuracy gains. However, accuracy gains can reflect incidental effects rather than information in the tokens: a fixed insertion position can become a reliable cue, a larger effective context/capacity from extra tokens can provide more computational strength, or auxiliary objectives for learning visual tokens might regularize training.

This paper asks a more pointed version of the question: *when visual reasoning tokens help, are gains driven by the information in the tokens, or by incidental position and budget effects?* We propose **Ablate-to-Validate**, an empirical principle for answering this. If a model truly relies on visual reasoning token information, then systematically perturbing token values, while holding the prompt, token budget, image features, and decoding procedure fixed, should induce predictable performance changes. We operationalize this principle with the **Token Replacement Test (TRT)**, our standardized inference-time replacement protocol that isolates (1) *span-position* bias via zero and random replacement, (2) *token-budget versus token-diversity* effects via first-repeat and count-matched variants when token counts differ, and (3) sensitivity to content via oracle injection, slot shuffling and distribution-matched noise (Figure 1).

We first apply TRT to off-the-shelf VLMs with visual reasoning tokens—Mirage (Yang et al., 2026b), Mull-Tokens (Ray et al., 2026), and Chain-of-Visual-Thought (CoVT) (Qin et al., 2025)—on their own benchmarks. To make these interventions precise, we then develop a controlled testbed based on questions that require depth reasoning. Concretely, we insert an explicit depth-token span into the language stream and train models to both predict and consume continuous depth embeddings at those positions. We implement this pipeline for multiple VLM backbones (LLaVA (Liu et al., 2023b) and Qwen2.5-VL (Bai et al., 2025)), enforcing a fixed visual token budget per configuration. We also compare against a discrete depth baseline (Bigverdi et al., 2025) that quantizes depth into a VQ-VAE codebook and trains via cross-entropy with the same span placement.

Empirically, TRT reveals a recurring pattern: for continuous visual tokens, accuracy is largely insensitive to the span's content. Across all methods with continuous visual tokens, replacing the span with distribution-matched noise or shuffling it across slots barely moves accuracy, and injecting an oracle yields no measurable gain; replacing the span with zeros or unrelated random vectors, which the model never saw in training, can still lower accuracy or stop the model from answering. We hypothesize that models can profit from the presence and placement of a visual-token span—its budget, its anchoring, and its training-time effects—without reliably decoding the span's semantic content as a bottleneck for the final decision. In our testbed, accuracy with discrete depth tokens is more sensitive to their content (Table 7). TRT provides a standardized validation step that shows how much and where the content of visual reasoning tokens matters for accuracy, and we recommend reporting TRT-style diagnostics as standard practice when introducing new latent-token mechanisms.

2 Related work

How VLMs process visual content. Early VLMs such as Flamingo (Alayrac et al., 2022) showed that inserting cross-modal attention into a (largely) frozen LLM can yield strong few-shot transfer,

while later open models such as LLaVA (Liu et al., 2023b) and Qwen2.5-VL (Bai et al., 2025) popularized simpler projection-based designs that map visual features into the LLM embedding space.

As evaluation has shifted toward multi-step reasoning regimes, research has explored whether explicitly eliciting intermediate structure improves multimodal performance. Multiple efforts have tried to include intermediate rationales or structured traces to support visual–linguistic inference (Zhang et al., 2024; Lu et al., 2022). However, serializing geometric or perceptual computation into text can be lossy, which motivates approaches that augment VLMs with visual token streams intended to serve as intermediate reasoning substrates.

Spatial and depth reasoning benchmarks as testbeds. A large body of work diagnoses where VLMs fail on spatial reasoning: basic predicates such as left/right and above/below under minimal-pair designs (SpatialSense (Yang et al., 2019), VSR (Liu et al., 2023a), What'sUp (Kamath et al., 2023)), 3D and viewpoint sensitivity (3DSRBench (Ma et al., 2025), ViewSpatial-Bench (Li et al., 2025)), and consistent spatial maps across images, video and limited views (VSI-Bench (Yang et al., 2025), MMSI-Bench (Yang et al., 2026a), MindCube (Wang et al., 2026)); shortcuts remain pervasive and motivate debiasing through iterative filtering (Brown et al., 2025).

These benchmarks locate failure modes but do not test how sensitive task accuracy is to the content of an intermediate mechanism such as a visual token stream. TRT is such a test, which we instantiate in a depth reasoning testbed where token replacements and oracle injection are well-defined.

Discrete visual tokens for reasoning. One line of work introduces explicit intermediate artifacts as tokens and conditions subsequent reasoning on them. Perception tokens (Bigverdi et al., 2025) exemplify this approach by encoding intermediate representations such as depth maps or bounding boxes into discrete tokens that the model can generate and then consume during inference.

Latent visual reasoning tokens. Motivated by the hypothesis that continuous representations better preserve perceptual structure than discretized alternatives, several recent methods augment VLMs with latent token streams that lie outside the standard vocabulary. MetaMorph (Tong et al., 2025) introduces interleaved continuous visual tokens in the generation stream and trains models for multimodal understanding and generation. Chain-of-Visual-Thought (CoVT) (Qin et al., 2025) distills information from multiple vision experts into a compact continuous span that the VLM predicts before generating answers. Mirage (Yang et al., 2026b) frames latent tokens as "machine mental imagery," recasting hidden states into a latent visual channel through a two-stage training paradigm. Mull-Tokens (Ray et al., 2026) generalizes this idea to modality-agnostic latent thinking tokens, and Latent Implicit Visual Reasoning (Li et al., 2026) trains models to discover visual reasoning tokens without explicit supervision.

Across these works, visual reasoning token mechanisms often report improvements on a variety of perception and reasoning benchmarks. However, accuracy gains alone do not establish that a model relies on *visual reasoning token content*. Improvements can arise from confounds such as fixed span placement (a reliable marker), changes in effective capacity via token budget, or visual objectives that regularize training. Our work directly targets this gap: we propose the Token Replacement Test (TRT), a standardized content-replacement diagnostic that holds span placement and token budget fixed, varies only the span's content, and reports the measured sensitivity of task accuracy to each replacement.

3 METHOD

3.1 PRELIMINARIES: VISUAL TOKENS IN VLMS

Autoregressive VLMs generate an output sequence conditioned on an image I and a textual prompt x. We write the textual output as $y_{1:T}$ with $y_t \in \mathcal{V}$, and the next-token distribution

$$p(y_t \mid y_{<t}, x, I) = \mathrm{Softmax}(W h_t), \tag{1}$$

where $h_t \in \mathbb{R}^H$ is the hidden state and W is the output projection.

A growing class of methods augments this process with a *visual token span* intended to act as an intermediate representation. We denote the visual span by $u_{1:K}$, where K is the visual token budget.

Depending on the method, the visual tokens may be

$$u_i \in \begin{cases} \mathcal{V}_u & \text{(discrete visual tokens)} \\ \mathbb{R}^D & \text{(continuous/latent visual tokens).} \end{cases} \tag{2}$$

The model produces (or is given) a visual span at designated positions and then generates text conditioned on it, so it may learn to *predict* visual tokens (e.g., via a visual loss) without *using their content* for the final answer. Since most prior work reports only end-task accuracy, it remains unclear whether visual tokens are acting as an information-bearing bottleneck or merely as a scaffold (e.g., a fixed marker, extra budget, or regularization).

Our goal is to test *sensitivity to span content*: does changing the *values* of $u_{1:K}$ (while keeping everything else fixed) change downstream performance in the way we would expect if the model were relying on visual content?

3.2 Ablate-to-Validate Principle

We propose **Ablate-to-Validate**: if a model genuinely relies on the information encoded in a visual span, then systematically perturbing that span at inference should induce predictable changes in the final output quality. We instantiate this principle as the **Token Replacement Test (TRT)**, a standardized suite of *content-replacement interventions* applied to the visual span $u_{1:K}$ while holding fixed the prompt x, image features from I, the token budget K, and the decoding procedure.

3.3 Token Replacement Test (TRT)

All TRT replacements are applied at inference, after training is complete, on each method's test data (Appendix B.2); no model is retrained. We treat the visual span as an explicit object in the forward pass. Let $\mathrm{Inject}(x, u_{1:K})$ denote the augmented input sequence (or internal stream) in which the visual span is consumed. TRT evaluates a model by running the same example under a collection of replacements $u_{1:K} \leftarrow \tilde{u}_{1:K}$ and measuring the change in task performance.

Replacement interventions. TRT uses the following replacements, designed to separate distinct confounds:

- **Identity re-injection.** $\tilde{u}_{1:K} \leftarrow u_{1:K}$, verifying that the interception and replacement pathway is correct.
- **Zero replacement.** $\tilde{u}_{1:K} \leftarrow 0$ (or a designated null token for discrete spans), testing sensitivity to removing visual content.
- **Random replacement.** $\tilde{u}_{1:K} \leftarrow \epsilon$, with ϵ sampled i.i.d. (e.g., Gaussian for continuous spans), a draw that matches nothing about the span.
- **Distribution-matched noise.** $\tilde{u}_{1:K} \leftarrow \epsilon$, a Gaussian with the same mean and standard deviation as the span it replaces or its oracle (per method: Appendix D.3).
- **First-repeat.** $\tilde{u}_i \leftarrow u_1$ for all i, preserving budget and span placement while removing token diversity.
- **Count-matched variants.** When comparing methods whose native visual spans differ in length, we apply the same replacement while matching the number of intervened tokens to the number used by the corresponding identity run for that example.[1] This separates "more tokens" from "better tokens."
- **Oracle injection.** When an oracle exists (the visual tokens the model was trained to produce), $\tilde{u}_{1:K} \leftarrow u^{\star}_{1:K}$ diagnoses whether the model *can* exploit a high-quality visual signal.
- **Slot shuffle.** $\tilde{u}_s \leftarrow u^{\star}_{\pi(s)}$, where π is one fixed random reordering of the K slots that moves every slot (a derangement), shared by all examples; it measures how sensitive accuracy is to the oracle's arrangement and is not a full distribution match.

[1] Some methods, e.g., Mirage (Yang et al., 2026b), do not fix the number of visual reasoning tokens during generation; the count-matched variant then matches the number of ablated tokens to the number generated in the identity condition for the same example and forces the end marker once that count is reached.

Interpreting TRT. TRT is a sensitivity test: each replacement reports the change in test accuracy under a stated intervention. Distribution-matched noise and the slot shuffle change the span's content while holding its scale and statistics fixed, so they measure how much accuracy depends on that content; where an oracle exists, oracle injection measures how much accuracy it can gain. Zero and i.i.d. random replacements also push the span off its training distribution, so a drop under them alone is not evidence of content use. A decoding collapse (the model gives no answer after the span) is scored as wrong. First-repeat and count-matched variants further separate "token diversity" effects from "token budget" effects.

3.4 A CONTROLLED TESTBED: DEPTH SPANS FOR COUNTERFACTUAL INTERVENTION

To apply TRT cleanly, we require a setting where (i) visual content is well-defined, (ii) counterfactual replacements are meaningful, and (iii) oracle visual signals are available. We choose **depth reasoning** as a controlled testbed because depth provides a continuous, geometrically grounded representation that supports clean interventions (zeroing or randomizing embeddings) and admits oracle depth injection. We train on relative-depth questions over 19,279 point-marked ADE20K images (Zhou et al., 2019) and test on HardBLINK (Bigverdi et al., 2025): 372 relative-depth questions with 3, 4 or 5 marked points on BLINK (Fu et al., 2024) images (Appendix C).

Depth span markup and fixed budget. We insert an explicit depth span into the response using boundary tokens `<DEPTH_START>` and `<DEPTH_END>`. At the text level, we include a single placeholder token `<DEPTH_TOKEN>` to keep responses readable; internally, the model expands this placeholder into a fixed-length span of K visual tokens. This enforces a strictly matched visual budget across training, inference, and all TRT replacements in our controlled depth setting. For discrete depth spans, we use a uniform budget of $K = 100$ tokens across methods, following Aurora (Bigverdi et al., 2025); for continuous depth spans, K is held fixed within each method and across all TRT ablations for that method.

Depth targets. For each example, we obtain a depth map from Depth Anything ViT-S (Yang et al., 2024), since neither ADE20K (Zhou et al., 2019) nor BLINK provides depth annotations (Appendix C.1), and convert it into a token-aligned target representation under two instantiations:

- **Continuous depth tokens:** real-valued embeddings in a frozen feature space (Yang et al., 2026b; Tong et al., 2025; Ray et al., 2026).
- **Discrete depth tokens:** quantized codebook indices trained with cross-entropy (Bigverdi et al., 2025).

The two share span placement and supervision locations, but their budgets are not literally equivalent: discrete variants expand the output vocabulary, whereas continuous variants inject real-valued vectors and may differ in feature dimension.

3.5 CONTINUOUS DEPTH SPANS

In the continuous variant, depth is represented as $z \in \mathbb{R}^{K \times D}$, a sequence of K depth embeddings extracted in a frozen encoder space of dimension D. We support multiple frozen encoders (e.g., SigLIP2 (Tschannen et al., 2025), DINOv2 (Oquab et al., 2024), and CLIP (Radford et al., 2021)) to test how the target feature geometry affects utilization. When an encoder yields a different patch count than K, we interpolate patch embeddings (using 2D interpolation when preserving a grid is appropriate) to obtain exactly K targets.

Injection. At training time, we learn a **depth projector** $P : \mathbb{R}^D \to \mathbb{R}^H$ that maps each depth embedding into the language hidden size. During the forward pass, the embeddings at the depth span positions are replaced by $P(z)$, the projector applied to each of the K depth embeddings.

Prediction. To train the model to produce depth content at the depth span, we learn a **depth head** $H_d : \mathbb{R}^H \to \mathbb{R}^D$ that maps language hidden states back into the frozen depth feature space. Let $\hat{z} = H_d(h)$ denote the predicted depth embeddings at the depth span positions. We supervise depth with a regression-style objective (e.g., MSE or cosine loss) applied only on the depth span.

Table 1: **Mirage on VSP Spatial Planning**, accuracy (%) on 400 questions. Oracle (helper image): the embedding of the maze image with its solution path drawn, the latent target of Mirage's first training stage.

Replacement	Accuracy
Identity (self)	**76.25**
First-repeat	75.75
Oracle (helper image)	75.00
Oracle (helper image, count-matched)	**76.25**
Random (i.i.d.)	74.75
Random (helper-image dist)	74.75
Random (model dist)	**76.25**
Zero	51.50

Training objective. We jointly optimize $\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}}$, where $\mathcal{L}_{\text{depth}}$ is computed only at depth span positions and λ_{depth} balances the two objectives.

3.6 Discrete depth spans

In the discrete baseline, depth is quantized into a codebook and represented as token IDs $d_{1:K} \in \{1, \ldots, |\mathcal{C}|\}$. Training reduces to cross-entropy over discrete depth tokens, and inference can optionally enforce constrained decoding inside the depth span to ensure only valid depth tokens appear.

3.7 TRT on off-the-shelf VLMs

While depth spans provide a controlled environment for matched token budgets and oracle injection, TRT is intended as a general diagnostic. To demonstrate breadth, we apply TRT to additional "visual reasoning" models that expose visual reasoning token streams. For each model, we intercept the visual tokens at their designated positions and replace them with the TRT interventions immediately before they are consumed for downstream decoding: the latent vectors of Mirage (Yang et al., 2026b), the latent token segment of Mull-Tokens (Ray et al., 2026), and the input embeddings of the CoVT (Qin et al., 2025) span. Because Mirage does not enforce a fixed visual reasoning token budget during generation, we also run its oracle setting under the count-matched protocol (Table 1). No architectural change is required.

4 Experiments

Our experiments ask a targeted question: *when visual reasoning tokens improve accuracy, how much does their content matter?* We answer this by (i) applying the **Token Replacement Test (TRT)** to three published off-the-shelf visual reasoning methods on their own benchmarks, and (ii) analyzing the outcome in a controlled depth-span testbed with aligned span placement and controlled budget protocols for **continuous** and **discrete** visual reasoning tokens.

4.1 Main result: do off-the-shelf methods use their visual reasoning tokens?

We begin with the setting the question is actually about: real, published visual reasoning methods. We apply TRT to Mirage, Mull-Tokens and CoVT as described in Section 3.7 (Tables 1–3).

Mirage (Yang et al., 2026b) interleaves latent visual reasoning tokens with text during decoding by reusing hidden states as compact "visual reasoning tokens." We reproduce Mirage's VSP (Wu et al., 2024) Spatial Planning (SP) evaluation. As shown in Table 1, zeroing latents lowers SP accuracy (76.25% $\rightarrow$ 51.50%). Reproduction details are provided in Appendix F.1.

Mull-Tokens (Ray et al., 2026) adds latent tokens for intermediate states. At inference the Mull span is 20 copies of a single static `<|latent_pad|>` embedding inserted in the prompt, whose norm is about 1.4% of a typical token embedding's, so zero and random replacement leave the input nearly unchanged. Both nonetheless change the generated text on a few questions (15 to 21 of the 700 BLINK and 4 to 12 of the 300 SAT questions, depending on the replacement and model), while accuracy moves by at most 1 pt, and first-repeat is the identity by construction (Table 2). Because this input barely changes, we also test one level down: during generation, we overwrite the attention values that the span's positions hold in the model's cache, in every layer, with distribution-matched noise. This also leaves accuracy essentially unchanged (Appendix A.4). Mull's gain therefore does not run through the content of its latent span.

Table 2: **Mull-Tokens on BLINK and SAT** (Ray et al., 2025), accuracy (%). †Identical to identity by construction: the span is 20 copies of one pad embedding. ‡A second draw from the same distribution as Random.

Replacement	Mull-Tokens		Mull-Tokens-GRPO	
	BLINK	SAT	BLINK	SAT
Identity (self)	**63.43**	**76.33**	64.71	**77.67**
First-repeat†	63.43	76.33	64.71	77.67
Random	63.00	76.00	**65.29**	77.00
Random (second draw)‡	63.00	76.00	64.43	76.67
Zero	**63.43**	**76.33**	64.57	77.33

Table 3: **CoVT-7B-depth on CV-Bench** (Tong et al., 2024), overall accuracy (%). †Identical to identity by construction: the anchor pads share one input embedding.

Replacement	2D	3D
Identity (self)	**76.26**	**80.83**
First-repeat†	76.26	80.83
Random (i.i.d.)	21.39	19.00
Random (same dist)	74.65	79.17
Zero	58.97	69.00

Table 4: **Depth reasoning on HardBLINK**, accuracy (%). Continuous (best): best of Table 5.

Backbone	No-aux	Discrete ($K = 100$)	Continuous (best)
LLaVA-13B	75.54	**77.96**	74.46
Qwen2.5-VL-3B	66.40	**71.24**	68.28

Table 5: **Continuous depth spans by encoder and span length** K, accuracy (%). Full: K=256 (SigLIP2, DINOv2) or 576 (CLIP).

Backbone	Encoder	$K = 4$	$K = 16$	$K = 64$	full
LLaVA-13B	SigLIP2	73.39	72.58	**74.46**	71.77
	CLIP	72.31	73.12	73.39	65.86
	DINOv2	66.94	63.71	61.56	58.87
Qwen2.5-VL-3B	SigLIP2	54.57	61.02	55.11	63.98
	CLIP	64.25	**68.28**	55.38	67.20
	DINOv2	65.32	66.94	67.47	64.25

CoVT (Qin et al., 2025) distills continuous visual reasoning tokens from vision experts. Identity matches baseline, while i.i.d. random noise collapses performance; distribution-matched random largely recovers it (Table 3), so at the input level CoVT tolerates in-distribution replacements but not off-distribution ones. As for Mull-Tokens, we repeat the test one level down on the cached attention values: distribution-matched and first-repeat swaps again leave accuracy unchanged, while off-distribution swaps lower it (Appendix A.4).

Content of visual tokens matters less than their presence in many settings.

4.2 Our controlled depth testbed: do visual tokens help?

Off-the-shelf methods differ in data, objectives, and token budgets, so these experiments show *how much* accuracy depends on the content of their visual reasoning tokens, but not *why*. We therefore use a testbed we control end to end (Section 3.4), where every TRT replacement, including oracle injection, is exactly defined.

Benchmark and variants. We test on HardBLINK (Section 3.4; 124 questions in each of the 3/4/5-point subsets). The depth-span models output a brief rationale and then a single final label, which we parse deterministically; unparseable answers count as incorrect. We vary backbone (**LLaVA-13B**, **Qwen2.5-VL-3B**), frozen depth feature space (**SigLIP2**, **CLIP**, **DINOv2**, **VQ-VAE**[2]), and the continuous token budget $K \in \{4, 16, 64, \text{full}\}$; the discrete baseline uses $K = 100$.

Training and implementation details. We train depth-span models on ADE20K (19,279 images) for a fixed horizon. The depth encoder is frozen; we train the depth projector P, the depth head H_d and a small set of language parameters (LoRA for LLaVA; Appendix B.1 for Qwen2.5-VL), with $\mathcal{L}_{\text{depth}}$ an MSE or cosine loss on continuous spans and token-level cross-entropy over codebook indices on discrete spans. Hyperparameters and compute: Appendices B.1 and G.

We compare **No-aux**, the plain VQA baseline trained to output only the final answer, with the **Discrete** ($K = 100$) and **Continuous** variants (Table 5). Table 4 therefore compares representation families under their own budget protocols, not the same K in every row.

Gains depend on backbone and representation. As summarized in Table 4, visual reasoning tokens can outperform No-aux, but the gains depend on both backbone and representation. For LLaVA-13B, the discrete model improves modestly over No-aux (77.96 vs. 75.54; not significant on

[2]See Appendix A.1.

Table 6: **TRT on the continuous depth spans**, accuracy (%).

Replacement	3-pt	4-pt	5-pt	Avg
LLaVA-13B (SigLIP2, $K = 64$)				
Identity (self)	78.23	73.39	71.77	74.46
Oracle (GT)	79.03	73.39	71.77	**74.73**
Slot shuffle (oracle)	78.23	73.39	71.77	74.46
Random (i.i.d.)	79.03	72.58	70.97	74.19
Zero[§]	0.00	0.00	0.00	0.00
First-repeat	78.23	73.39	71.77	74.46
Qwen2.5-VL-3B (DINOv2, $K = 64$)				
Identity (self)	69.35	68.55	64.52	67.47
Oracle (GT)	68.55	70.97	66.94	**68.82**
Slot shuffle (oracle)	69.35	70.16	65.32	68.28
Random (i.i.d.)	67.74	70.97	62.10	66.94
Zero	69.35	70.16	64.52	68.01
First-repeat[§]	5.65	1.61	5.65	4.30

[§]Decoding collapse: no answer after the span on 372/372 (LLaVA-13B) and 343/372 (Qwen2.5-VL-3B) questions, scored as wrong.

Table 7: **TRT on the discrete depth tokens**, accuracy (%).

Replacement	3-pt	4-pt	5-pt	Avg
LLaVA-13B (discrete)				
Identity (self)	83.87	74.19	75.81	77.96
Oracle (GT)	83.06	82.26	79.03	**81.45**
Slot shuffle (oracle)	77.42	70.16	70.97	72.85
Random (i.i.d.)	73.39	66.94	66.13	68.82
Constant / zero	78.23	71.77	70.97	73.66
Qwen2.5-VL-3B (discrete)				
Identity (self)	72.58	75.00	66.13	71.24
Oracle (GT)	83.06	84.68	74.19	**80.65**
Slot shuffle (oracle)	54.84	61.29	45.16	53.76
Random (i.i.d.)	55.65	47.58	33.06	45.43
Constant / zero	66.94	61.29	49.19	59.14

the 372 items, exact McNemar p=0.36, not prompt-matched), whereas the best continuous configuration reaches 74.46. For Qwen2.5-VL-3B, the strongest continuous configuration reaches 68.28 (CLIP, K=16; Table 5), and the discrete result reported in Table 4 is 71.24 (+4.84 over No-aux; not significant on the 372 items, exact McNemar p=0.105, 46 vs. 64 discordant).

Visual tokens can help—but continuous is not consistently better.
Visual backbone choice matters—but longer visual spans are not consistently better.

The ordering of frozen feature spaces is backbone-specific: SigLIP2 and CLIP lead DINOv2 at every budget on LLaVA-13B, while on Qwen2.5-VL-3B DINOv2 leads SigLIP2 at every budget and splits with CLIP. No row's accuracy rises monotonically with span length. Each configuration is a single training run, so we report the observed ordering rather than a ranking.

4.3 TRT ON THE CONTROLLED TESTBED: CONTINUOUS LESS SENSITIVE, DISCRETE MORE SENSITIVE

Main accuracy gains do not imply reliance on span *content*. We apply TRT to the continuous configurations (LLaVA: SigLIP2, $K = 64$; Qwen: DINOv2, $K = 64$; Appendix D.2) by replacing depth-span embeddings at inference while holding fixed prompt, image features, span placement, and decoding. We also run the analogous TRT on the discrete models by replacing token IDs (with constrained decoding for validity), with the span kept at the model's own length to rule out span-length effects. Because random and constant tokens also move the span off its training distribution, we add a slot shuffle that keeps the oracle's own tokens but changes their order. The corresponding continuous and discrete results are reported in Tables 6 and 7, respectively.

Continuous spans are robust to content corruption; discrete spans are not.

For the two K=64 continuous spans, accuracy changes little under i.i.d. random replacement and oracle injection. For LLaVA-13B, oracle injection improves average accuracy only marginally over self-predicted tokens (74.46 → 74.73). For Qwen2.5-VL-3B, replacing the model's predicted depth span with oracle embeddings slightly raises accuracy (67.47 → 68.82). Neither gain is detectable (Table 13). By contrast, the discrete setting shows substantially stronger sensitivity to token content. For Qwen2.5-VL-3B, oracle replacement produces a large improvement (71.24 → 80.65, +9.41), while random replacement causes a severe drop (71.24 → 45.43, −25.81), and constant/zero tokens also degrade performance significantly (to 59.14). Shuffling the oracle tokens over the depth grid lowers accuracy from 80.65 to 53.76, below identity, so the shuffle costs more than oracle injection gained. These effects are much larger than those observed in the continuous setting, where analogous content perturbations that still produce answers typically change accuracy by only 1–2 points. Taken together, these results indicate that discrete depth tokens carry content that accuracy depends on, whereas continuous spans are comparatively insensitive to the correctness of their content and are more consistent with reliance on span presence, placement, or coarse distributional structure. With the same depth tokens and span length for both encodings, the continuous span stays insensitive (Table 9); swapping the span's attention values for distribution-matched noise lowers accuracy in the Qwen2.5-VL-3B discrete model but not in the LLaVA-13B continuous model (Table 12). We report these as differences in measured sensitivity between implemented systems.

4.4 THREE TRT SIGNATURES

Across the three off-the-shelf methods and our two continuous $K{=}64$ models, accuracy changes little under distribution-matched noise or the slot shuffle, and oracle injection adds no measurable accuracy. Three TRT signatures indicate that a visual reasoning token gain is not driven by content. Each can produce accuracy gains without the model reading the tokens' content:

(1) Corruption tolerated — distribution-matched or shuffled content preserves the gain. Even such corrupted content does not punish the model.
(2) Diversity unused — first-repeat often suffices. A single vector tiled K times keeps performance up.
(3) Content unused — oracle injection adds no measurable headroom. Even a strong signal is not being read.

Across off-the-shelf methods and our depth testbed, we see strong evidence for corruption tolerated in many configurations, mixed evidence for diversity unused, and weaker evidence for content unused, motivating intervention-based validation alongside accuracy improvements. In both discrete models, accuracy is more sensitive to the slot shuffle and to oracle injection.

5 DISCUSSION

Our results suggest a simple but important takeaway: an intermediate token channel can *exist* and even correlate with higher accuracy without becoming a strict causal bottleneck for the final prediction. This mirrors a nuanced pattern in the chain-of-thought (CoT) literature: models can preserve much of the benefit of CoT when demonstration steps are invalid, yet irrelevant or inaccurate rationale noise can also cause substantial degradation (Wang et al., 2023; Zhou et al., 2024). In other words, intermediate "reasoning traces" can function as training or prompting scaffolds without being faithfully *consumed* as information at inference time.

Our discrete and continuous models differ in measured sensitivity; the comparison does not isolate the encoding as the cause. In the discrete setting, we observe substantially stronger sensitivity to token content: on Qwen2.5-VL, replacing predicted tokens with oracle depth tokens yields large gains and shuffling the oracle tokens causes severe degradation, while on both backbones random tokens lower accuracy, as does a distribution-matched swap of the span's attention values on Qwen2.5-VL. This indicates that discrete visual reasoning tokens can carry content that accuracy depends on. In contrast, the continuous setting exhibits much weaker sensitivity to content, suggesting that continuous latent spans are often used through coarse or structural signals rather than as fine-grained intermediate representations.

For continuous visual reasoning tokens, low sensitivity may arise from training and architectural effects. Modern latent visual reasoning methods use multi-stage pipelines and auxiliary objectives that can improve representations or calibration even if the model does not ultimately use the span as an information-bearing channel at test time. When the visual reasoning span is always present at a consistent location, the model may also treat it as a structural cue without decoding its fine-grained content.

A complementary explanation is an information-bottleneck mismatch: continuous spaces are high-dimensional and redundant relative to a depth decision that often depends on a few coarse comparisons, so the model may exploit marginal statistics or positional regularities of the span instead of the intended semantics, producing the TRT signature we observe: limited gains from oracle injection and little change when the span's content is shuffled or replaced by distribution-matched noise.

Visual reasoning token evaluation therefore requires more than end-task accuracy: a method can report a gain while accuracy is insensitive to the replacements tested, and the way to tell is to perturb content with placement, budget, prompt, image and decoding fixed.

Limitations. The controlled testbed covers one task, relative depth; in future work, we will try TRT on more tasks and applications.

AI USE STATEMENT

We used generative AI tools for result aggregation and visualization and to edit parts of this paper for readability. We did not use them to generate datasets or to develop models or conceptual frameworks. All AI-assisted work was reviewed by the authors, who chose the research questions, made every design decision, reviewed and approved any AI-edited wording, verified every reported number against the run artifacts, and take responsibility for the final content.

ETHICS STATEMENT

This work is a diagnostic study of intermediate visual tokens in vision-language models. It involves no human subjects and no personal data: all evaluations use publicly available benchmarks (BLINK, the BLINK-derived HardBLINK set, SAT, VSP, and CV-Bench) and the ADE20K-derived depth training data described in Appendix C, and the off-the-shelf methods are evaluated with their publicly released checkpoints on their own benchmarks (Appendix F). We release no model with capabilities beyond those of the underlying methods; the intended use of the Token Replacement Test is to make reported gains from latent visual tokens verifiable.

REPRODUCIBILITY STATEMENT

Section 3.3 and Appendices D.3, E.5 and A.4 give the exact mechanics of every replacement of the Token Replacement Test; Appendix C the data construction, prompting, and answer-extraction protocol; Appendices B and B.1 the training recipes and key hyperparameters of the controlled models; Appendix F the released checkpoints and evaluation wrappers used for the off-the-shelf methods, with a verification of their published numbers in Appendix A.3; and Appendix G the compute used. Code and the depth-augmented training data will be released upon publication.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23716–23736, 2022. doi: 10.52202/068431-1723.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2425–2433, 2015. doi: 10.1109/ICCV.2015.279.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025. doi: 10.48550/arXiv.2502.13923. URL https://arxiv.org/abs/2502.13923.

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

Ellis Brown, Jihan Yang, Shusheng Yang, Rob Fergus, and Saining Xie. Benchmark designers should "train on the test set" to expose exploitable non-visual shortcuts. *arXiv preprint arXiv:2511.04655*, 2025.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, Dahua Lin, and Kai Chen. VLMEvalKit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 11198–11201, 2024. doi: 10.1145/3664647.3685520.

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 *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 15081 of *Lecture Notes in Computer Science*, pp. 148–166. Springer, 2024. doi: 10.1007/978-3-031-73337-6_9.

Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's "up" with vision-language models? Investigating their struggle with spatial reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9161–9175, 2023. doi: 10.18653/v1/2023.emnlp-main.568.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 787–798, 2014. doi: 10.3115/v1/D14-1086.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.48550/arXiv.2205.11916. URL https://arxiv.org/abs/2205.11916.

Dingming Li, Hongxing Li, Zixuan Wang, Yuchen Yan, Hang Zhang, Siqi Chen, Guiyang Hou, Shengpei Jiang, Wenqi Zhang, Yongliang Shen, Weiming Lu, and Yueting Zhuang. ViewSpatial-Bench: Evaluating multi-perspective spatial localization in vision-language models. *arXiv preprint arXiv:2505.21500*, 2025. URL https://arxiv.org/abs/2505.21500.

Kelvin Li, Chuyi Shang, Leonid Karlinsky, Rogerio Feris, Trevor Darrell, and Roei Herzig. Latent implicit visual reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 33457–33466, 2026.

Fangyu Liu, Guy Emerson, and Nigel Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651, 2023a. doi: 10.1162/tacl_a_00566.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, 2023b. doi: 10.48550/arXiv.2304.08485. URL https://arxiv.org/abs/2304.08485.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems*, volume 35, pp. 2507–2521, 2022. doi: 10.52202/068431-0182.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*, 2024. doi: 10.48550/arXiv.2310.02255. URL https://openreview.net/forum?id=KUNzEQMWU7.

Wufei Ma, Haoyu Chen, Guofeng Zhang, Yu-Cheng Chou, Jieneng Chen, Celso M. de Melo, and Alan Yuille. 3DSRBench: A comprehensive 3D spatial reasoning benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6924–6934, 2025.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

Robert G. Newcombe. Improved confidence intervals for the difference between binomial proportions based on paired data. *Statistics in Medicine*, 17(22):2635–2650, 1998.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 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.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Arijit Ray, Jiafei Duan, Ellis Brown, Reuben Tan, Dina Bashkirova, Rose Hendrix, Kiana Ehsani, Aniruddha Kembhavi, Bryan A. Plummer, Ranjay Krishna, Kuo-Hao Zeng, and Kate Saenko. SAT: Dynamic spatial aptitude training for multimodal language models. In *Conference on Language Modeling (COLM)*, 2025.

Arijit Ray, Ahmed Abdelkader, Chengzhi Mao, Bryan A Plummer, Kate Saenko, Ranjay Krishna, Leonidas Guibas, and Wen-Sheng Chu. Mull-Tokens: Modality-agnostic latent thinking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 9477–9488, 2026.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Ziteng Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 87310–87356, 2024. doi: 10.52202/079017-2771.

Shengbang Tong, David Fan, Jiachen Zhu, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. MetaMorph: Multimodal understanding and generation via instruction tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17001–17012, 2025. doi: 10.1109/ICCV51701.2025.01579.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Henaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3156–3164, 2015. doi: 10.1109/CVPR.2015.7298935.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. Towards understanding chain-of-thought prompting: An empirical study of what matters. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2717–2739, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.153. URL https://aclanthology.org/2023.acl-long.153/.

Qineng Wang, Baiqiao Yin, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, Saining Xie, Jiajun Wu, Li Fei-Fei, and Manling Li. Spatial mental modeling from limited views. In *International Conference on Learning Representations (ICLR)*, pp. 82559–82622, 2026.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.48550/arXiv.2201.11903. URL https://arxiv.org/abs/2201.11903.

Qiucheng Wu, Handong Zhao, Michael Saxon, Trung Bui, William Yang Wang, Yang Zhang, and Shiyu Chang. VSP: Assessing the dual challenges of perception and reasoning in spatial planning tasks for VLMs. *arXiv preprint arXiv:2407.01863*, 2024.

Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10632–10643, 2025. doi: 10.1109/CVPR52734.2025.00994.

Kaiyu Yang, Olga Russakovsky, and Jia Deng. SpatialSense: An adversarially crowdsourced benchmark for spatial relation recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10371–10381, 2024. doi: 10.1109/CVPR52733.2024.00987.

Sihan Yang, Runsen Xu, Yiman Xie, Sizhe Yang, Mo Li, Jingli Lin, Chenming Zhu, Xiaochen Chen, Haodong Duan, Xiangyu Yue, Dahua Lin, Tai Wang, and Jiangmiao Pang. MMSI-Bench: A benchmark for multi-image spatial intelligence. In *International Conference on Learning Representations (ICLR)*, pp. 157051–157088, 2026a.

Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 33510–33520, 2026b.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. MM-Vet: Evaluating large multimodal models for integrated capabilities. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57730–57754. PMLR, 2024. URL https://proceedings.mlr.press/v235/yu24o.html.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9556–9567, 2024. doi: 10.1109/CVPR52733.2024.00913. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yue_MMMU_A_Massive_Multi-discipline_Multimodal_Understanding_and_Reasoning_Benchmark_for_CVPR_2024_paper.html.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *Transactions on Machine Learning Research*, 2024.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. doi: 10.1007/s11263-018-1140-0.

Zhanke Zhou, Rong Tao, Jianing Zhu, Yiwen Luo, Zengmao Wang, and Bo Han. Can language models perform robust reasoning in chain-of-thought prompting with noisy rationales? In *Advances in Neural Information Processing Systems*, volume 37, pp. 123846–123910, 2024. doi: 10.52202/079017-3936.

A Additional Results

A.1 HardBLINK additional runs (full blocks)

In Table 8, we present the comprehensive HardBLINK average accuracy results. We evaluate performance across different continuous depth encoders (CLIP, SigLIP2, DINOv2, and VQ-VAE), target sequence lengths (Full, 64, 16, 4), and various ablation settings, including learning rate variations and intermediate layer supervision. The VQ-VAE row is the continuous counterpart of our discrete model: it is trained on the same VQ-VAE depth tokens with the same 100-slot span, backbone, prompt and evaluation set, but reads each token as its codebook embedding rather than as a token ID. Running TRT on both (Table 9), the continuous model stays insensitive to its span's content, while random tokens and the slot shuffle lower the discrete model's accuracy.

Table 8: **HardBLINK average accuracy (%).** Performance across architectures, continuous depth encoders, sequence lengths, and ablation settings (learning rate, depth loss, and intermediate layers).

Method	Configuration	LLaVA 13B	Qwen2.5-VL 3B
Baselines	Base model	33.60	-
	No-aux	75.54	66.40
	Discrete tokens	**77.96**	**71.24**
Continuous (CLIP)	Full	65.86	67.20
	64	73.39	55.38
	16	73.12	**68.28**
	4	72.31	64.25
Continuous (SigLIP2)	Full	71.77	63.98
	64	**74.46**	55.11
	16	72.58	61.02
	4	73.39	54.57
Continuous (DINOv2)	Full	58.87	64.25
	64	61.56	67.47
	16	63.71	66.94
	4	66.94	65.32
Continuous (VQ-VAE)	100	70.70	-
SigLIP2 (64) ablations	No depth loss	72.85	-
	50 ep., LR 2×10^{-4}	24.73	-
	50 ep., LR 1×10^{-5}	60.48	-
	50 ep., LR 5×10^{-5}	66.67	-
SigLIP2 (16) Layers	Layer 9 (5 ep.)	71.51	-
	Layer 19 (5 ep.)	68.01	-
	Layer 29 (5 ep.)	69.09	-
	Layer -1 (5 ep.)	70.70	-
	Layer 19 (10 ep.)	72.04	-

Table 9: **Matched-budget control** (LLaVA-13B, HardBLINK, $N{=}372$): the same VQ-VAE depth tokens and 100-slot span, read as codebook embeddings or as token IDs. Δ: change from the model's own identity run (intervals: Appendix H). [§]Decoding collapse: no answer after the span on 372/372 questions, scored as wrong.

Encoding	Replacement	Acc.	Δ	95% CI
Continuous $E[\text{token}]$, $K{=}100$	Identity (self)	70.70	—	—
	Oracle (GT tokens)	72.31	$+1.61$	$[-1.24, +4.48]$
	Slot shuffle (oracle)	70.43	-0.27	$[-3.00, +2.46]$
	Random (distribution-matched)	70.43	-0.27	$[-3.38, +2.84]$
	First-repeat	72.31	$+1.61$	$[-1.35, +4.59]$
	Zero[§]	0.00	—	—
Discrete token IDs, $K{=}100$	Identity (self)	77.96	—	—
	Oracle (GT tokens)	81.45	$+3.49$	$[+0.17, +6.90]$
	Slot shuffle (oracle)	72.85	-5.11	$[-9.61, -0.63]$
	Random (i.i.d. tokens)	68.82	-9.14	$[-13.37, -4.92]$
	Constant / zero	73.66	-4.30	$[-8.72, +0.11]$

A.2 Scope and method coverage

TRT is a task-agnostic, inference-time diagnostic that measures how sensitive task accuracy is to a latent span's content. We apply TRT across multiple tasks, summarized in Table 10: three off-the-shelf continuous-token methods, two backbone families, five benchmarks spanning spatial, perceptual, and 2D/3D understanding, and the training setups of Table 10. This establishes TRT as a general protocol rather than a task- or model-specific probe. We use depth reasoning as a stress test: base VLMs do poorly on it (Table 11), it is interpretable, has an estimated answer key (the evaluation answer key uses estimated evaluation maps, with training targets produced by the corresponding estimation pipeline; Appendix C.1), and admits clean counterfactuals (zero, random, distribution-matched, first-repeat, oracle) under matched span placement and token budget, so that only the span's content varies. We do not propose a depth-specific model nor claim discrete tokens are categorically better; the discrete-depth setup is an interpretable diagnostic baseline.

Relation to MetaMorph (Tong et al., 2025) and "visual prediction" objectives. Our continuous depth span is conceptually aligned with the broader idea of training an LLM to predict continuous visual features in addition to text. MetaMorph's Visual Predictive Instruction Tuning (VPiT) (Tong et al., 2025) adds a visual head that maps language hidden states into a vision feature space and trains with a combined language + visual objective. We differ in three key respects: (i) our targets are depth-specific embeddings derived from depth maps in a frozen encoder space, (ii) supervision is localized to explicit depth spans under a strictly fixed budget K, enabling controlled counterfactual replacement, and (iii) we explicitly test *sensitivity to span content* using TRT rather than relying on accuracy alone. Since pretrained MetaMorph model weights were not publicly available at the time of this work, we do not include MetaMorph in our empirical evaluation; the authors do provide code, training scripts, and the training-data recipe.

A.3 Baseline reproduction verification

Our reimplementations (Sections 4.1 and 4.2) reproduce each method's published numbers to within 3.4 points, except Aurora-discrete (below). Table 11 verifies our setup against the paper-reported baseline and method numbers. Aurora-discrete is LLaVA-13B-based; Mirage and Mull-Tokens / Mull-Tokens-GRPO use Qwen2.5-VL-7B; CoVT covers both LLaVA-13B and Qwen2.5-VL-7B. Identity rows match within tolerance except for Aurora-discrete, which retrains on 19,279 ADE20K depth examples for 10 epochs (vs. the paper's curriculum mix), which may explain the higher reproduced accuracy. CoVT "Ours" averages CV-Bench-2D (76.26) and 3D (80.83). MetaMorph is architecturally similar to our continuous model but has no public checkpoint, so its paper numbers are listed in Table 11 for reference only.

A.4 KV cache and attention values

We first turned the key–value cache off over the span and the answer and recomputed every step (Qwen2.5-VL-3B, DINOv2, $K{=}64$). As expected, accuracy does not change beyond noise: identity moves by 0.00 points (95% CI $[-1.8, +1.8]$), and oracle injection and i.i.d. random replacement

Table 10: **Task/backbone/training coverage.** L13B=LLaVA-13B; Q3B/Q7B=Qwen2.5-VL-3B/7B. Training stages verified against the corresponding papers.

Method	Backbone	Training	Eval. task
Aurora discrete (Bigverdi et al., 2025)	L13B/Q3B	VQ-VAE + depth SFT	HardBLINK
Ours, depth (cont.)	L13B/Q3B	1-stage joint LM+depth	HardBLINK
Mirage (Yang et al., 2026b)	Q7B	2-stage: latent+text → text-only	VSP-SP
Mull-Tokens (Ray et al., 2026)	Q7B	2-stage SFT: anchored → free-form	BLINK/SAT
Mull-Tokens-GRPO (Ray et al., 2026)	Q7B	2-stage SFT + Stage 3 GRPO	BLINK/SAT
CoVT (Qin et al., 2025)	Q7B/L13B	4-stage curriculum	CV-Bench

Table 11: **Setup verification** (Paper baseline → method vs. Ours). Aurora baselines: LLaVA-13B base / fine-tuned-without-tokens. Mirage baseline: Direct-SFT (Qwen2.5-VL-7B). Mull-Tokens / CoVT baselines: Qwen2.5-VL-7B base. MetaMorph baseline / method: VQA-only vs. Cosine-Sim VPiT ablation on LLaMA-3 8B (their Table 3); no public checkpoint, "Ours" not reproduced.

Method	Task	Paper (base. → method)	Ours
Aurora discrete (Bigverdi et al., 2025)	HardBLINK	34.1 / 50.8 → 60.7	77.96
Mirage (Yang et al., 2026b)	VSP-SP	72.0 → 76.0	76.25
Mull-Tokens (Ray et al., 2026)	BLINK / SAT	55.33 / 59.00 → 66.80 / 77.66	63.43 / 76.33
Mull-Tokens-GRPO (Ray et al., 2026)	BLINK / SAT	55.33 / 59.00 → 67.13 / 77.00	64.71 / 77.67
CoVT (Qin et al., 2025)	CV-Bench	74.5 → 80.0	78.55
MetaMorph (Tong et al., 2025)	MMBench	73.1 → 73.8	—

move by $+0.27$ and $+1.61$. The cache cannot matter for TRT itself, since TRT replaces the span before any forward pass reads it.

Separately, during generation we overwrote the attention values stored in the cache at the span's positions, in every layer, with distribution-matched noise or with copies of the first position's values (first-repeat). This barely moves the LLaVA-13B continuous model and the off-the-shelf methods but lowers accuracy in the Qwen2.5-VL-3B discrete model (Table 12); with first-repeat values, it gives no answer after the span on 107 of 124 three-point questions. With i.i.d. random or zeroed cached values, CoVT keeps generating anchor tokens after its anchor block on 920 and 1108 of the 1200 items, and its accuracy falls from 80.83 to 68.92 and 53.50. The Qwen2.5-VL-3B discrete model may read depth information out of the span when it answers, while the other models may answer mostly from the image and the prompt.

B Additional Implementation Details

This appendix section gives the training hyperparameters and the evaluation datasets and sizes used in our runs.

B.1 Training hyperparameters (expanded)

LLaVA continuous (reported 10-epoch setting). Vision tower frozen; LoRA on the LLM ($r = 128$, $\alpha = 256$); max length 2048; AdamW with cosine LR; BF16+TF32; per-device batch size 16 on 8 GPUs (effective batch size 128); weight decay 0. Component learning rates: base/LoRA LR 2×10^{-4}; multimodal projector LR 2×10^{-5}; depth loss type cosine with normalization enabled; $\lambda_{\text{depth}} = 1.0$ for the reported continuous setting, so the depth head is trained to predict the span and the language model is trained to consume it. Training length and learning rate interact: at the reported 10-epoch horizon the base LR of 2×10^{-4} is stable, but as shown in the SigLIP2 (64) LR ablations in Table 8, extending the same LR to 50 epochs collapses accuracy to 24.73, whereas lowering it to 5×10^{-5} over that longer horizon recovers 66.67.

The "No depth loss" ablation in Table 8 is a separate run of the same SigLIP2, K=64 configuration with $\lambda_{\text{depth}} = 0$. Because the depth head H_d receives gradient only from $\mathcal{L}_{\text{depth}}$, setting the coefficient to zero leaves it at its initialization, while the depth projector P is still trained through $\mathcal{L}_{\text{LM}}$: during training the span is teacher-forced with encoder features, so P remains on the language path. The ablation therefore isolates a model that is trained to *consume* a depth span but never to *predict* one, and it reaches 72.85 against the 74.46 of the reported jointly-trained setting.

Table 12: **Attention-value checks.** Accuracy change (points) against each model's unmodified run, with 95% intervals; —: not measured. [a]Decoding collapse (see text). [b]Three-point questions only. The Qwen2.5-VL-3B discrete baseline is this check's own identity run of the Table 7 checkpoint.

Model	N	Distribution-matched noise	First-repeat
Our models (HardBLINK)			
Qwen2.5-VL-3B, discrete	372	-11.6 $[-16.6, -6.4]$	collapse[a]
LLaVA-13B, continuous	124[b]	0.0 $[-2.3, +2.3]$	-0.81
Off-the-shelf methods, each on its own benchmark			
Mull-Tokens (SAT, first 200)	200	-0.50 $[-2.7, +1.7]$	—
Mirage (VSP-SP)	400	-0.25 $[-1.3, +0.8]$	—
CoVT (CV-Bench-3D)	1200	-0.17 $[-0.6, +0.2]$	0.00

LLaVA discrete (reported 10-epoch setting). Vision tower frozen; LoRA on the LLM; max length 2048; AdamW with cosine LR; BF16+TF32; effective batch size 128. Each image is represented with $K = 100$ discrete depth tokens, where K denotes the discrete depth-token budget. The depth projector/head are linear with depth LR 1×10^{-5}; $\lambda_{\text{depth}} = 1.0$.

Qwen2.5-VL-3B (reported 10-epoch setting). Vision encoder frozen; LLM + visual MLP + embeddings are fine-tuned; AdamW with cosine LR; BF16; warmup ratio 0.03; effective batch size 128; max length 4096. The base learning rate is 2×10^{-4} for the discrete and No-aux models and 5×10^{-5} for the continuous models. Depth loss type MSE with $\lambda_{\text{depth}} = 1.0$; depth head/projector are linear and weight-tied ($W_{\text{head}} = W_{\text{proj}}^{\top}$).

B.2 Evaluation datasets and sizes (expanded)

HardBLINK. Three subsets (3/4/5-point) with 124 questions each (372 total), long-format question JSONLs; the LLaVA-13B Base-model and No-aux models use the short-format ones (Appendix C.3).

Mirage VSP-SP reproduction. `vsp-spatial-planning` with 400 test samples and 1000 train samples.

Mull-Tokens. BLINK (700 samples) and SAT (300 samples), greedy decoding; the value-level row of Table 12 uses the first 200 of these SAT items.

CoVT. CV-Bench-2D (ADE20K+COCO) and CV-Bench-3D (Depth+Distance), evaluated via VLMEvalKit (Duan et al., 2024).

Training-set sizes used locally. The training dataset used in our runs has 19,279 examples.

C Data and Evaluation Protocol

C.1 ADE point sampling generation

The point-marked images used for *training* are generated dynamically from the ADE20K dataset; the *evaluation* images are not. The generation pipeline scans the `ADE_depth` directory, matches each depth map to its corresponding RGB image, resizes both to 336×336 pixels, and samples 3 to 5 points per image.

To ensure task difficulty and spatial diversity, the sampling process enforces several constraints:

- Points are sampled within bounded coordinates ($x \in [10, 324]$, $y \in [95, 224]$).
- Points with a depth value of 0 are rejected.
- The minimum spatial distance between any two points must be at least 20 pixels.
- The minimum absolute depth difference between points must be at least 20.
- The generator uses up to 10,000 sampling attempts per image to satisfy these constraints.

Once valid points are found, they are sorted by depth, but their visual labels (A through E) are randomly shuffled so that alphabetical order does not correlate with depth order. The correct answer is defined as the label attached to the maximum depth value. Our final mixed-depth dataset contains 19,279 images, distributed as 6,736 (3-point), 6,562 (4-point), and 5,981 (5-point) examples.

Oracle. Neither ADE20K nor BLINK provides depth annotations, so every depth map here, training target and oracle alike, is an estimate from Depth Anything (ViT-S) (Yang et al., 2024); reading these maps at the marked points reproduces 362 of the 372 HardBLINK answer labels, so the depth maps behind the oracle carry the information the task needs.

C.2 Training data formats and splits

Our training data is formatted into JSON files depending on the architectural path:

- **Continuous Long (`mixed_depth_long/mixed_depth_long.json`):** Used for LLaVA and Qwen continuous training. It contains 19,279 examples pairing the ADE images with a long-form rationale. The depth target is represented by a single `<DEPTH_TOKEN>` bounded by `<DEPTH_START>` and `<DEPTH_END>`. The `embedding` key points to the corresponding precomputed `.npy` file.
- **Continuous Short (`mixed_depth_short.json`):** Contains the same 19,279 examples, but the assistant's target response is strictly the final multiple-choice letter (e.g., `(C)`).
- **Discrete Mixed-Depth Long (`mixed_depth/mixed_depth_long.json`):** Contains the same rationale format as the continuous version, but the depth span is populated with $K = 100$ discrete depth tokens per image (e.g., `<DEPTH_36><DEPTH_64>...`), where K is the discrete token budget.

C.3 Prompting protocol

We use two prompt variants for HardBLINK. The short prompt is:

> *"Multiple points are circled... Which point is the closest to the camera?"*

The long prompt appends a step-by-step scaffold that explicitly references the point coordinates, the depth map, and the rule that higher depth-map values indicate points closer to the camera.

C.4 Answer extraction

Predictions are evaluated using `eval_answers.py`. Ground-truth labels are extracted from `(A)`-style answers. The parser also supports several long-form answer patterns, including forms such as *"the answer is that point X is closer ..."*. Predictions that cannot be parsed are counted as incorrect rather than dropped.

D Depth Targets and TRT Mechanics

D.1 Continuous depth encoders

The continuous depth encoders listed in `encoder_config.json` are:

- `google/siglip2-large-patch16-256` with $K = 256$, $D = 1024$
- `facebook/dinov2-base` with $K = 256$, $D = 768$
- `openai/clip-vit-large-patch14-336` with $K = 576$, $D = 1024$
- `ait-vqvae-continuous-100x512` with $K = 100$, $D = 512$. For this model, we extract continuous codebook embeddings from the same VQ-VAE used for discrete token generation, obtaining dense per-image feature representations for each input view.

Configuration names ending in `_interploate_N` override the token horizon so that the target sequence has exactly $K = N$ tokens.

D.2 CONTINUOUS TARGET RESIZING

Continuous targets are resized to exactly K tokens before supervision. For square token grids, we use bilinear interpolation. For non-square token layouts, we use one-dimensional linear interpolation. As shown in Table 8, the best-scoring target sequence length K varies by encoder. For instance, CLIP achieves peak performance on LLaVA at $K = 64$ (73.39%), while DINOv2 performs best under extreme compression at $K = 4$ (66.94% on LLaVA), suggesting differing spatial densities in their respective feature maps. We also probed how much depth information each budget keeps: a probe on the interpolated DINOv2 encoding of the depth map recovers the answer on 48.9% of questions at K=4 and 65.9% at K=16, against 80.6% at K=64 and 79.4% for the full 16×16 grid. K=64 is therefore the smallest budget that keeps the depth information of the full grid, and even at K=64 accuracy stays insensitive to the span's content.

D.3 TRT-STYLE ABLATIONS

In continuous LLaVA mode, TRT-style ablations are

$$\{\texttt{gt}, \texttt{gt_permuted}, \texttt{random}, \texttt{zero}, \texttt{random_gt_dist}, \texttt{model}, \texttt{first_repeat}\}.$$

Here, `random_gt_dist` denotes distribution-matched random replacement: for each sample, replacement vectors are sampled from a Gaussian $\mathcal{N}(\mu, \sigma^2)$ using the mean and standard deviation of that sample's oracle depth-token embeddings.

What the two statistics are taken over differs by method: the whole oracle span for the matched-budget model; each replaced input embedding's hidden dimension for Mull-Tokens (both "Random" rows) and CoVT ("Random (same dist)") and the hidden dimension of Mirage's own latent at each step ("Random (model dist)"); and all entries of the helper-image latents for Mirage's "Random (helper-image dist)".

In continuous Qwen mode, the same ablation family is implemented through `set_depth_ablation(...)` together with constrained generation of exactly K `<DEPTH_TOKEN>` tokens after `<DEPTH_START>`.

E BACKBONE-SPECIFIC IMPLEMENTATION

E.1 BACKBONES: LLAVA AND QWEN2.5-VL

We implement the depth-span interface and TRT instrumentation in two widely used open VLM families.

LLaVA. LLaVA (Liu et al., 2023b) couples a vision encoder to an LLM via a learned multimodal projector. We extend the sequence construction by expanding each `<DEPTH_TOKEN>` placeholder into K depth positions and injecting $P(z)$ at those positions. Attention masks, position IDs, and loss masks are updated to reflect the expanded span. Depth supervision is applied via H_d only on the depth positions, while text positions are trained with the standard next-token loss.

Qwen2.5-VL. Qwen2.5-VL (Bai et al., 2025) provides a native multimodal transformer with strong localization behavior. We integrate the same depth projector/head interface into Qwen's embedding path: depth tensors are provided alongside token IDs, depth positions are masked out of the text loss, and $\mathcal{L}_{\text{depth}}$ is computed analogously to the LLaVA variant.

E.2 LLAVA CONTINUOUS PATH

In continuous LLaVA, image features are inserted at image-token positions and a single textual `<DEPTH_TOKEN>` placeholder is expanded into K depth embeddings. All depth positions are masked from the cross-entropy loss. Continuous depth loss is applied autoregressively, using hidden state $t-1$ to predict depth target t.

E.3 QWEN CONTINUOUS PATH

In continuous Qwen2.5-VL, preprocessing expands `<DEPTH_TOKEN>` into K repeated tokens, masks all depth-token labels from the cross-entropy loss, and applies depth loss using hidden state $t-1 \rightarrow$ depth target t. Position IDs follow Qwen2.5-VL's multimodal RoPE path.

During decoding, a `ContinuousDepthLogitsProcessor` forces exactly K `<DEPTH_TOKEN>` tokens after `<DEPTH_START>`, followed by one `<DEPTH_END>` token.

E.4 INTERMEDIATE LAYER DEPTH SUPERVISION

In addition to standard training, we tested whether applying the auxiliary depth loss to different intermediate transformer layers, rather than strictly the final decoder output, would change downstream performance. This allows us to evaluate the representation of spatial depth at varying levels of model abstraction.

This mechanism is controlled via the `depth_loss_layer` argument (defaulting to `-1` for the final hidden state). Setting this argument to a specific intermediate block k forces the model's forward pass to retain all hidden states (`output_hidden_states=True`) and routes `hidden_states[k+1]` to the depth loss computation, bypassing the rest of the decoder blocks for that specific objective.

Once the specified layer's state is extracted, the supervision behaves identically to the default setting: the loss is applied autoregressively such that the hidden state at position $t-1$ predicts the target depth vector at position t. The projection is handled via either a tied linear map or a separate `depth_head`, and compared against the target using MSE, cosine, or softmax loss.

This intermediate layer-selection pathway is implemented symmetrically in both the LLaVA and Qwen2.5-VL architectures. Our evaluated layer-ablation checkpoints (e.g., `depthlayer_9`, `depthlayer_19`, `depthlayer_29`, and the default `depthlayer_m1`) correspond directly to these targeted transformer blocks. The quantitative impact of supervising these intermediate layers is reported in Table 8 in Appendix A; among the evaluated intermediate-layer configurations, supervising Layer 19 for 10 epochs gives the best result (72.04%).

E.5 DISCRETE DECODING CONSTRAINTS

In discrete LLaVA decoding, custom `LogitsProcessor`s force either ground-truth depth tokens, random depth tokens, or an all-`<DEPTH_0>` sequence inside the depth span, then force `<DEPTH_END>`.

F OFF-THE-SHELF MODEL REPRODUCTION

F.1 MIRAGE

Our Mirage reproduction uses `Qwen/Qwen2.5-VL-7B-Instruct` with checkpoint `Miiche/vsp_spatial_planning_direct_sft`. Latent replacements are injected inside Mirage's modified generation loop after `<|latent_start|>`.

Mirage's oracle (helper-image) mode encodes a task-specific helper image with `model.visual(...)`—the maze with its BFS solution path drawn on it—and compresses the resulting sequence to the fixed latent count (default `latent_size=4`) by averaging groups. The count-matched oracle variant uses a two-pass protocol: first generate once, count latent tokens, then recompress the helper-image latents to that count and regenerate. The conditions of Table 1 use Mirage's released decoding (sampling at temperature 0.7 and top-p 0.9, fixed seed); the value-level runs of Table 12 decode greedily at batch size 1, and their identity scores 75.25 where Table 1 prints 76.25.

F.2 MULL-TOKENS

Our Mull-Tokens reproduction uses `array/Qwen2.5-VL-Mull` and `array/Qwen2.5-VL-MullGRPO`. The prompt template appends an assistant message containing

```
"<think>" + "<|latent_pad|>"*20 + "</think>".
```

Answers are extracted from `<answer>...</answer>`.

Mull replacements are applied to the 20 prefill `<|latent_pad|>` input embeddings. The released checkpoints' evaluation recipe carries the latent span in the prompt and never emits `<|latent_start|>`, so at inference there is no autoregressive latent to replace. The *same-dist random* setting samples each replacement vector independently from $\mathcal{N}(\mu_i, \sigma_i^2)$, where μ_i and σ_i are computed from that vector's own statistics. The oracle and oracle-distribution modes require an external helper-image directory. Since the span is 20 copies of one static pad embedding, input-level first-repeat equals identity by construction.

F.3 COVT

Our CoVT reproduction uses four VLMEvalKit-wrapped checkpoints under `Wakals/`: `CoVT-7B-depth`, `CoVT-7B-seg_depth_dino`, `CoVT-7B-seg_depth_dino_edge`, and `CoVT-LLaVA-13B-depth`; Tables 3 and 12 use `CoVT-7B-depth`.

At evaluation time, input-level ablations replace the anchor-pad token embeddings associated with tokens such as `<|sam_pad|>`, `<|dino_pad|>`, and `<|depth_pad|>`. Supported modes are

$$\{\texttt{zero}, \texttt{random}, \texttt{same}, \texttt{random_dist}, \texttt{first_repeat}\}.$$

Oracle and count-matched modes are not implemented in the CoVT evaluation wrappers. The anchor pads share one input embedding, so input-level first-repeat equals identity by construction; the attention-value check (Appendix A.4) edits the anchor positions the model generates.

G COMPUTE AND RESOURCES

The compute settings used in our experiments are summarized below: 1 node with up to 8 NVIDIA GPUs (L40S, A40, A100 or H200), 40 CPUs per task, and 300 GB RAM. 10 epochs of continuous training takes approximately 10 wall-clock hours.

H PAIRED INFERENCE FOR THE TRT TABLES

Paired inference. HardBLINK's 372 questions come from 124 images, three per image, so questions from the same image are not independent. For each paired comparison we therefore report the accuracy change with a 95% interval and a p-value computed over images rather than questions: the interval is Newcombe's paired-proportion interval (Newcombe, 1998), widened for the correlation between questions from the same image, and the p-value comes from a sign-flip test over images. For the 124 three-point questions alone (one question per image) we use the standard interval and an exact McNemar test (McNemar, 1947). Table 13 lists every condition of Table 6 against its own identity run.

Table 13: **Paired changes for Table 6** (HardBLINK, 372 questions, 124 images). Δ: condition minus comparator (points); *b*/*c*: questions only the comparator / only the condition answers correctly.

Backbone	Condition	Comparator	Δ	95% CI	b/c	p
LLaVA-13B (SigLIP2, $K = 64$)	Oracle (GT)	Identity	$+0.27$	$[-2.0, +2.6]$	$0/1$	1.00
	Slot shuffle (oracle)	Identity	0.00	$[-2.1, +2.1]$	$0/0$	1.00
	Slot shuffle (oracle)	Oracle (GT)	-0.27	$[-2.6, +2.0]$	$1/0$	1.00
	Random (i.i.d.)	Identity	-0.27	$[-2.9, +2.4]$	$2/1$	1.00
	Zero	Decoding collapse: $372/372$ (100%)				
	First-repeat	Identity	0.00	$[-2.1, +2.1]$	$0/0$	1.00
Qwen2.5-VL-3B (DINOv2, $K = 64$)	Oracle (GT)	Identity	$+1.34$	$[-0.6, +3.3]$	$4/9$	0.27
	Slot shuffle (oracle)	Identity	$+0.81$	$[-1.2, +2.8]$	$5/8$	0.58
	Slot shuffle (oracle)	Oracle (GT)	-0.54	$[-3.5, +2.4]$	$4/2$	0.69
	Random (i.i.d.)	Identity	-0.54	$[-3.1, +2.0]$	$12/10$	0.82
	Zero	Identity	$+0.54$	$[-1.5, +2.6]$	$6/8$	0.79
	First-repeat	Decoding collapse: $343/372$ (92.2%)				