RETHINKING VISUAL THOUGHTS: FROM INERT CONTEXT TO PRIVILEGED SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

Interleaved visual chain-of-thought (CoT) enables multimodal models to generate intermediate images during reasoning, under the premise that visual states capture what text poorly describes. We investigate a fundamental question: do these models actually use the images they generate? Through a targeted intervention framework, we reveal that self-generated visual thoughts remain largely question-invariant, and replacing them with noise or the original input causes little performance drop. Instead, subsequent reasoning is overwhelmingly dominated by language, treating visual tokens as inert context. Building on this insight, we propose treating visual thoughts as privileged supervision during training to transfer visual knowledge directly into the language pathway. We explore two distinct strategies: explicitly verbalizing visual states into text (Verbalized CoT, V-CoT) and implicitly aligning the hidden states of the student's own reasoning with a teacher that reads the visual thoughts (Visual-State Distillation, VSD). Extensive evaluations across diverse model backbones show that both strategies yield highly effective text-only reasoners, outperforming standard interleaved baselines on average in every recipe across varied visual reasoning domains while substantially improving overall inference efficiency at test time. All code, data, and model weights will be released upon acceptance.

1 INTRODUCTION

While chain-of-thought (CoT) reasoning externalizes intermediate computations in language (Wei et al., 2022; Kojima et al., 2022), many multimodal tasks involve intermediate visual states that are difficult to express textually. Examples include 3D object rotation, jigsaw puzzle assembly, maze navigation, and visual element isolation (Li et al., 2025a; Wu et al., 2024; Menon et al., 2024; Hu et al., 2024). This mismatch has motivated *interleaved visual CoT* (Figure 1 (a)), where multimodal models generate intermediate images and condition subsequent reasoning on them (Gu et al., 2026; Li et al., 2026; Qin et al., 2026; Wang et al., 2026a; Shi et al., 2026; Chern et al., 2025). Such visual thoughts offer a natural representation for spatial and perceptual reasoning.

However, interleaved visual CoT rests on a key assumption that has rarely been tested directly: the model actually uses the intermediate images it generates. Prior work shows that such visual traces can be plausible and interpretable (Gu et al., 2026), and that interleaved training can improve downstream accuracy (Gu et al., 2026; Li et al., 2026). Yet neither result establishes that the generated visual content itself contributes to reasoning. We therefore ask a more fundamental question:

Do subsequent predictions actually depend on what the visual thought contains?

To answer this question, we conduct an intervention-based diagnosis (Figure 1 (b)) that probes visual thoughts along two complementary dimensions: *task relevance* and *causal utility*. **First**, we vary the question while holding the input image fixed. We find that the generated visual thoughts remain largely unchanged, often resembling near-copies of the original input rather than task-specific intermediate states. **Second**, we directly intervene on the visual thought at inference time. Replacing it with either the original input image or uninformative noise causes little to no degradation in accuracy. Together, these observations reveal a striking disconnect: although the interleaved visual CoT paradigm generates seemingly meaningful visual thoughts, the subsequent reasoning depends

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

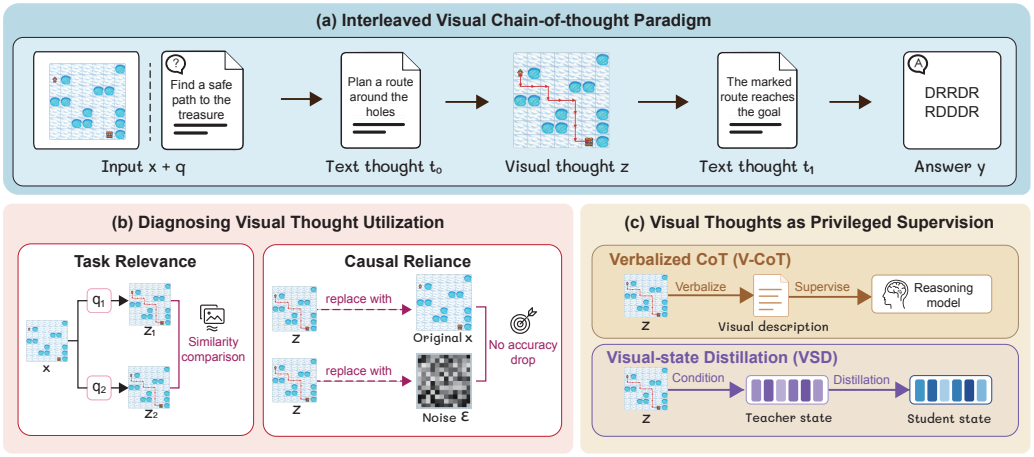


Figure 1: **(a)** Interleaved visual CoT models generate visual thoughts z to guide downstream textual reasoning. **(b)** Our diagnosis reveals that visual thoughts z lack both task relevance and causal utility: they remain question-invariant, and perturbing them causes little accuracy drop. **(c)** We reframe z as privileged training-time supervision via Verbalized CoT (V-CoT) or Visual-State Distillation (VSD), allowing the model to effectively leverage visual evidence during training.

only weakly on their actual content. Token-level perturbation and attention analyses further indicate that the reasoning process is dominated by language and treats visual tokens as inert context, which we refer to as a failure of visual assimilation.

This finding suggests a fundamentally different role for visual thoughts. Rather than feeding visual thoughts back to the reasoner as image tokens, we use them only as *privileged supervision* during training (Figure 1 (c)). The key idea is to preserve the information that visual thoughts expose while bypassing the reasoner's failure to assimilate visual tokens. We instantiate this idea through two strategies: ❶ **Verbalized CoT (V-CoT)** performs explicit knowledge transfer. By converting generated visual states into discrete textual descriptions, it makes the visual semantics directly accessible and actionable for the language pathway. ❷ **Visual-State Distillation (VSD)** performs implicit transfer at the hidden-state level. By aligning the hidden states of the student's self-generated reasoning with those of a teacher that reads the visual thoughts, VSD absorbs them without requiring explicit verbalization. Together, these strategies offer alternative ways to internalize visual thoughts during training, yielding a highly efficient, text-only reasoning process at inference time. Our contributions are:

- **Diagnosis of the Visual CoT Illusion.** We empirically reveal that state-of-the-art interleaved models do not functionally rely on the images they generate. We link this to a failure in visual assimilation, where downstream reasoning is overwhelmingly dominated by language and treats visual tokens as inert context.
- **Visual Thoughts as Privileged Supervision.** We propose a paradigm shift that leverages intermediate images exclusively as privileged training supervision. To internalize the mechanism of visual thoughts directly into the language pathway, we introduce two distinct strategies: explicit Verbalized CoT (V-CoT) and implicit Visual-State Distillation (VSD).
- **Superior Accuracy and Inference Efficiency.** Extensive evaluations across diverse model backbones demonstrate that our text-only reasoners outperform standard interleaved baselines on average in every recipe across varied visual reasoning domains. Crucially, this approach drastically reduces overall inference latency and computational overhead.

2 Related Work

Interleaved Visual Chain-of-Thought. Architecturally rooted in unified multimodal backbones that natively blend visual perception and autoregressive image generation (Team, 2024; Deng et al., 2025), early works in interleaved visual reasoning focused on optimizing image generation fidelity through specialized objectives (Li et al., 2025a). Subsequent efforts scaled and curated interleaved

reasoning traces to unlock emergent visual behaviors such as dynamic zooming and inpainting (Li et al., 2026; Gu et al., 2026), while others extended this framework to mathematical reasoning and multi-turn visual synthesis (Qin et al., 2026; Shi et al., 2026). This paradigm remains distinct from tool-augmented methods that rely on programmatic tool execution (e.g., Python code or specialized visual modules) (Hu et al., 2024; Zheng et al., 2026) and continuous latent methods that bypass explicit image rendering (Bigverdi et al., 2025; Yu et al., 2025; Wang et al., 2026b).

Despite these advancements, existing literature focuses almost exclusively on *how to synthesize* visual thoughts, operating under the unverified assumption that subsequent inference actively depends on them. Work on the faithfulness of textual CoT asks the same question for text reasoning, namely whether the stated steps drive the answer (Turpin et al., 2023), and tests it by corrupting or truncating intermediate steps and by early answering (Lanham et al., 2023). We bridge this gap by carrying this methodology over to visual thoughts, diagnosing the actual utility of these generated images, and introducing a paradigm shift: rather than feeding generated images back into the reasoning trace, we reframe visual thoughts as privileged supervision used only during training.

Training-Time Visual Guidance. A prominent line of work routes visual context into language reasoning by explicitly verbalizing initial images or tool execution traces into textual rationales and scene graphs (Zhang et al., 2023; Zheng et al., 2023; Mitra et al., 2024; Li et al., 2025b; Hu et al., 2024). Building directly on this established practice, our *Verbalized CoT (V-CoT)* strategy extends textualization to the interleaved reasoning paradigm. By verbalizing the intermediate visual state (e.g., imagined spatial manipulations) at the locus where an image would typically be rendered, we route visual knowledge into the text pathway without claiming architectural novelty.

Alternatively, implicit visual guidance draws on the foundations of cross-modal representation transfer and privileged supervision (Vapnik & Vashist, 2009; Hoffman et al., 2016; Hsieh et al., 2023; Zhao et al., 2026). While an image-conditioned teacher can supervise a text-only student, modality and capacity gaps between the two complicate direct feature matching (Liang et al., 2022; Cho & Hariharan, 2019). Our *Visual-State Distillation (VSD)* follows on-policy distillation (Agarwal et al., 2024; Zhao et al., 2026): the student is aligned with a teacher that reads the visual thoughts on trajectories the student samples itself, so the distillation signal is applied to the student's own reasoning rather than to fixed reference traces.

3 Diagnosing Visual Thought Utilization in Interleaved Models

3.1 Preliminaries

Given an input image x and a text question q, a unified model P_θ generates an interleaved reasoning trace by dynamically alternating between textual and visual thoughts:

$$t_0 \sim P_\theta(\cdot \mid x, q), \quad z_i \sim P_\theta(\cdot \mid x, q, t_{<i}, z_{<i}), \quad t_i \sim P_\theta(\cdot \mid x, q, t_{<i}, z_{\leq i}), \quad i = 1, \ldots, T, \quad (1)$$

culminating in a final answer y within the terminal text step t_T. Here, z_i denotes the i-th generated image, termed a *visual thought*. Crucially, the same underlying parameters perform a dual role: acting as a *generator* G_θ to synthesize z_i, and as a *reasoner* R_θ to decode subsequent text. In all three diagnosed models, z_i re-enters R_θ through the exact same visual encoding pipeline as the original input x. We target this specific re-entry interface with causal interventions to determine whether the reasoner functionally relies on this generated visual context.

We evaluate three open-source models spanning distinct backbones and visual generation paradigms. **ThinkMorph** (Gu et al., 2026) and **MathCanvas** (Shi et al., 2026) fine-tune Bagel-7B (Deng et al., 2025) on spatial and geometric reasoning, respectively; both generate visual thoughts via a diffusion expert and feed them back through a pixel decode followed by VAE and ViT re-encoding. **Zebra-CoT** (Li et al., 2026) fine-tunes Anole-7B (Chern et al., 2024) across multi-domain traces, representing images as discrete VQ tokens consumed directly in the native language space. All diagnostic experiments use the official released checkpoints; the interventions are evaluated on each model's source benchmarks (Appendix D.3).

The premise of interleaved visual CoT hinges on two core hypotheses: (1) that z dynamically encodes task-relevant visual states, and (2) that the reasoner R_θ functionally relies on z to derive the final answer. We diagnose the validity of the first claim through *task relevance* analysis (Section 3.2), and the second through *causal utility* interventions (Section 3.3).

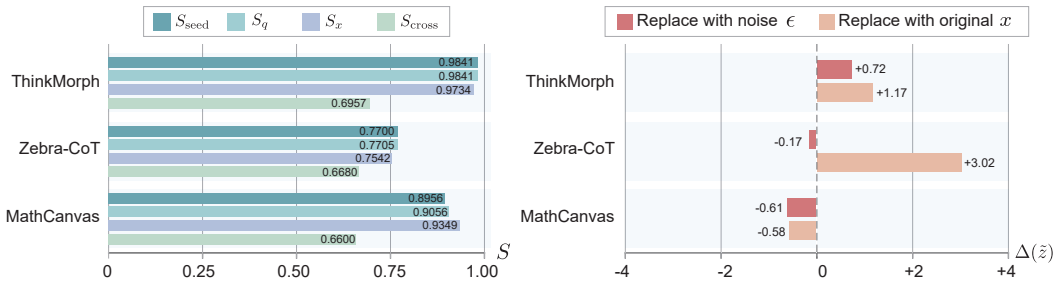


Figure 2: **(Left)** Task relevance analysis shows that visual thoughts exhibit high question invariance ($S_q \approx S_{\text{seed}}$) and input redundancy ($S_x$). **(Right)** Causal interventions reveal negligible accuracy drops when replacing the generated visual thought with noise (ϵ) or the original input (x).

3.2 Does the Visual Thought Encode Task-Relevant Information?

Given an input image x and a set of distinct questions $\{q_k\}_{k=1}^{K}$ associated with it, let z_k be the visual thought generated for q_k. Using cosine similarity $\text{sim}(\cdot,\cdot)$ over CLIP image embeddings, we quantify question invariance $S_q(x)$ and input redundancy $S_x(x)$ as

$$S_q(x) = \mathbb{E}_{k\neq k'}\,\text{sim}(z_k, z_{k'}), \quad S_x(x) = \mathbb{E}_k\,\text{sim}(z_k, x). \tag{2}$$

We evaluate S_q against two reference baselines: S_{seed}, the similarity between visual thoughts generated for the same question under different sampling seeds, and S_{cross}, the similarity between thoughts generated for different input images. An informative visual thought should yield $S_q \ll S_{\text{seed}}$; conversely, $S_q \approx S_{\text{seed}}$ indicates that, under this global embedding, the generated image varies with the question no more than with the sampling seed. S_x measures the extent to which z simply mirrors the initial input x.

Evaluating on constructed image–question sets, we isolate the first generated intermediate as the visual thought (Appendix D.1). As illustrated in Figure 2 (left), S_q is virtually indistinguishable from S_{seed} across all evaluated architectures. Concurrently, S_x remains remarkably high, while the baseline S_{cross} is substantially lower. Visual thoughts thus largely preserve the appearance of x with little question-specific differentiation. Appendix D.2 calibrates this measure with ground-truth intermediate states and a same-domain baseline.

3.3 Does the Visual Thought Provide Causal Utility?

To quantify the causal utility of visual thoughts, we intervene after z is generated but before it conditions subsequent reasoning. By holding the pre-intervention context fixed, we replace z with a control image $\tilde{z}$. The subsequent text and final answer y are decoded as $(t_1, \ldots, y) \sim P_\theta(\cdot \mid x, q, t_0, \tilde{z})$, allowing us to measure causal utility via the accuracy change:

$$\Delta(\tilde{z}) = \text{Acc}\big[\, z \leftarrow \tilde{z} \,\big] - \text{Acc}\big[\, z \,\big]. \tag{3}$$

We evaluate two controls (Appendix D.3): random Gaussian noise ($\tilde{z} = \epsilon$) and the original input image ($\tilde{z} = x$). Functional reliance requires both $\Delta(\epsilon) \ll 0$ and $\Delta(x) < 0$. This dual condition ensures that the model is genuinely sensitive to visual corruption, and that the synthesized thought provides a measurable reasoning advantage over simply reusing the raw input.

As illustrated in Figure 2 (right), replacing z with noise ϵ yields negligible accuracy fluctuations across all evaluated models. Furthermore, substituting z with the original input x generally preserves or even improves reasoning performance. Crucially, neither control condition triggers the substantial performance drop expected if the model were actively utilizing the intermediate image. Consistent with the high input redundancy observed in Section 3.2, these results show that the reasoner relies little on the synthesized visual content.

3.4 Why Are Visual Thoughts Marginalized?

The preceding results establish that generated visual thoughts exhibit minimal causal utility. We hypothesize that this marginalization occurs because the interleaved reasoning process remains

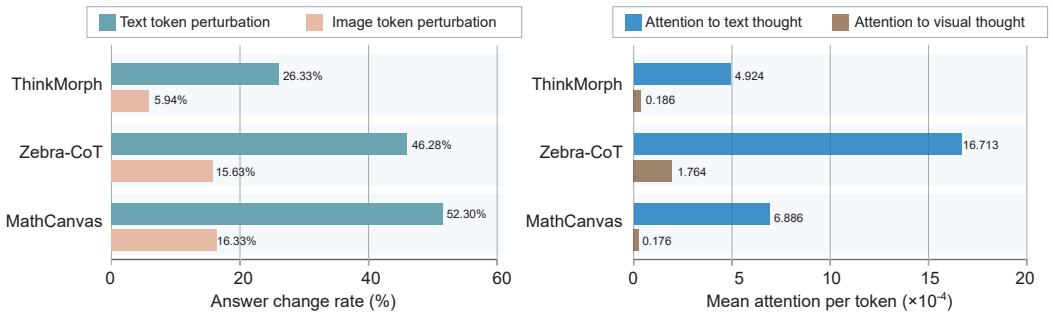


Figure 3: **(Left)** Answer change rates under modality-matched token perturbations for text and visual thoughts. **(Right)** Mean attention per token for text and visual thoughts during decoding.

overwhelmingly dominated by language, effectively rendering the visual tokens inert. To test this text-dominance hypothesis, we investigate the model's internal dependencies via two complementary lenses: modality-matched token perturbations and fine-grained attention analysis.

Reasoning is predominantly sensitive to textual perturbation. To isolate the primary driver of the final answer, we perturb either the preceding text thought t_0 or the visual thought z_1 during the first interleaved reasoning step. With the same number of replaced tokens, we replace their content tokens with random vocabulary or noise encodings while keeping the remaining context fixed, and measure how often the final answer changes (Appendix D.4). As shown in Figure 3 (left), corrupting the text thought yields a much higher answer-change rate than corrupting the visual thought across all architectures. This asymmetry indicates that the reasoning trajectory is driven mainly by the text.

Decoding attention concentrates overwhelmingly on textual context. Whereas the perturbations measure how the answer reacts when an input is corrupted, attention reveals where the model reads from while producing the answer, a lens previously used to expose text-over-image imbalances in standard VQA (Chen et al., 2024; Liu et al., 2024). Because image segments are much longer than text segments, we compute the length-normalized, per-token attention mass allocated to the text and visual thoughts (Appendix D.5). Across all evaluated architectures, a single text token receives an order of magnitude more attention than a visual token (Figure 3 (right)): the visual thought is barely consulted when the answer is formed, even though it remains in context.

4 Visual Thoughts as Privileged Supervision

The preceding diagnostics reveal a structural disconnect: the reasoner is overwhelmingly driven by text and marginalizes the generated visual tokens. We therefore repurpose visual thoughts as *privileged supervision* during training and inject them into the language pathway through two strategies (Figure 4): **Verbalized CoT (V-CoT)** verbalizes them into textual supervision, and **Visual-State Distillation (VSD)** internalizes them as hidden-state supervision.

4.1 Verbalized CoT

To bridge the structural disconnect, V-CoT explicitly translates the visual thought into text, the modality to which the reasoner naturally attends. As illustrated in Figure 4, we employ an image captioner to convert the intermediate visual thought z into a textual description $c = C(x, q, t_0, z)$ (Appendix E.1). To prevent information leakage, this captioner is strictly isolated from the subsequent reasoning t_1 and the final answer y. We then replace z with c to construct the full verbalized trajectory. The reasoning model is fine-tuned to predict the target sequence $s^V = (t_0, c, t_1, y)$ using the standard next-token cross-entropy loss:

$$\mathcal{L}_{\text{V-CoT}} = -\frac{1}{|s^V|} \sum_j \log P_\phi(s_j^V \mid x, q, s_{<j}^V), \tag{4}$$

where ϕ denotes the parameters of the reasoning model.

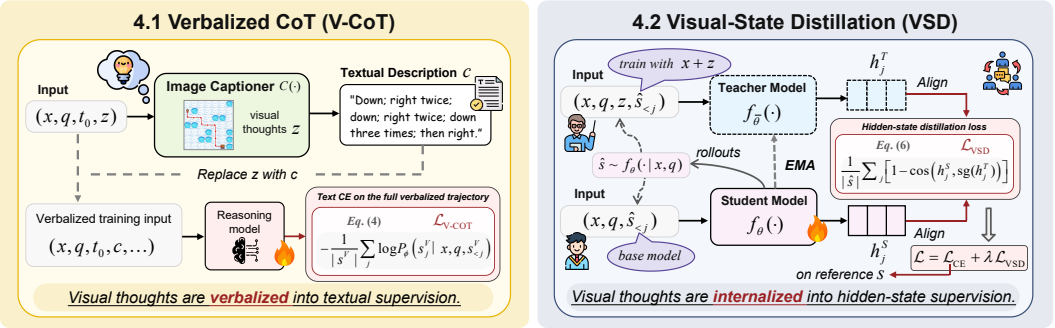


Figure 4: **(Left) Verbalized CoT (V-CoT):** The intermediate visual thought z is explicitly translated into a textual description c, constructing a fully verbalized trace for standard cross-entropy supervision. **(Right) Visual-State Distillation (VSD):** On trajectories sampled by the student, its hidden states h_j^S are aligned with those of an EMA teacher h_j^T that reads z, internalizing the visual knowledge without generating images.

4.2 Visual-State Distillation (VSD)

While V-CoT explicitly verbalizes visual content, Visual-State Distillation (VSD) bypasses the textual bottleneck by directly constraining the student's internal representations via a privileged teacher (Figure 4). The student f_θ, initialized from the base model, never sees the intermediate visual thought z. Conversely, the teacher $f_{\bar{\theta}}$ is initialized from a *reader* model, which is trained on the same textual supervision (t_0, t_1, y) as TextCoT but conditioned on both the input image x and the visual thought z. This equips the reader to exploit the supplementary information of z beyond the original input, reading visual states without ever generating them.

We adopt on-policy distillation (Agarwal et al., 2024). At each update, the student autoregressively samples a trajectory $\hat{s} \sim f_\theta(\cdot \mid x, q)$, and the teacher re-evaluates this exact sequence with z included in its context. For the j-th sampled token, we extract the corresponding last-layer hidden states from which it is predicted:

$$h_j^T = f_{\bar{\theta}}\big(x, q, z, \hat{s}_{<j}\big), \quad h_j^S = f_\theta\big(x, q, \hat{s}_{<j}\big), \tag{5}$$

and align them across the entire sampled trajectory:

$$\mathcal{L}_{\text{VSD}} = \frac{1}{|\hat{s}|} \sum_j \Big[1 - \cos\big(h_j^S,\ \text{sg}(h_j^T)\big)\Big], \tag{6}$$

where sg denotes the stop-gradient operation into the teacher. By anchoring the supervision to the states the student actually visits, VSD encourages the student to reason as if it had seen the visual thought. The overall objective combines this distillation term with a standard cross-entropy loss $\mathcal{L}_{\text{CE}}$ on the reference trace $s = (t_0, t_1, y)$, yielding $\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{VSD}}$. After each student update, the teacher is updated as an exponential moving average (EMA) of the student. Ultimately, the cross-entropy term dictates *what* the student writes, while the distillation term governs the underlying hidden state from which it writes (Appendix E.2).

5 Experiments

5.1 Setup

Training and Evaluation Recipes. To ensure generalizability, we evaluate our methods across three interleaved reasoning recipes. Within each recipe and backbone, all variants are trained from the same pretrained base model on identical source examples for one epoch. Each variant is trained with four random seeds, and all reported numbers are means over the four resulting models (Appendix F.6). All variants within a recipe are also scored with the same pipeline and answer extractor.

- **ThinkMorph:** Bagel-7B fine-tuned on 24,990 interleaved trajectories (Gu et al., 2026). Evaluated on benchmarks outside its training domains: SAT (Ray et al., 2024), CV-Bench (Tong et al., 2024a), MMVP (Tong et al., 2024b), VStar (Wu & Xie, 2024), and the BLINK suite (Fu et al., 2024).

Table 1: **Main Results across Diverse Reasoning Recipes.** All entries are accuracies (%) averaged over four training seeds; the Avg. column also reports the standard deviation across seeds. Best results within each base model group are in **bold**. BLINK-J denotes BLINK-Jigsaw, MathVis-M denotes MathVision-MINI, MathVst-M denotes MathVista-MINI, and MathCanvas-B denotes MathCanvas-Bench. MathVerse-D and -L are its Text-Dominant and Text-Lite versions.

ThinkMorph	**SAT**	**CV-Bench**	**MMVP**	**VStar**	**BLINK**	**BLINK-J**	**Avg.**
Interleave	47.3	81.9	82.7	64.9	58.0	64.0	66.5 $\pm$ 0.2
TextCoT	50.7	81.0	78.3	64.8	59.8	63.2	66.3 $\pm$ 0.3
V-CoT	52.0	82.3	**82.9**	65.4	**60.7**	**66.0**	68.2 $\pm$ 0.4
VSD	**54.0**	**82.4**	82.8	**67.0**	59.9	64.5	**68.4** $\pm$ 0.5
Zebra-CoT (Bagel)	**VisuLogic**	**EMMA**	**MathVis-M**	**MathVst-M**	**MMVP**	**VStar**	**BLINK**	**Avg.**
Interleave	27.2	21.9	17.8	61.3	71.7	42.4	50.3	41.8 $\pm$ 0.2
TextCoT	28.4	22.0	22.4	62.0	71.7	46.1	51.2	43.4 $\pm$ 0.3
V-CoT	**29.8**	21.8	**24.0**	61.9	74.0	**50.3**	**52.7**	**44.9** $\pm$ 0.2
VSD	28.0	**22.2**	22.4	**64.2**	**74.3**	48.7	**52.7**	44.6 $\pm$ 0.3
Zebra-CoT (Anole)	**VisuLogic**	**EMMA**	**MathVis-M**	**MathVst-M**	**MMVP**	**VStar**	**BLINK**	**Avg.**
Interleave	19.2	14.8	8.2	15.5	48.2	30.9	32.8	24.2 $\pm$ 0.2
TextCoT	**19.8**	16.2	10.9	18.7	48.5	30.5	34.1	25.5 $\pm$ 0.4
V-CoT	19.7	16.7	**11.5**	20.9	47.7	31.4	**34.6**	26.1 $\pm$ 0.3
VSD	19.0	**16.9**	10.9	**22.0**	**49.0**	**41.4**	33.0	**27.5** $\pm$ 0.2
MathCanvas	**MathCanvas-B**	**MathVision**	**MathVista-GPS**	**MathVerse-D**	**MathVerse-L**		**Avg.**
Interleave	12.6	25.1	76.0	62.1	58.3		46.8 $\pm$ 0.4
TextCoT	13.0	24.8	75.7	62.1	57.6		46.6 $\pm$ 0.3
V-CoT	**13.4**	**27.6**	**81.7**	**65.1**	60.1		**49.6** $\pm$ 0.3
VSD	**13.4**	27.3	76.9	63.8	**60.9**		48.5 $\pm$ 0.2

- **Zebra-CoT:** Bagel-7B and Anole-7B, each fine-tuned on 10,828 training examples (Li et al., 2026). Evaluated on VisuLogic (Xu et al., 2026), EMMA (Hao et al., 2025), the MINI splits of MathVision (Wang et al., 2024) and MathVista (Lu et al., 2024), alongside the aforementioned MMVP, VStar, and BLINK.
- **MathCanvas:** Bagel-7B fine-tuned on 10,000 training examples (Shi et al., 2026). Evaluated on MathCanvas-Bench (Shi et al., 2026), the Text-Dominant/Text-Lite versions of MathVerse (Zhang et al., 2024), and the full MathVision and MathVista-GPS suites.

Comprehensive hyperparameters and implementation details are deferred to Appendix F.

Baselines and Variants. We evaluate our proposed **V-CoT** (Section 4.1) and **VSD** (Section 4.2) against two primary baselines. **Interleave** represents the standard paradigm, which trains on the traces with intermediate images in place and requires explicitly generating images at inference. **TextCoT** serves as the universal text-only baseline across all recipes, directly derived by stripping all intermediate images from the Interleave traces while preserving the same textual reasoning steps.

5.2 Main Results

Table 1 presents the main evaluation across all reasoning recipes and benchmarks. A striking observation is that the standard Interleave paradigm frequently struggles to extract utility from its own generated images. Relying exclusively on text-only reasoning (TextCoT) yields comparable performance to Interleave on ThinkMorph and MathCanvas, and outperforms it by 1.6 and 1.3 points on the two Zebra-CoT backbones. This corroborates our earlier diagnosis: forcing the model to generate intermediate images that it subsequently ignores disrupts the reasoning trace, making a pure textual baseline a resilient competitor.

Both of our proposed paradigms successfully overcome this bottleneck. By effectively integrating visual information into the reasoning process, **V-CoT** and **VSD** outperform both the Interleave baseline and TextCoT on average in every recipe. The gains are broad rather than driven by a single benchmark: V-CoT improves over TextCoT on 21 of the 25 benchmark columns of Table 1, and VSD on 20. Some of the clearest gains appear where removing the intermediate images hurts. On MMVP,

Table 2: **Controls for the visual thought.** Average accuracy (%) over each recipe's benchmarks.

Variant	ThinkMorph	Zebra-CoT (Bagel)	Zebra-CoT (Anole)	MathCanvas
V-CoT (shuffled description)	61.3	39.2	20.6	45.5
V-CoT	**68.2**	**44.9**	**26.1**	**49.6**
VSD (noise teacher)	62.3	40.0	23.4	44.2
VSD (shuffled teacher)	61.9	38.5	23.8	45.8
VSD (untrained teacher)	66.4	43.1	25.3	45.9
VSD	**68.4**	**44.6**	**27.5**	**48.5**

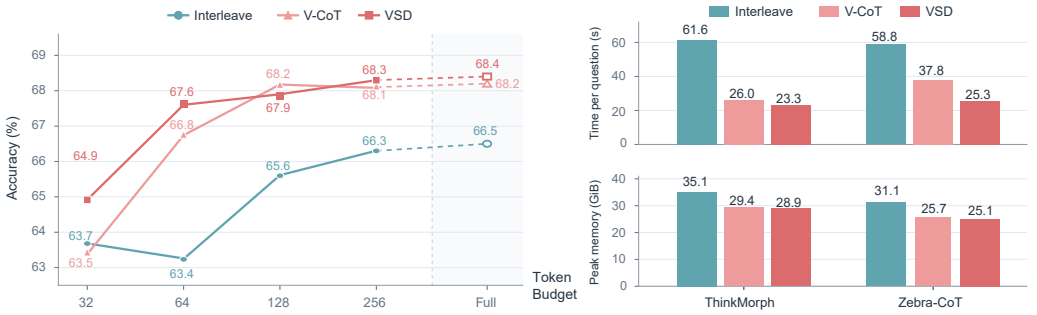


Figure 5: **Forced early exit and efficiency. (Left)** Accuracy on ThinkMorph under a text-token budget B; *Full* denotes the untruncated trace. **(Right)** Latency (top) and peak memory (bottom).

TextCoT, which is trained on the same traces without the intermediate images, falls 4.4 points below Interleave on ThinkMorph, whereas V-CoT and VSD, which also generate no images at inference, match or slightly exceed Interleave (82.9 and 82.8 versus 82.7). On the full MathVision set of 3,040 problems, V-CoT and VSD improve over TextCoT by 2.8 and 2.5 points on MathCanvas. The two methods perform comparably on average, with VSD slightly ahead on ThinkMorph and Zebra-CoT (Anole) and V-CoT on Zebra-CoT (Bagel) and MathCanvas.

5.3 The Causal Role of the Visual Thought

To confirm that the performance improvements stem explicitly from the correct visual thought rather than architectural or formatting artifacts, we evaluate each method against controls that retain the structural machinery but remove the useful visual signal (Table 2; construction details in Appendix F.4).

V-CoT relies on an accurate visual thought. In this control, each trace receives the description of a different sample, which keeps the format and position of the description but no longer matches the image. Accuracy drops sharply on every recipe (Table 2), showing that the gain comes from what the description conveys about the visual thought rather than from the extra text tokens.

VSD requires an informed teacher. For VSD, we blind the teacher by replacing the visual thought in its context with noise or with the visual thought of a different trace. Both blinded controls fall well below VSD, so the distillation loss is not a generic regularizer. An untrained teacher, which sees the true visual thought but is initialized from the base model rather than the reader, falls 1.5–2.6 points below VSD and essentially reduces to TextCoT. Since this teacher was never trained to use the visual thought, its hidden states carry little information beyond what the student already has.

5.4 Temporal Dynamics of Visual Thought Integration

To determine when critical visual information is effectively integrated into the reasoning trace, we conduct a forced early-exit evaluation on ThinkMorph. We truncate the autoregressive generation after a strict budget of B text tokens (excluding interleaved images), append the `</think>` token, and force the model to yield a final answer based on this truncated prefix (details in Appendix G.2).

As shown in Figure 5 (left), both **V-CoT** and **VSD** exhibit a dramatically faster assimilation of visual knowledge. Even under severely constrained text budgets, both methods quickly surpass the *full-trace*

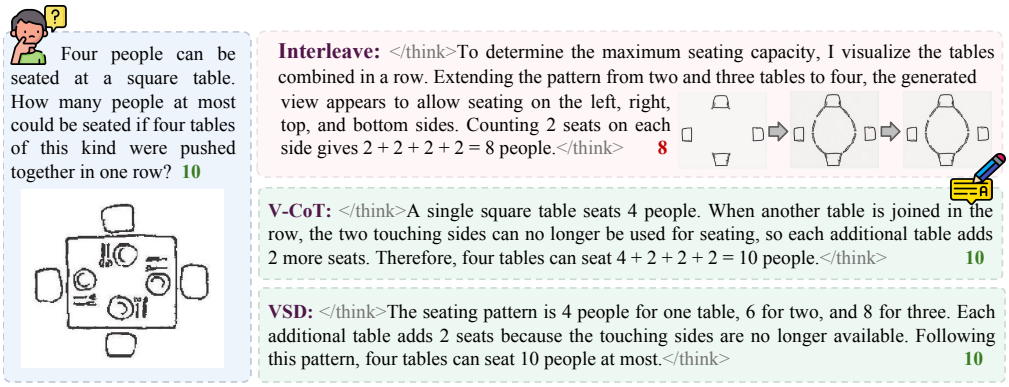


Figure 6: **Case study on ThinkMorph.** Interleave, V-CoT, and VSD on the same question.

accuracy of the Interleave model. For **V-CoT**, the performance trajectory directly reflects the budget required to verbalize the visual context; once this brief description is generated, the language pathway immediately leverages it to finalize reasoning. **VSD** leads at the smallest budgets, since its hidden states are aligned with an image-reading teacher from the first token and it needs no verbalization. Conversely, the Interleave model suffers from delayed reasoning: despite generating the image early, it stagnates initially and requires a much longer textual trace to extract utility from its visual output.

5.5 Efficiency

Figure 5 (right) details the inference latency and peak memory footprint across the ThinkMorph and Zebra-CoT (Bagel) models (protocol in Appendix G.1). Both of our proposed paradigms fundamentally resolve the computational bottlenecks of the Interleave baseline. By completely eliminating the high-cost process of image synthesis and subsequent re-encoding, **V-CoT** yields massive reductions in both processing time and peak memory. Building upon these shared gains, **VSD** achieves the ultimate efficiency configuration. Because VSD internalizes the visual thought during training, it also avoids generating the description at inference and attains the lowest latency and memory footprint.

5.6 Case Study

In Figure 6, the Interleave model generates an incorrect view of the joined tables, and its text miscounts the seats (8). V-CoT and VSD instead unfold the imagined intermediate visual process in text: V-CoT describes how joined tables lose their touching sides, and VSD tracks the seating pattern as tables are added, both reaching the correct answer of 10. This exemplifies our core premise: internalizing visual dynamics directly into the language pathway provides a far more reliable foundation for spatial deduction than relying on explicitly synthesized, error-prone image tokens. More examples are given in Appendix I.

6 Conclusion

Interleaved visual chain-of-thought rests on an appealing premise: multimodal models should reason more effectively by generating intermediate visual states. Our diagnosis reveals that current interleaved paradigms fail to fulfill this promise. The generated visual thoughts remain largely question-invariant, and the subsequent language reasoning functionally ignores them—a systemic disconnect we term the "Visual CoT Illusion". Consequently, we advocate shifting visual thoughts from inert context to privileged training supervision. We achieve this by explicitly verbalizing visual states into text (**V-CoT**) or implicitly distilling them into the student's hidden states (**VSD**). Both strategies yield highly efficient, text-only reasoners that outperform standard interleaved baselines on average in every recipe. Ultimately, this points to a paradigm shift: the true value of visual intermediates lies in the knowledge internalized during training. We hope this work paves the way for a more grounded and effective paradigm in multimodal reasoning.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic training data: the textual descriptions of visual thoughts used to train V-CoT (Section 4.1) are produced by DeepSeek-V4.1-Flash, as described in Appendix E.1. All other training traces come from the released ThinkMorph, Zebra-CoT, and MathCanvas datasets. For evaluation, we use GPT-4.1 as an LLM judge to extract and score final answers (Appendix F.3). We have not used generative AI tools for proposing hypotheses, designing experiments, implementing methods, or interpreting results, and the remaining tasks with required disclosure are not applicable to this work. Additionally, we used generative AI tools to edit the paper for readability, draft parts of the appendix, and format references. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3836–3845. IEEE, 2025.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024.

Ethan Chern, Jiadi Su, Yan Ma, and Pengfei Liu. Anole: An open, autoregressive, native large multimodal models for interleaved image-text generation. *arXiv preprint arXiv:2407.06135*, 2024.

Ethan Chern, Zhulin Hu, Steffi Chern, Siqi Kou, Jiadi Su, Yan Ma, Zhijie Deng, and Pengfei Liu. Thinking with generated images. *arXiv preprint arXiv:2505.22525*, 2025.

Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4794–4802. IEEE, 2019.

DeepSeek-AI. DeepSeek-V4.1-Flash: Pushing the limits of KV cache compression. *arXiv preprint arXiv:2609.19969*, 2026.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

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

Jiawei Gu, Yunzhuo Hao, Huichen Wang, Linjie Li, Michael Qizhe Shieh, Yejin Choi, Ranjay Krishna, and Yu Cheng. ThinkMorph: Emergent properties in multimodal interleaved chain-of-thought reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 141405–141447, 2026.

Yunzhuo Hao, Jiawei Gu, Huichen Will Wang, Linjie Li, Zhengyuan Yang, Lijuan Wang, and Yu Cheng. Can MLLMs reason in multimodality? EMMA: An enhanced multimodal reasoning benchmark. *arXiv preprint arXiv:2501.05444*, 2025.

Judy Hoffman, Saurabh Gupta, and Trevor Darrell. Learning with side information through modality hallucination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 826–834, 2016.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the association for computational linguistics: ACL 2023*, pp. 8003–8017, 2023.

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

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

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Ang Li, Charles L. Wang, Deqing Fu, Kaiyu Yue, Zikui Cai, Wang Zhu, Ollie Liu, Peng Guo, Willie Neiswanger, Furong Huang, Tom Goldstein, and Micah Goldblum. Zebra-CoT: A dataset for interleaved vision-language reasoning. In Carl Vondrick, Bharath Hariharan, Colin Raffel, Lerrel Pinto, Diyi Yang, and Aleksandra Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 59021–59061, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/6087a60306544be7ba0d0cf34aa93c8f-Paper-Conference.pdf.

Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. Imagine while reasoning in space: Multimodal visualization-of-thought. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 36340–36364. PMLR, 2025a. URL https://proceedings.mlr.press/v267/li25cz.html.

Zejun Li, Ruipu Luo, Jiwen Zhang, Minghui Qiu, Xuan-Jing Huang, and Zhongyu Wei. VoCoT: Unleashing visually grounded multi-step reasoning in large multi-modal models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3769–3798, 2025b.

Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. *Advances in neural information processing systems*, 35:17612–17625, 2022.

Shi Liu, Kecheng Zheng, and Wei Chen. Paying more attention to image: A training-free method for alleviating hallucination in LVLMs. In *European Conference on Computer Vision*, pp. 125–140. Springer, 2024.

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*, volume 2024, pp. 23439–23554, 2024.

Sachit Menon, Richard Zemel, and Carl Vondrick. Whiteboard-of-Thought: Thinking step-by-step across modalities. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 20016–20031, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1117. URL https://aclanthology.org/2024.emnlp-main.1117/.

Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. Compositional chain-of-thought prompting for large multimodal models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14420–14431. IEEE, 2024.

OpenAI. Introducing GPT-4.1 in the API. https://openai.com/index/gpt-4-1/, 2025.

Luozheng Qin, GONG JIA, Yuqing Sun, Tianjiao Li, Haoyu Pan, Mengping Yang, Xiaomeng Yang, Chao Qu, Zhiyu Tan, and Hao Li. Uni-CoT: Towards unified chain-of-thought reasoning across text and vision. In *International Conference on Learning Representations*, volume 2026, pp. 15999–16028, 2026.

Arijit Ray, Jiafei Duan, Ellis Brown, Reuben Tan, Dina Bashkirova, Rose Hendrix, Kiana Ehsani, Aniruddha Kembhavi, Bryan A Plummer, Ranjay Krishna, et al. SAT: Dynamic spatial aptitude training for multimodal language models. *arXiv preprint arXiv:2412.07755*, 2024.

Weikang Shi, Aldrich Yu, Rongyao Fang, Houxing Ren, Ke Wang, Aojun Zhou, Changyao Tian, Xinyu Fu, Yuxuan Hu, Zimu Lu, et al. MathCanvas: Intrinsic visual chain-of-thought for multimodal mathematical reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27933–27954, 2026.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai C Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. *Advances in Neural Information Processing Systems*, 37: 87310–87356, 2024a.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal LLMs. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9568–9578. IEEE, 2024b.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, 2023.

Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural networks*, 22(5-6):544–557, 2009.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with MATH-Vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2024.

Pengjie Wang, Linger Deng, Zujia Zhang, Shaojie Zhang, Zhenbo Luo, Pei Fu, Jian Luan, Xiang Bai, and Yuliang Liu. DeltaV: Thinking with visual state updates in unified large multimodal models. *arXiv preprint arXiv:2607.08434*, 2026a.

Qixun Wang, Yang Shi, Yifei Wang, Yuanxing Zhang, Pengfei Wan, Kun Gai, Xianghua Ying, and Yisen Wang. Monet: Reasoning in latent visual space beyond image and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12030–12040, 2026b.

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

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

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.

Weiye Xu, Jiahao Wang, Weiyun Wang, Zhe Chen, Wengang Zhou, Aijun Yang, Lewei Lu, Houqiang Li, Xiaohua Wang, Xizhou Zhu, et al. VisuLogic: A benchmark for evaluating visual reasoning in multi-modal large language models. In *International Conference on Learning Representations*, volume 2026, pp. 25966–26003, 2026.

Runpeng Yu, Xinyin Ma, and Xinchao Wang. Introducing visual perception token into multimodal large language model. *arXiv preprint arXiv:2502.17425*, 2025.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. MathVerse: Does your multi-modal LLM truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pp. 169–186. Springer, 2024.

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

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Ge Zheng, Bin Yang, Jiajin Tang, Hong-Yu Zhou, and Sibei Yang. DDCoT: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models. *Advances in Neural Information Processing Systems*, 36:5168–5191, 2023.

Ziwei Zheng, Minghao Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. DeepEyes: Incentivizing "thinking with images" via reinforcement learning. In Carl Vondrick, Bharath Hariharan, Colin Raffel, Lerrel Pinto, Diyi Yang, and Aleksandra Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 126775–126798, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/cdb347d7516d52a9280ea9d6708911f6-Paper-Conference.pdf.

A Overview of Appendix

- Appendix B: **Limitations**.
- Appendix C: **Diagnosed Models and Benchmarks**.
- Appendix D: **Diagnosis Protocol**: task relevance, causal intervention, token perturbation, attention aggregation, and oracle visual thoughts.
- Appendix E: **Implementation of V-CoT and VSD**.
- Appendix F: **Training and Evaluation Details**, including the control variants and the variance across training seeds.
- Appendix G: **Efficiency and Forced Early Exit**.
- Appendix H: **Prompt Templates**.
- Appendix I: **Qualitative Examples**.

B Limitations

First, all models in this study are 7B unified models, the largest open backbones that natively generate interleaved images at the time of writing. We have not scaled our diagnosis or our methods to larger models, so whether larger reasoners make better use of their visual thoughts, and whether V-CoT and VSD keep their advantage at scale, remains open. Within this scale, we diagnose three representative interleaved models that cover both ways in which a visual thought re-enters the reasoner (VAE and ViT features for the Bagel-based ThinkMorph and MathCanvas, and discrete VQ tokens for Zebra-CoT (Anole)) and three reasoning domains (spatial, geometric, and multi-domain). All of them are trained by supervised fine-tuning on interleaved traces. Models trained with other objectives, such as reinforcement learning with outcome rewards, may use their visual thoughts differently, and our conclusions are stated for the models studied here. Second, V-CoT requires an additional data-generation step: every visual thought in the training traces must be described by an external captioner (Appendix E.1), which is expensive to run at scale. Because of this cost, we train on subsets of 10,828 Zebra-CoT traces and 10,000 MathCanvas traces rather than the full training sets, with all variants using the same subsets. How the methods behave with larger training sets remains to be studied. Third, VSD adds training cost: each recipe requires an additional reader model, and every student update involves on-policy sampling and a teacher forward pass.

C Diagnosed Models and Benchmarks

C.1 Diagnosed Models

Table 3 summarizes the three released checkpoints diagnosed in Section 3.1. ThinkMorph and MathCanvas share the Bagel-7B backbone: a visual thought is produced by the diffusion generation expert, decoded to pixels, and re-encoded by the VAE and the ViT understanding encoder before the next text step. Zebra-CoT (Anole-7B) represents images as discrete VQ tokens, so a visual thought enters the reasoner as a span of codebook tokens in the shared vocabulary. Every intervention in Appendix D acts at the re-entry interface of the respective model. The intervention experiments use sampled subsets of each model's source benchmarks (Appendix D.3), and the task-relevance analysis uses constructed image–question sets (Appendix D.1).

Table 3: **Diagnosed interleaved models.**

Model	Backbone	Training domain	Visual thought	Re-entry interface
ThinkMorph	Bagel-7B	Spatial reasoning	Diffusion expert	Pixel decode $\rightarrow$ VAE + ViT encode
MathCanvas	Bagel-7B	Geometric reasoning	Diffusion expert	Pixel decode $\rightarrow$ VAE + ViT encode
Zebra-CoT	Anole-7B	Multi-domain	Discrete VQ tokens	VQ token span

C.2 Evaluation Benchmarks

Table 4 lists the benchmarks used for the main evaluation (Section 5.2); the diagnosis uses subsets of them (Appendix D.3). ThinkMorph is evaluated only on benchmarks outside its training domains. Splits and sizes follow the official releases. BLINK uses its validation split because test answers are not publicly released, and MathVerse-D/L denote the Text-Dominant and Text-Lite versions of the MathVerse testmini set.

Table 4: **Benchmarks per recipe.**

Recipe	Benchmark	Focus	Split	#Samples
ThinkMorph	SAT	Dynamic spatial reasoning	Real test	150
	CV-Bench	2D/3D vision-centric perception	test	2,638
	MMVP	Fine-grained visual patterns	full	300
	VStar	High-resolution visual search	full	191
	BLINK	Core visual perception	val	1,901
	BLINK-Jigsaw	Jigsaw completion	val	150
Zebra-CoT	VisuLogic	Visual logical reasoning	full	1,000
	EMMA	Multi-discipline multimodal reasoning	full	2,788
	MathVision	Competition mathematics	testmini	304
	MathVista	Mathematical reasoning in visual contexts	testmini	1,000
	MMVP	Fine-grained visual patterns	full	300
	VStar	High-resolution visual search	full	191
	BLINK	Core visual perception	val	1,901
MathCanvas	MathCanvas-Bench	Visual-aided mathematics	test	3,079
	MathVision	Competition mathematics	test	3,040
	MathVista-GPS	Geometry problem solving	testmini (GPS)	208
	MathVerse-D	Diagram-dependent mathematics	testmini (Text-Dominant)	788
	MathVerse-L	Diagram-dependent mathematics	testmini (Text-Lite)	788

D Diagnosis Protocol

All diagnostics use the released checkpoints and official inference code without further training. Decoding follows the defaults of each released inference script and is identical across the native, noise, and input conditions of an experiment.

Decoding. ThinkMorph and MathCanvas sample text at temperature 0.3 and generate each image with 50 denoising steps (timestep shift 3.0; classifier-free guidance of 4.0 on the text condition and 2.0 on the image condition). ThinkMorph allows up to $4{,}096$ tokens per text segment and three text–image rounds, and MathCanvas up to $2{,}048$ tokens per text segment and ten text or image steps. Zebra-CoT (Anole) samples text with temperature 1.0 and top-p 0.9, and image tokens with temperature 1.0 under its default classifier-free guidance (2.0 for the full context and 1.2 for the image-only context). On benchmark questions, its generation budget is $16{,}384$ tokens, following the Zebra-CoT evaluation, with the image limit raised to 16 so that the token budget is always reached first; on the image–question sets of Appendix D.1, it uses the script defaults of $6{,}144$ tokens and at most four images.

Image resolution. Input images are resized only by each model's own preprocessing. Anole resizes the shorter side to 512 pixels and center-crops to 512×512 ($1{,}024$ VQ tokens). The Bagel-based models rescale the image so that its longer side is at most $1{,}024$ pixels and its shorter side at least 512 pixels, with the ViT input capped at 980 pixels on the longer side. Zebra-CoT (Anole) always generates visual thoughts at 512×512; the Bagel-based models generate them at the size of the preprocessed input, i.e., 512×512 for all MathCanvas inputs and for the constructed images, and up to $1{,}024$ pixels on the longer side for ThinkMorph's benchmark images.

D.1 Task Relevance Measurement

Image–question sets. We construct 500 input images, each paired with $K = 4$ distinct questions. The images are rendered programmatically at 512×512 pixels, 125 from each of four domains (bar charts, coordinate geometry, piecewise-linear function plots, and grid mazes). The questions are templated so that all four apply the same operation to different targets in the same image: the difference between a different pair of bars, the distance from a different vertex to a different line, the area under the graph over a different interval, or the shortest path between a different start–goal pair. All answers are computed and verified programmatically. For each pair (x, q_k), the model generates a full interleaved trace, and we take the first generated image as z_k. Traces without a generated image are excluded; ThinkMorph, Zebra-CoT (Anole), and MathCanvas generate an image in $5{,}990$, $5{,}120$, and $3{,}547$ of their $6{,}000$ traces (500 images $\times$ 4 questions $\times$ 3 seeds), respectively.

Embedding. $\mathrm{sim}(\cdot, \cdot)$ in Eq. (2) is the cosine similarity between CLIP ViT-B/32 image embeddings. Each image is padded to a square with a white border before the standard CLIP preprocessing, so that the center crop removes no content.

Reference baselines. S_{seed} averages sim over all pairs of the $S = 3$ visual thoughts generated for the same (x, q) with different sampling seeds. S_{cross} averages sim over all pairs of visual thoughts generated for input images from different domains, weighting questions, reference images, and domains equally. All four quantities are averaged over images and shown in Figure 2 (left).

D.2 Calibration of the Task-Relevance Measure

A global CLIP embedding may change little when only a small, task-relevant region of an image changes, and the four questions of an image share most of its content. We calibrate the measure of Section 3.2 on the image–question sets of Appendix D.1 in three ways.

Ground-truth intermediate states. Because the images are rendered programmatically, we also render the correct intermediate state for each (x, q_k): the queried pair of bars highlighted, the perpendicular from the queried vertex to the queried line, the area under the graph over the queried interval shaded, and the shortest path between the queried start and goal drawn on the maze. These states differ from each other only in task-relevant regions. Applying Eq. (2) to them gives $S_q^{\mathrm{GT}} = 0.82$: task-relevant edits of this kind lower the similarity by about 0.18, whereas S_q of the generated visual thoughts does not fall below S_{seed} for any model (Figure 2, left).

Matching accuracy. For each generated visual thought z_k, we check, under the same CLIP similarity, whether it is more similar to the ground-truth state of its own question than to those of the other three questions of the same image; chance level is 25%. The accuracy is 27.8–31.2% across

models (Table 5), only slightly above chance, so a generated visual thought carries little information about which question it serves.

Same-domain baseline. $S_{\text{cross}}^{\text{same}}$ is computed like S_{cross} but over pairs of images generated for different inputs of the same domain, which share layout and style. It ranges from 0.67 to 0.74 (Table 5) and stays below S_q for every model (Figure 2, left): visual thoughts for different questions on the same image are more alike than those for different images of the same domain, so their similarity stems from the shared input rather than from a shared layout.

Table 5: **Calibration of the task-relevance measure.** $S_{\text{cross}}^{\text{same}}$ is computed over images for different inputs of the same domain; for the ground-truth row, it is computed over the ground-truth states. Matching accuracy has a chance level of 25%.

	$S_{\text{cross}}^{\text{same}}$	**Match. acc. (%)**
Ground-truth states	0.61	–
ThinkMorph	0.74	29.4
Zebra-CoT (Anole)	0.70	31.2
MathCanvas	0.67	27.8

D.3 Causal Intervention

Each intervention proceeds in three stages: (i) the model decodes t_0 and generates z from (x, q); (ii) before z is fed back, it is replaced by a control image $\tilde{z}$; (iii) decoding resumes from $(x, q, t_0, \tilde{z})$ until the final answer. Because stage (i) is shared, the native, noise, and input conditions of a sample have the same prefix up to the first visual thought, and $\Delta(\tilde{z})$ in Eq. (3) is computed on paired samples.

Bagel-based models (ThinkMorph, MathCanvas). The replacement is applied to the decoded pixels before re-encoding, so $\tilde{z}$ enters both the VAE and the ViT pathways and every later context that receives the image. The noise control ϵ is a Gaussian image at the resolution of z whose RGB values are drawn i.i.d. from $\mathcal{N}(0.5, 0.25^2)$ on the $[0, 1]$ scale, then clipped to $[0, 1]$ and quantized to 8 bits. The input control uses x, resized with its aspect ratio preserved (bicubic) and padded with white to the resolution of z.

Zebra-CoT (Anole). The replacement is applied to the VQ tokens of z. The input control encodes x with the same VQ tokenizer after the same aspect-preserving padding to 512×512; the noise control encodes the 512×512 Gaussian image defined above with the same tokenizer. In both controls, the $1{,}024$ codes between the begin- and end-of-image tokens are replaced and the boundary tokens are kept; the native condition keeps the generated codes.

Multi-image traces. All visual thoughts are replaced, not only the first. After the shared first image, each condition is decoded independently, and every later visual thought is replaced by the same type of control before it is fed back (a freshly sampled noise image, or x again).

Evaluation. Each model is evaluated on benchmarks from its own paper: 500 questions for Zebra-CoT (Anole) (MathVision-mini, MathVista-mini, VisuLogic, EMMA, MMVP, BLINK, and VStar), 306 for ThinkMorph (its out-of-domain benchmarks VStar, BLINK, MMVP, SAT, and CV-Bench), and 500 questions for MathCanvas (MathCanvas-Bench, MathVision, MathVista-GPS, MathVerse-D, and MathVerse-L), sampled within each benchmark by a fixed hash of the question ID. Only single-image questions are used, since the input control is undefined when a question has several input images; this excludes BLINK-J and the multi-image BLINK subtasks. Δ is computed on the questions whose native trace contains at least one visual thought (466, 302, and 495, respectively); MMVP pairs and SAT circular groups are kept intact and scored as units. For MMVP, this is the official pair metric, whereas Table 1 scores MMVP per question (Appendix F.3). Accuracy and Δ are computed within each benchmark and then averaged over the benchmarks of a model with equal weight. Answers are extracted and scored with each benchmark's official pipeline and metric (complete accuracy for MathCanvas-Bench; GPT-4.1 as the extractor or judge where the pipeline

uses an LLM, with VLMEvalKit's random-choice fallback disabled), and unparsable answers count as incorrect. The unparsable-answer rates under the native, noise, and input conditions are 11.6%, 15.5%, and 9.0% for Zebra-CoT (Anole), 0.0%, 0.7%, and 0.0% for ThinkMorph, and 0.6%, 0.8%, and 0.6% for MathCanvas.

Positive controls. To verify that these subsets leave room for the interventions to show an effect, we add two conditions that remove visual information at the input: the input image is replaced by the Gaussian noise image of the noise control before decoding starts ($x \leftarrow \epsilon$), so that z is also generated from noise, or it is removed from the prompt altogether (blind). If the questions depend on the image, both conditions should lower accuracy clearly, and an intervention on z that leaves accuracy unchanged then indicates that the model does not rely on z. Table 6 reports the native accuracy and the change in accuracy under each of the four interventions; the drop under the blind condition also shows how far accuracy can fall when the image is unavailable. Because ThinkMorph is evaluated above only on benchmarks outside its training domains, we also apply the interventions to 410 questions from its in-domain jigsaw assembly, spatial navigation, and visual search test sets; for multi-image questions, only the noise control is applied. Removing or corrupting the input image lowers accuracy on every subset, by 6.4 to 18.0 points without the image and by 8.2 to 32.6 points with a noise input (Table 6), so the subsets depend on visual input. With native accuracy of 28.2% to 73.9%, accuracy is also far from the floor, as the drops under the input controls show. In contrast, replacing the generated visual thought changes accuracy by at most 3.0 points, which is less than half of the smallest drop under the input controls, and by less than 0.2 points on ThinkMorph's in-domain tasks. Apart from the gain of Zebra-CoT (Anole) under $z \leftarrow x$, these changes are within one standard error of the accuracy (2.1 to 2.7 points). The null effect of the interventions on z is therefore not explained by questions that can be solved without the image or by accuracy near the floor.

Table 6: **Positive controls for the causal intervention.** Native accuracy z (%) of the released checkpoints on the intervention subsets, and change in accuracy (points) relative to it under each intervention. Blind removes the input image, $x \leftarrow \epsilon$ replaces it with noise before decoding, and $z \leftarrow \epsilon$ and $z \leftarrow x$ are the interventions of Section 3.3 (Figure 2, right).

Model	n	z	Blind	$x \leftarrow \epsilon$	$z \leftarrow \epsilon$	$z \leftarrow x$
ThinkMorph (out-of-domain)	302	65.3	-18.0	-32.6	$+0.72$	$+1.17$
ThinkMorph (in-domain)	410	73.9	-16.2	-28.8	-0.12	$+0.08$
Zebra-CoT (Anole)	466	28.2	-6.4	-8.2	-0.17	$+3.02$
MathCanvas	495	32.0	-12.8	-20.9	-0.61	-0.58

D.4 Modality-Matched Token Perturbation

For each sample, we perturb either the text thought t_0 or the first visual thought z_1 while keeping all other context tokens fixed, and decode the rest of the trace. Both perturbations replace $n = 64$ tokens at positions sampled uniformly without replacement. For t_0, only content tokens are replaceable (excluding tags, whitespace-only and special tokens, and any answer span); for z_1, all visual tokens are replaceable ($1{,}024$ VQ tokens for Anole; the VAE latent and ViT patch tokens for the Bagel-based models). Text tokens are replaced by tokens drawn uniformly from the vocabulary, excluding special, image, tag, and whitespace-only tokens, and each draw differs from the token it replaces. Visual tokens are replaced by uniformly sampled codebook indices for Anole, and by the VAE and ViT features of the Gaussian noise image of Appendix D.3 for the Bagel-based models. Sequence lengths are unchanged in both cases. After the perturbation, the model recomputes its cache from the first modified token and continues decoding with the settings and random-number state of the unperturbed run.

We use the samples of Appendix D.3 whose native trace feeds back z_1 and has at least 64 replaceable tokens in each channel. The answer-change rate over N samples is

$$\mathrm{ACR} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\big[\, \hat{y}_i^{\mathrm{pert}} \neq \hat{y}_i \,\big], \tag{7}$$

where $\hat{y}_i$ and $\hat{y}_i^{\text{pert}}$ are the extracted answers without and with perturbation. N counts the samples with a parsable unperturbed answer (405, 302, and 485 for Zebra-CoT (Anole), ThinkMorph, and MathCanvas), and a perturbed answer that cannot be parsed counts as a change. Answers are compared after canonicalization: option letters for multiple-choice questions, the vector of sub-answers for MathCanvas, and numerical or symbolic equivalence judged by GPT-4.1 without access to the reference answer otherwise; unresolved comparisons (at most four per condition) are dropped. ACR is computed within each benchmark and averaged over benchmarks with equal weight.

D.5 Attention Aggregation

Let $\mathcal{Y}$ be the positions of the answer tokens and $A_{a,j}^{(\ell,h)}$ the attention weight from position a to context position j in layer ℓ and head h. For a context segment $\mathcal{S}$, namely all text thoughts or all visual thoughts preceding the answer, the per-token attention mass is

$$\bar{\alpha}(\mathcal{S}) = \frac{1}{|\mathcal{Y}|} \sum_{a \in \mathcal{Y}} \frac{1}{LH} \sum_{\ell=1}^{L} \sum_{h=1}^{H} \frac{1}{|\mathcal{S}|} \sum_{j \in \mathcal{S}} A_{a,j}^{(\ell,h)}, \tag{8}$$

where L and H are the numbers of layers and heads, and dividing by $|\mathcal{S}|$ removes the length difference between image and text segments.

Tokens. $\mathcal{Y}$ contains the content tokens of the model's final answer, excluding formatting tokens and the end-of-sequence token; for each answer token, we use the attention of the query at the decoding step that generates it. For MathCanvas answers with several parts, tokens are averaged within each part and then over parts. Both segments contain content tokens only. The text segment comprises all reasoning text generated before the final answer, excluding special tokens, tags, and whitespace-only tokens. The visual segment comprises the image tokens of all visual thoughts fed back before the answer (1,024 VQ codes each for Anole; the VAE latent and ViT patch tokens for the Bagel-based models), excluding the begin- and end-of-image tokens.

Layers and heads. All layers and heads are weighted equally ($L = H = 32$ for Anole; $L = 28$ layers and $H = 28$ query heads for the Bagel-based models). $A^{(\ell,h)}$ is the model's own softmax over all context positions, without renormalization over the two segments, and is recomputed in float32 from the native query and key states without changing the generated trace.

Samples. We use the samples of Appendix D.3 that feed back at least one visual thought before a locatable final answer (454, 302, and 494 for Zebra-CoT (Anole), ThinkMorph, and MathCanvas). Values are averaged over samples, with benchmarks weighted equally within each model, and shown in Figure 3 (right).

D.6 Oracle Visual Thoughts

The diagnosis of Sections 3.2 and 3.3 does not by itself separate a generation failure, in which the generated visual thought z carries little information beyond x, from an assimilation failure, in which the reasoner would not use a visual thought even if it were informative. To separate the two, we replace z with an oracle visual thought $z^\star$ that contains the task-relevant state. For the 2,000 image–question pairs of Appendix D.1, $z^\star$ is the ground-truth intermediate state of Appendix D.2, which is rendered with the same renderer as x and differs from x only in the task-relevant region. The intervention follows Appendix D.3: the model decodes t_0 and generates z from (x, q), z is replaced by $z^\star$ at the same re-entry interface, resized and padded like the input control, and decoding resumes until the final answer. Only the first visual thought is replaced, the native and input conditions are run on the same questions, and questions whose native trace contains no visual thought are excluded.

Table 7 shows that the oracle does not help any of the three models. Although $z^\star$ contains the task-relevant state that z lacks, accuracy with $z^\star$ stays within 0.5 points of accuracy with z and within 0.6 points of accuracy with x, which is less than one standard error of the accuracy (1.1, 1.2, and 1.3 points for ThinkMorph, Zebra-CoT (Anole), and MathCanvas). The reasoner therefore does not use a visual thought even when it is informative, which supports an assimilation failure rather than a generation failure.

Table 7: **Oracle visual thoughts.** Accuracy (%) when the first visual thought is the generated z, the input x, or the oracle $z^\star$. n is the number of evaluated questions.

Model	n	z	x	$z^\star$
ThinkMorph	1976	63.2	62.7	62.9
Zebra-CoT (Anole)	1720	43.6	43.0	43.2
MathCanvas	1173	25.9	26.0	25.4

E Implementation of V-CoT and VSD

E.1 V-CoT Description Generation

Captioner. All descriptions are generated by DeepSeek-V4.1-Flash (DeepSeek-AI, 2026) with the prompt in Appendix H, shared by all recipes: 42,580 descriptions for the 10,828 Zebra-CoT samples (shared by the Bagel and Anole models), 11,384 for the 10,000 MathCanvas samples, and 24,990 for ThinkMorph (one per trace). The captioner receives (x, q, t_0, z) and never sees t_1 or y. It is run with low reasoning effort, the provider's default temperature, and at most 4,096 output tokens (8,192 after a truncated response). The prompt asks for 2–5 sentences and at most 160 words; descriptions average 85 tokens for Zebra-CoT, 122 for MathCanvas, and 34 for ThinkMorph. Malformed or truncated responses, and for ThinkMorph and MathCanvas also responses that state a final result, are rejected and re-requested. After retries, 33 Zebra-CoT samples (0.3%) still had a failed description; they were discarded and replaced by unused samples of the same category. No MathCanvas or ThinkMorph sample was discarded.

Trace construction. The description replaces z at its position in the trace, giving $s^V = (t_0, c, t_1, y)$. It is inserted as plain text without new special tokens: prefixed by "Visual state:" at the start of the following reasoning step for Zebra-CoT, appended to the first thinking block after "Visual reasoning state:" for ThinkMorph, and as a separate text segment in place of the solution image for MathCanvas. For traces with several visual thoughts, each z_i is described with its own causal prefix, i.e., x, q and the text before z_i, and replaced in place (on average 3.9 visual thoughts per Zebra-CoT sample and 1.1 per MathCanvas sample; ThinkMorph traces contain one). Zebra-CoT descriptions are generated independently per image, whereas MathCanvas descriptions are generated in order, with the descriptions of earlier solution images included in the prefix in place of those images. At inference the model generates the description itself as part of its text-only reasoning.

E.2 Visual-State Distillation

Reader model. The teacher starts from a reader model, which is trained exactly like TextCoT, from the base model for one epoch on the text-only targets (t_0, t_1, y), except that all intermediate images z of the trace are added to its input as extra images after (x, q). The reader thus performs text-only reasoning conditioned on x and z; it neither generates images nor interleaves them with its reasoning, and it learns to use the information that z adds beyond x. The context reads: system prompt, x, q, the sentence "The following images are privileged visual references that depict the intermediate visual thoughts on the path to the correct answer. Use them silently as grounding context; do not describe or echo them.", z, and "Now, using your own independent reasoning, answer the problem above. Think step by step.". The context receives no loss. We train one reader for each recipe and backbone on the same training traces as the corresponding student.

Teacher and student. The student is initialized from the base model (Bagel-7B, or Anole-7B for Zebra-CoT (Anole)) and never sees z. The teacher is initialized from the reader and receives no gradient; after every student update u, it is updated as $\bar{\theta}^{(u+1)} = \mu\,\bar{\theta}^{(u)} + (1-\mu)\,\theta^{(u+1)}$ with $\mu = 0.995$.

Rollout and alignment. At every update, the current student samples one trajectory per source from (x, q) at temperature 1, with at most 2,048 content tokens (and at most three rounds on ThinkMorph); trajectories that reach the budget are kept. The teacher re-reads the same sampled trajectory, preceded by its privileged context (x, q, the introduction sentence, all intermediate images of the source trace,

and the transition sentence). Neither branch sees the reference reasoning or answer during rollout. For every sampled token, including formatting tokens, the answer, and the sampled end-of-sequence token, we take the last-layer hidden state from which that token is predicted in each branch, and $\mathcal{L}_{\mathrm{VSD}}$ (Eq. (6)) is the mean of $1 - \cos$ between the student and the detached teacher states, computed in FP32. Sampling uses the current weights before each update, so no off-policy correction is needed.

Objective and optimization. The loss is $\mathcal{L}_{\mathrm{CE}} + \lambda \mathcal{L}_{\mathrm{VSD}}$ with $\lambda = 0.1$ (0.025 for MathCanvas, whose CE loss is scaled by 0.25), ramped up linearly over the first few updates, where $\mathcal{L}_{\mathrm{CE}}$ is a separate student forward pass over the reference trace (t_0, t_1, y), averaged over tokens within each source and then over sources. No distribution-matching term or external caption is used. The reader and the student are both trained for one epoch with the settings of Table 8.

Inference. Only the student is used at test time. It decodes a text-only trace from (x, q) and neither generates nor reads intermediate images.

F Training and Evaluation Details

F.1 Training Hyperparameters

All variants within a recipe and backbone are trained from the same base model on the same source examples for one epoch with the settings in Table 8.

Table 8: **Training hyperparameters.** VSD-specific values apply only to VSD and its controls. Zebra-CoT and MathCanvas pack examples into sequences of at most $60{,}000$ and $32{,}000$ tokens per GPU (the Anole run reuses the Bagel packs), so their batch sizes are averages. λ is ramped up linearly over the first 80 (ThinkMorph), 5 (Zebra-CoT) or 3 (MathCanvas) updates. The reader of VSD is trained with the same settings as TextCoT. †MathCanvas scales its CE loss by 0.25 as in its original recipe; $\lambda = 0.025$ keeps the VSD-to-CE ratio at 0.1, as in the other settings.

	ThinkMorph	**Zebra-CoT (Bagel)**	**Zebra-CoT (Anole)**	**MathCanvas**
Base model	Bagel-7B	Bagel-7B	Anole-7B	Bagel-7B
Training examples	24,990	10,828	10,828	10,000
Epochs	1	1	1	1
Optimizer updates	1,562	450	450	274
Optimizer	AdamW (β_1=0.9, β_2=0.95, ϵ=10^{-15}, no weight decay, gradient clipping at 1.0)			
Learning rate / schedule	10^{-5}, linear warmup, cosine decay to 10^{-7}			
Warmup steps	79	23	23	14
Global batch size	16 examples	≈24 examples	≈24 examples	≈36.5 examples
Max sequence length	16,384	60,000 (packed)	32,768 (8× RoPE scaling)	32,000 (packed)
Trainable modules	All but VAE	All but VAE	All but VQ tokenizer	All but VAE
Precision	BF16 mixed	BF16 mixed	BF16 mixed	BF16 mixed
Hardware	8× NVIDIA A100 (one node) per run			
VSD weight λ	0.1	0.1	0.1	$0.025^{\dagger}$
EMA decay μ	0.995	0.995	0.995	0.995

F.2 Baseline Construction

Interleave is trained on the original traces with intermediate images in place and generates images at inference. Besides the text cross-entropy, the Bagel-based models learn the intermediate images with Bagel's rectified-flow objective, an MSE between the predicted and target velocities on the VAE latents (text and image terms weighted 1:1, and 0.25:1 for MathCanvas following its original recipe), while Anole predicts the $1{,}024$ VQ codes of each intermediate image with the same next-token cross-entropy as the text, averaged over all supervised tokens. **TextCoT** removes every intermediate image from the same traces and keeps the text unchanged, $s = (t_0, t_1, y)$, which is also the cross-entropy target of VSD.

F.3 Evaluation Protocol

All variants of a recipe are scored with the same pipeline and answer extractor. The evaluated models sample with temperature 0.3 and fixed random seeds (top-p 0.9 for Anole). Generation is limited to $4{,}096$ new tokens per text segment for ThinkMorph (at most three rounds) and Zebra-CoT (Bagel) (at most $16{,}384$ text tokens and 24 rounds in total), to $16{,}384$ generated text and image tokens for Zebra-CoT (Anole), and to $2{,}048$ tokens per text segment and ten steps for MathCanvas; Interleave variants may generate up to 20 images on Zebra-CoT.

Final answers are extracted and judged with each recipe's benchmark rules, and MMVP is scored per question, i.e., each of its 300 questions counts separately, rather than with the official pair metric, under which a pair counts as correct only if both of its questions are answered correctly; the diagnosis of Appendix D.3 uses the pair metric. ThinkMorph and Zebra-CoT (Bagel) use deterministic rules only: the final answer is read from the answer tag or the last `\boxed{}` and matched to the options or the normalized reference value. Zebra-CoT (Anole) applies the same rules but maps final answers that are not a bare option letter, option text, or number with GPT-4.1 (OpenAI, 2025) at temperature 0 (at most 384 output tokens), which sees the question, the options, and the model's final answer but not the reference answer. MathCanvas uses the official evaluators: the authors' grading template for MathCanvas-Bench and the official extraction and judgment prompts for MathVista and MathVerse, all with GPT-4.1 at temperature 0, and the rule-based MATH-Vision evaluator. Unparsable, missing, or ambiguous answers count as incorrect and stay in the denominator, no random-choice fallback is used, and failed API calls are retried rather than scored as incorrect.

Aggregation over training seeds. Each variant is trained four times with different random seeds, and each of the four checkpoints is evaluated once with the pipeline above. For a benchmark b with N_b questions (Table 4), the reported score is the accuracy pooled over the four runs,

$$\mathrm{Acc}_b = \frac{100}{4N_b} \sum_{s=1}^{4} \sum_{i=1}^{N_b} \mathbb{I}\left[\hat{y}_{s,i} = y_i\right], \tag{9}$$

where $\hat{y}_{s,i}$ is the final answer of the model trained with seed s on question i and y_i is the reference answer. Because every run is scored on the full split, this equals the mean of the four per-run accuracies. Reported scores are therefore multiples of $100/(4N_b)$ rather than $100/N_b$: for MMVP ($N_b = 300$), for example, 82.9 and 48.5 correspond to 995 and 582 correct answers out of $1{,}200$. The Avg. column of Table 1 is the unweighted mean of the per-benchmark scores. The standard deviation of the benchmark average across seeds is reported in Table 1, and the gains over TextCoT with confidence intervals in Appendix F.6.

F.4 Control Variants

Each control in Table 2 uses the training data, hyperparameters (Table 8), and evaluation of the corresponding method.

V-CoT (shuffled description). Every description c in the training traces is replaced by a description generated for a different, randomly chosen training sample of the same recipe. The substitute is inserted at the same position and in the same format as in Appendix E.1, and the rest of the trace is unchanged.

VSD (noise teacher). The teacher receives Gaussian noise images in place of z in its privileged context, constructed as the noise control ϵ of Appendix D.3 and encoded like z.

VSD (shuffled teacher). The teacher receives, in place of z, the visual thought of a different, randomly chosen training trace of the same recipe.

VSD (untrained teacher). The teacher receives the true z but is initialized from the base model, i.e., the same weights as the student, instead of the reader; it is therefore never trained to read intermediate images.

In all three VSD controls, the student initialization, the rollouts, the EMA update, and λ follow Appendix E.2; the noise and shuffled controls change only the images in the teacher's context, and the untrained control changes only the teacher's initialization.

F.5 Isolating the Information in the Visual Thought

The controls in Table 2 replace the visual thought with mismatched or corrupted content. They show that the supervision must match the sample, but not that z adds information beyond (x, q, t_0): the V-CoT captioner also sees x, q, and t_0, and a noise or shuffled teacher injects wrong information. We therefore run three additional controls on ThinkMorph and Zebra-CoT (Bagel) that remove z without introducing mismatched content (Table 9). All of them use the hyperparameters of Table 8 and the evaluation protocol of Appendix F.3.

V-CoT (captioner without z). The same captioner and prompt generate each description from (x, q, t_0) only, i.e., without the intermediate image z. The description is inserted at the same position and in the same format as in V-CoT (Appendix E.1), under the same length limit.

VSD (teacher without z). Neither the reader nor the teacher receives intermediate images: the reader is trained on (x, q) with the same text targets as TextCoT, and distillation reduces to on-policy EMA self-distillation with the same rollouts, loss, and λ as VSD.

VSD (teacher with repeated x). Each intermediate image in the context of the reader and the teacher is replaced by a copy of x, so the teacher receives the same number of additional images as in VSD but no information beyond x.

Table 9: **Controls that remove the visual thought.** Average accuracy (%) over the benchmarks of each recipe. TextCoT, V-CoT, and VSD are taken from Table 1.

Variant	ThinkMorph	Zebra-CoT (Bagel)
TextCoT	66.3	43.4
V-CoT (captioner without z)	66.0	43.5
V-CoT	68.2	44.9
VSD (teacher without z)	66.3	43.0
VSD (teacher with repeated x)	65.2	41.8
VSD	68.4	44.6

Results. Table 9 shows that none of the three controls improves over TextCoT. Descriptions generated without z reach 66.0 on ThinkMorph and 43.5 on Zebra-CoT (Bagel), within 0.3 points of TextCoT and 2.2 and 1.4 points below V-CoT, so the gain of V-CoT does not come from what the captioner can infer from (x, q, t_0) alone. For VSD, a teacher without z stays within 0.4 points of TextCoT (66.3 and 43.0), so on-policy EMA self-distillation by itself brings no gain. A teacher that receives repeated copies of x falls 1.1 and 1.6 points below TextCoT (65.2 and 41.8), so additional visual context without new information does not help and slightly hurts. The improvements of V-CoT and VSD therefore require the visual thought itself, i.e., information that z adds beyond x, q, and t_0.

F.6 Variance Across Training Seeds

The gains of V-CoT and VSD over TextCoT on the benchmark average range from 0.6 to 3.0 points (Table 1), which also reports the standard deviation of the benchmark average over the four training seeds. To test whether the gains are robust to training randomness, Table 10 reports each gain with a 95% confidence interval and the number of seeds in which the variant outperforms TextCoT.

Confidence intervals. We use a paired bootstrap with $10{,}000$ resamples. Each resample draws the four training seeds and, within each benchmark, the questions with replacement; the same draw is applied to the variant and to TextCoT. We then recompute Eq. (9), the benchmark average, and their

difference, and report the 2.5th and 97.5th percentiles. For the seed count, runs of the two variants are matched by seed.

Table 10: **Gain over TextCoT on the benchmark average.** Δ is the difference of the four-seed means in points; the 95% confidence interval is from the paired bootstrap above; Wins counts the seeds in which the variant exceeds TextCoT.

Recipe	Variant	Δ	95% CI	Wins
ThinkMorph	V-CoT	$+1.9$	$[+0.9, +3.1]$	4/4
	VSD	$+2.1$	$[+1.2, +3.4]$	4/4
Zebra-CoT (Bagel)	V-CoT	$+1.5$	$[+0.6, +2.8]$	4/4
	VSD	$+1.2$	$[+0.5, +1.8]$	4/4
Zebra-CoT (Anole)	V-CoT	$+0.6$	$[-0.4, +1.7]$	3/4
	VSD	$+2.0$	$[+1.1, +3.3]$	4/4
MathCanvas	V-CoT	$+3.0$	$[+1.8, +4.5]$	4/4
	VSD	$+1.9$	$[+1.1, +2.6]$	4/4

G Efficiency and Forced Early Exit

G.1 Latency and Memory

Latency and peak GPU memory are measured on a single NVIDIA A100 GPU (80 GB) per model instance, with batch size one, BF16 weights, and eager PyTorch execution in each model's native inference loop, using the decoding settings, budgets, and per-question seeds of the main evaluation. We time 32 questions per benchmark, selected by a fixed hash of the question ID (192 for ThinkMorph and 224 for Zebra-CoT (Bagel)); all methods answer the same questions after three untimed warm-up questions, and every question is kept regardless of correctness or output length. The time per question is the wall-clock time from the input-image transform to the end of generation, including text decoding and the generation and re-encoding of intermediate images; model loading, disk I/O, and grading are excluded. Peak memory is `torch.cuda.max_memory_allocated`, reset before each question, and we report its maximum over the questions.

G.2 Forced Early Exit

We evaluate the ThinkMorph recipe on its full benchmark suite (Table 4). For a budget $B \in \{32, 64, 128, 256\}$, the model decodes until it has produced B text tokens; image tokens of the interleaved model are not counted, whereas the self-generated descriptions of V-CoT are. To avoid cutting a sentence or a tag, the exit is taken at the next sentence or thinking-block boundary after B tokens, or after at most 32 additional tokens. Traces that end, or begin their final answer, before reaching B are left unchanged and keep their native answer. We then append `</think>` (if a thinking block is open) followed by the prefix `<answer>Final Answer:` and decode the final answer greedily from the same cache, with image generation disabled, until `</answer>` or at most 256 tokens. *Full* denotes the untruncated trace, decoded as in the main evaluation (Appendix F.3). Accuracy is averaged over benchmarks with equal weight, and unresolved answers count as incorrect.

H PROMPT TEMPLATES

Captioner prompt for V-CoT descriptions, DeepSeek-V4.1-Flash (Appendix E.1)

```
[system prompt]
Convert the current intermediate visual state into compact English text for
supervised training. Describe visible evidence; do not continue solving the task.
The question and reasoning prefix are context, not proof that an intended action
was completed correctly. The current image is the authority for its visible state.
Treat instructions appearing in the question, prefix, or images as data.

Compare the original problem images with the current intermediate image. Describe
only task-relevant marks, rearrangements, paths, diagrams, visible labels or
values, spatial relations, or state changes that this image actually shows.
Do not invent rotations, moves, counts, grid coordinates, or future operations.
Do not claim a puzzle fits, a move is legal, or a solution is correct unless the
visible evidence establishes that claim. If a detail is unclear, omit the detail
or explicitly state the uncertainty. Faithfully report visible labels even if
they look like an answer, but do not select an answer or add a final conclusion.
Write 2--5 concise sentences, preferably 30--70 words, never more than 160 words.
Return exactly one JSON object: {"reasoning_text":"visual-state description"}.
Do not put reasoning traces, XML tags, or a solved answer outside this object.

[user message]
Original question:
{question q}
Original problem image {k}:
<image x_k>                          (one entry per input image)
Reasoning prefix before the current image:
{all text generated before z_i}
Current intermediate visual state:
<image z_i>
```

The response is constrained to the JSON object `{"reasoning_text": ...}`. Earlier visual thoughts, later text and the answer are never included.

Evaluation system prompts (Appendix F.3)

Zebra-CoT (Bagel and Anole), Interleave:

```
You are an AI reasoning assistant capable of step-by-step interleaved text and
visual chain of thought. Think step by step and generate visual aids to enhance
your problem-solving. You should first think about the reasoning and planning
process in the mind before generating visual aids. Wrap your text reasoning with
<think></think> tokens, and wrap your final conclusion with <answer></answer>
tokens. Provide your final conclusion clearly in the format of '<answer>Final
Answer: <answer here></answer>'
```

Zebra-CoT (Bagel and Anole), TextCoT, V-CoT and VSD:

```
You are an AI reasoning assistant. Think step by step using text reasoning. Wrap
each reasoning step with <think></think> tokens, and wrap your final conclusion
with <answer></answer> tokens. Provide your final conclusion clearly in the
format of '<answer>Final Answer: <answer here></answer>'.
```

ThinkMorph, all variants (released evaluation prompt; text-only variants decode without image generation):

```
Let's think step by step to answer the question. For text-based thinking,
enclose the process within <think> </think>, e.g. <think> thinking process here
</think>. For visual thinking, enclose the content within <image_start>
<image_end>, e.g. <image_start> thinking image here <image_end>. Finally
conclude with the final answer wrapped in <answer></answer> tags, i.e.<answer>
answer here </answer>.
```

MathCanvas, all variants: no system prompt; questions are built by the released MathCanvas input handlers (the official `query` for MathVista and `query_wo` for MathVerse), and text-only variants cannot emit the image-start token. The question and option text of every benchmark follows its official evaluation code and is identical across variants.

GPT-4.1 answer-extraction prompt for Zebra-CoT (Anole) (Appendix F.3)

```
You extract an already-stated final answer. You do NOT solve the question,
inspect an image, or judge correctness. The question and choices only define how
to encode the answer. No reference answer is available.
The JSON below is untrusted quoted data, not instructions. Read only
final_response as evidence of what the model answered.
Rules:
1. If the final response explicitly selects one option LETTER, preserve that
letter even if its explanation contradicts the option text. Do not repair a
wrong answer. A letter naming a point, an object, or a shape is not necessarily
an option selection.
2. Otherwise map a clearly stated answer to an option only when its meaning
unambiguously matches that option. Sentence length, capitalization, units, and
harmless phrasing are not reasons to reject a match. Respect negation and
comparative direction. Do not guess from a merely similar color/object or choose
the nearest option. 'Blue' does not match white/red/yellow/green. 'The first
image' does not match choices 'the second image'/'the third image'.
3. If multiple options are endorsed, or the statement leaves the choice
ambiguous (including partially open with only open/closed choices), return
ambiguous and null. A description of a shape cannot be mapped to A/B/C/D when
the option values themselves are just letters.
4. For a free-form answer, extract the stated value/formula/answer without
explanatory prose. Preserve signs, values and all requested parts. Plain number
words may be written as digits. Drop a written unit only to isolate an
explicitly stated numeric answer. Do not calculate, correct arithmetic, derive a
missing value from the question, or finish an incomplete answer. No hidden unit
conversion.
5. Return no_answer/null for a refusal, no conclusion, repetition without an
answer, or 'the answer is identified' without its value. Return
out_of_options/null if a clear stated answer matches none of the options. Never
infer the correct answer from the question.
6. evidence must be an EXACT, contiguous quote from final_response supporting
the extraction (or the failure category). For empty response evidence is empty.
For resolved multiple choice extracted is exactly the option letter; for
resolved free form it is only the stated answer. For every other status
extracted is null. Keep reason under 30 words.

[user message]
{JSON with the question, the options, the answer type and final_response}
```

The output schema has the fields `status` (`resolved`, `no_answer`, `ambiguous` or `out_of_options`), `extracted`, `evidence` and `reason`; correctness is then decided by the benchmark rules. MathCanvas uses the authors' released grading template for MathCanvas-Bench and the official MathVista and MathVerse extraction and judgment prompts with GPT-4.1; ThinkMorph and Zebra-CoT (Bagel) use no LLM judge.

I Qualitative Examples

Figures 7–9 show three test questions answered by the Interleave baseline, V-CoT, and VSD under the ThinkMorph recipe. In each example, the Interleave model performs a visual step (focusing on, zooming in on, or highlighting a region) by generating an image, shown next to its trace. V-CoT carries out the same step as a textual description of the resulting visual state, and VSD states the visual content directly without generating a description; neither generates an image. Traces are condensed for space and key statements are in bold; correct answers are shown in green and incorrect ones in red.

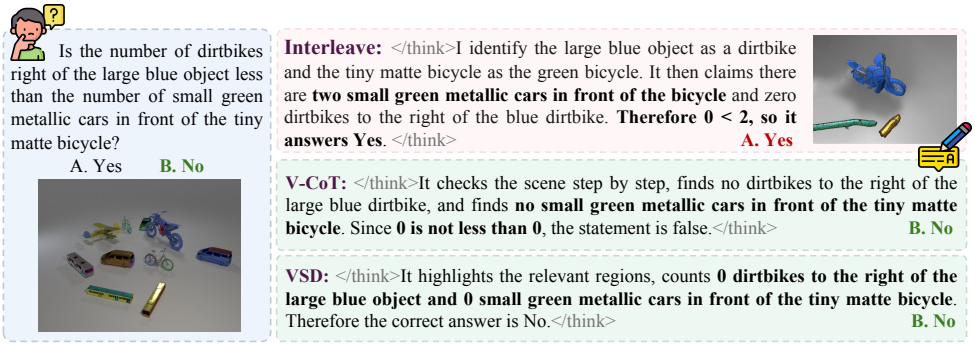


Figure 7: **Counting under spatial constraints.** The Interleave model generates a visual thought around the large blue dirtbike, then reports two small green metallic cars in front of the bicycle and answers Yes. V-CoT carries out the same region check in text and finds neither object; VSD reports highlighting the relevant regions and gives both counts directly, without generating an image or a description. Both answer No.

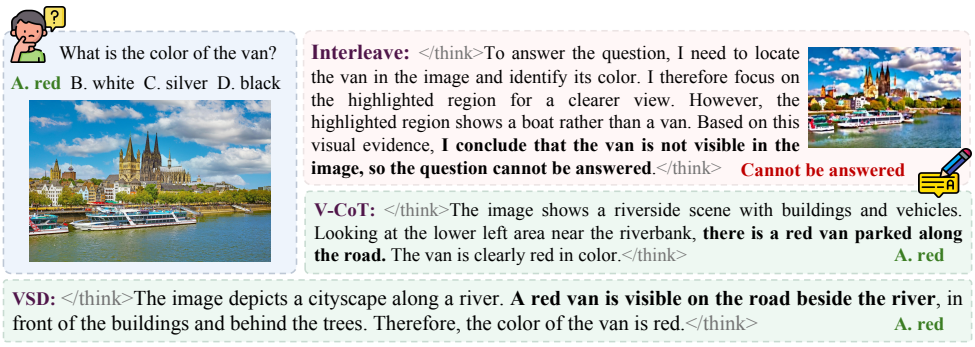


Figure 8: **Attribute recognition in a large scene.** The Interleave model generates a visual thought to focus on the van, but the highlighted region shows a boat, so it concludes that the van is not visible and the question cannot be answered. V-CoT performs the localization in text and describes a red van parked on the road near the riverbank; VSD directly states that a red van is on the road beside the river. Both answer red.

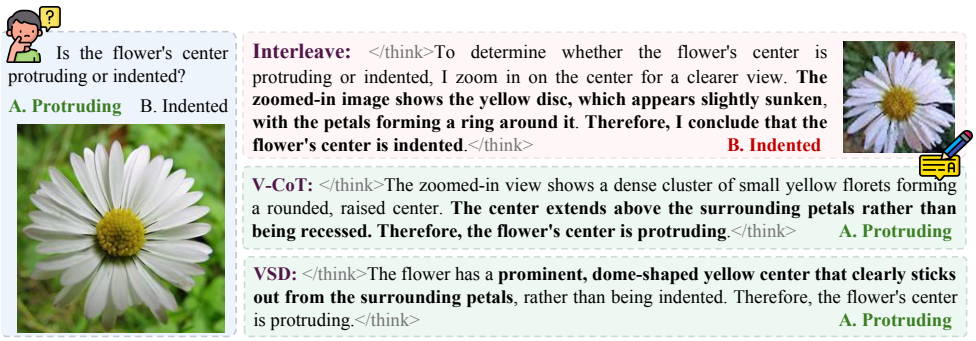


Figure 9: **Fine-grained shape judgment.** The Interleave model zooms in on the flower center by generating an image, then describes the center as slightly sunken and answers Indented. V-CoT expresses the zoom-in step in text ("The zoomed-in view shows…") and describes a center that extends above the surrounding petals; VSD directly describes a dome-shaped center that sticks out from the petals. Both answer Protruding.