ThinkMorph-SSL: Scaling Interleaved Multimodal Reasoning with Pretext Tasks

Jiawei Gu[1]* **Linjie Li**[1]* **Yiming Liu**[2]* **Yunzhuo Hao**[3]* **Huichen Wang**[1]*
Guanzheng Chen[4]* **Zhichao Peng**[5] **Luxin Xu**[6] **Xinyu Zhang**[6] **Li Luo**[7]
Disen Lan[6] **Zican Hu**[8] **Mingyang Song**[6] **David Valente**[4] **Alex Jinpeng Wang**[5]
Yafu Li[9] **Ganqu Cui**[2] **Zhengyuan Yang**[10] **Qizhe Xie**[4]†
Yejin Choi[11]† **Ranjay Krishna**[1]† **Yu Cheng**[9]†

[1]University of Washington [2]Tsinghua University [3]Zhejiang University
[4]National University of Singapore [5]Central South University [6]Fudan University
[7]The University of Hong Kong [8]Nanjing University [9]The Chinese University of Hong Kong
[10]University of Rochester [11]Stanford University

* Project lead * Equal contribution † Equal advising

https://thinkmorph.github.io/ssl/

Abstract

People reason with pictures as well as words. Interleaved multimodal reasoning gives models a similar way to solve problems by creating intermediate images and using them to guide further reasoning. Training models with this ability usually relies on task-specific demonstrations, which are difficult to collect. We introduce ThinkMorph-SSL, a two-stage training recipe that uses images themselves as a source of supervision. The first stage, *pretext reasoning*, converts visual pretext tasks, such as image colorization and denoising, into *instruction-free* interleaved traces. Models describe a corrupted or transformed image, generate a restored or canonical image, and interpret the result. The second stage uses automatically constructed demonstrations for question-driven instruction tuning. At the same instruction-tuning budget, models trained with pretext reasoning outperform those trained with instruction tuning alone. The full recipe improves over the base model by an average of 11.33 accuracy points across twelve benchmarks. These results show that supervision derived from images can build the visual skills needed for interleaved reasoning and improve learning from task-specific demonstrations.

1 Introduction

Sketching and imagery are basic forms of thinking, used for reasoning [27], ideation [82], and problem-solving [81]. Unlike language alone, they can express visuo-spatial structure directly [114]. An auxiliary line can simplify a geometry proof [45]. A spatial puzzle can become easier after mentally rotating an object [35, 43], and diagrams help tell stories and communicate scientific ideas. These visual representations are working states, not merely illustrations. They reorganize the problem and make later inference easier [42]. As multimodal language models mature, we increasingly ask them to solve math [24, 54], scientific [33, 91], and multimodal [25, 94, 110] tasks where people naturally rely on such intermediate visual reasoning.

This motivates *interleaved reasoning*: creating intermediate visual representations and using them to guide subsequent language-based inference. Unified models [17, 30, 61] support this exchange natively. However, coordinating visual and language steps requires training traces where intermediate images support the question being answered. Constructing these traces involves task-specific pipelines and data review [46, 69]. ThinkMorph, for example, designs interleaved traces around four tasks,

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

each with tailored visual manipulations [30]. These pipelines can generate more examples, but covering new tasks often requires new designs, making interleaved supervision difficult to scale.

Images themselves can supply training targets for interleaved reasoning. In classical self-supervised learning (SSL), pretext tasks such as jigsaw reconstruction [58] and image inpainting [64] use an image's original structure or content as supervision. We introduce ThinkMorph-SSL, a two-stage recipe illustrated in Figure 1. Stage 1, ***pretext reasoning***, converts seven visual pretext tasks into question-free interleaved traces, hereafter called *instruction-free*, that train the model to read a transformed image, generate a restored or canonical visual state, and reason over the result (Pretext-270K). Stage 2 uses instruction tuning on question-driven traces constructed with nine automatic, task-specific pipelines (Instruct-180K).

Our experiments show that instruction-free pretext practice makes subsequent instruction tuning more effective. With the tuning budget held fixed, this practice adds an average of 6.49 points in accuracy on three representative benchmarks, and its benefits generalize across unified backbones. Surprisingly, instruction-free traces still yield slightly better results than explicitly instructed ones, despite the added difficulty of inferring the task itself. Furthermore, the full two-stage recipe outperforms the baseline by 11.33 points on average across twelve benchmarks spanning three levels: *basic visual perception*, *static spatial understanding*, and *mental world modeling*. Improvements extend to nine benchmarks outside the instruction-tuning task families, supporting generalization to new tasks.

These findings suggest that interleaved reasoning can benefit from supervision beyond task-specific demonstrations. This brings the self-supervised learning principle to interleaved multimodal reasoning: images themselves can provide training supervision, reducing the need to identify suitable tasks and design demonstrations for them.

2 A Two-Stage Recipe for Scaling Interleaved Reasoning

The recipe trains a unified multimodal backbone to generate and reuse visual states within a single autoregressive reasoning trace. Its key design is to separate scalable, task-independent supervision from the task-specific kind. Section 2.1 formalizes the trace format, and Section 2.2 describes the two training stages, pretext reasoning and task-specific instruction tuning.

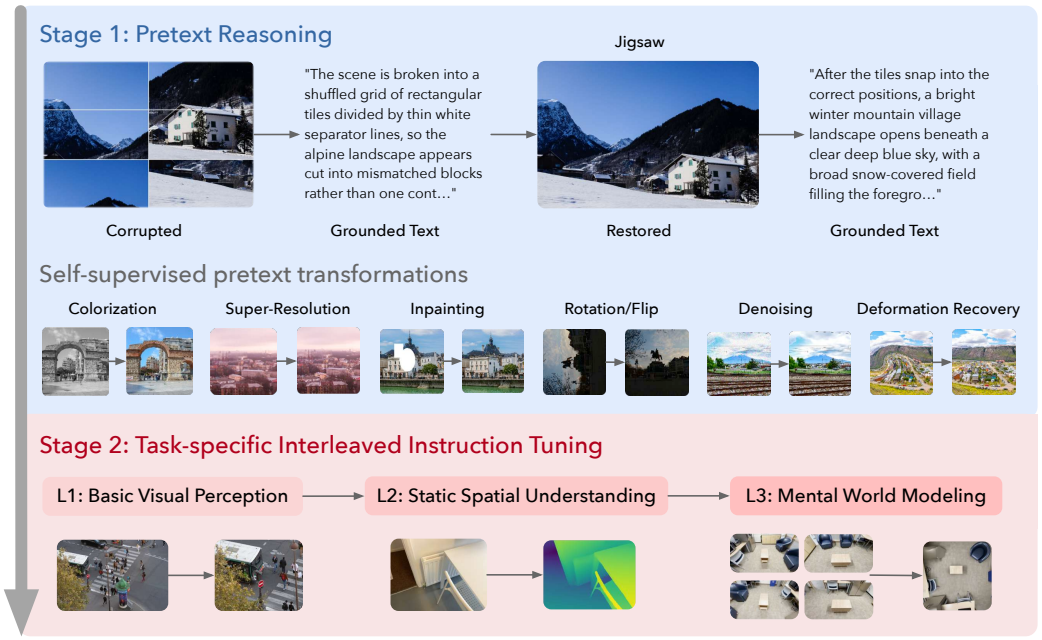


Figure 1: Overview of the two training stages. *Stage 1 (top)* uses seven instruction-free visual pretext tasks to practice producing and reading intermediate images. *Stage 2 (bottom)* uses question-driven instruction tuning to teach the model to use image content to advance reasoning across basic visual perception, static spatial understanding, and mental world modeling.

2.1 Interleaved Reasoning Format

We formulate interleaved multimodal reasoning as autoregressive generation over text and image segments. Given input x, the model generates a trace

$$\mathcal{S} = (S_1, \ldots, S_K), \qquad p_\theta(\mathcal{S} \mid x) = \prod_{k=1}^{K} p_\theta(S_k \mid x, S_{<k}),$$

where each S_k is either a text segment T_k or an image segment I_k. Image segments are enclosed by `<gen_image>` delimiters, allowing the same model to alternate between language and image generation. Unlike text-only chain-of-thought or tool-augmented reasoning, the model itself generates intermediate visual states and conditions later reasoning on them.

2.2 Two Sources of Interleaved Supervision

Both sources of supervision share the same interleaved autoregressive interface (Figure 1). In Stage 1, pretext reasoning builds task-independent traces from unlabeled images; in Stage 2, task-specific instruction tuning uses traces that connect intermediate images to questions and answers. The model thus first learns the mechanics of producing and reading visual states, then how to use them for vision-centric reasoning.

Stage 1: pretext reasoning. This stage constructs *instruction-free* traces from corrupted–clean image pairs (Figure 1, top): the input is a corrupted or transformed image I_c with no instruction, not even a question. The target trace is

$$I_c \to T_0 \to I_0 \to T_1,$$

where T_0 describes the visible evidence, I_0 is the restored or canonical image, and T_1 describes the resulting visual state. This can be factorized as

$$p_\theta(T_0, I_0, T_1 \mid I_c) = p_\theta(T_0 \mid I_c) p_\theta(I_0 \mid I_c, T_0) p_\theta(T_1 \mid I_c, T_0, I_0).$$

We convert classical pretext tasks—jigsaw, colorization, super-resolution, masked inpainting, rotation/flip recovery, denoising, and deformation recovery—into such traces. Unlike edit-style instruction tuning, these traces do not name the operation; the model must infer the transformation from the visual input, generate the intermediate state, and interpret it.

Stage 2: task-specific interleaved instruction tuning. This stage keeps the same interface but changes the goal from restoration to task solving. Each example contains an image, a question, and a target trace with reasoning text, one or more intermediate visual states, and a final answer:

$$(x^{\text{img}}, x^{\text{text}}) \to T_0 \to I_0 \to \cdots \to T_K.$$

Here, the generated image is chosen because it helps answer the question. Instruction tuning therefore teaches when to construct a visual state, what state to construct, and how to use it for the final answer. These tasks progress from inspecting visible evidence to reconstructing unseen scene states. We organize them into three levels, referred to as L1, L2, and L3 throughout the paper:

- **L1 (Basic Visual Perception)** concerns directly visible content, such as objects, attributes, and local evidence.
- **L2 (Static Spatial Understanding)** concerns the spatial organization of the visible scene, including depth, layout, relative position, and counts.
- **L3 (Mental World Modeling)** requires reconstructing scene states beyond the visible input, such as alternate viewpoints and cross-view correspondences.

Training objective. Both stages train the model on the same interleaved trace format, with losses applied only to the target text and image segments. The exact token-level and image-generation losses depend on the unified multimodal backbone.

3 Training Corpus

We curate two datasets. Pretext-270K, used in Stage 1, is generated automatically from raw images with programmatic corruptions and contains instruction-free Text–Image–Text traces. Instruct-180K,

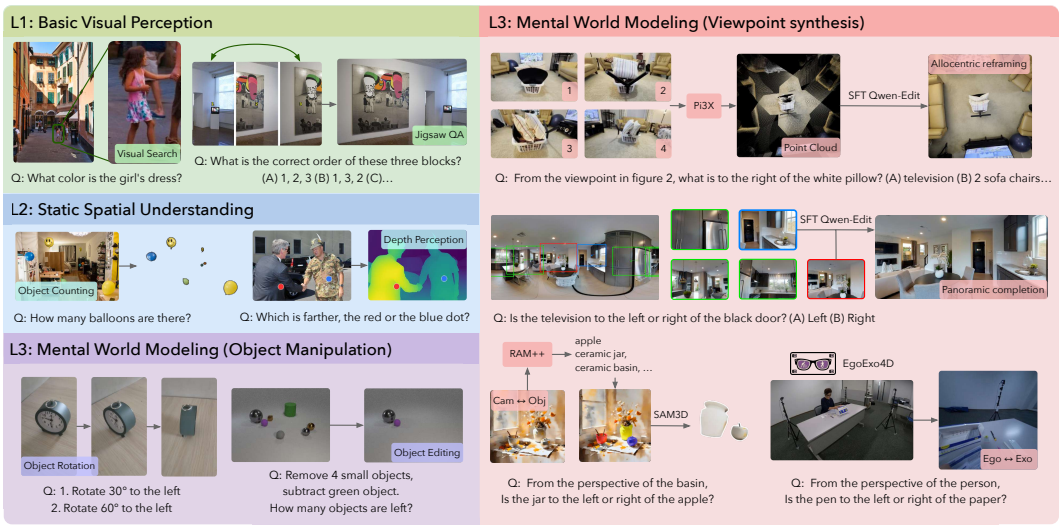


Figure 2: Instruct-180K construction pipelines. The pipelines are grouped into three levels of vision-centric reasoning: basic visual perception, static spatial understanding, and mental world modeling. Each panel shows how one pipeline constructs the intermediate visual state used in an interleaved reasoning example, such as a localized crop, a reordered view, a depth map, an edited object state, or a synthesized viewpoint. Full details are provided in Appendix D.

used in Stage 2, relies on task-specific pipelines to generate questions, intermediate visual states, and answers across basic visual perception, static spatial understanding, and mental world modeling. Both share the same text-image trace interface but differ in supervision, and Figure 8 in the Appendix summarizes their composition.

3.1 Pretext-270K: Instruction-Free Interleaved Reasoning Data

The corpus contains 270,896 traces, each derived from a corrupted–clean image pair (Figure 1, top). It is balanced across seven programmatic transformations, including **(1) Jigsaw** [58]: restore scrambled patches to the original layout; **(2) Colorization** [115]: recover color from grayscale; **(3) Super-resolution** [108]: reconstruct details from low resolution; **(4) Masked inpainting** [64]: complete missing regions; **(5) Rotation and flip recovery** [26]: restore canonical orientation; **(6) Denoising** [83]: remove noise while preserving content; and **(7) Deformation recovery** [18]: undo local geometric warps. Together, these tasks cover complementary forms of visual-state recovery, from appearance restoration to spatial and structural correction. Appendix F provides examples.

Source images. We sample images from eight public datasets spanning natural images, web images, charts, screenshots, documents, indoor scenes, panoramas, and 3D environments: SA-1B [41], PixMo-Cap [16], VSI-590K [103], LLaVA-OneVision-1.5-Instruct-Data [2], DeepVision-103K [73], Matterport3D [9, 23], Realsee3D [48], and 360+x [10]. This diversity keeps the corpus from being tied to a single image domain, and the source-agnostic pipeline accepts new image pools.

Automatic construction pipeline. For each image, we apply one of the seven transformations to create a corrupted input and use the original image as the clean target. We then synthesize the two text segments with a strong multimodal model [59]: first captioning the clean image, then generating T_0 for the corrupted view and T_1 for the restored view using both images and the reference caption.

Importantly, the inputs carry no instruction such as "denoise this image" or "restore the jigsaw." The model must infer the transformation from visual evidence and autoregressive context, which keeps the corpus closer to pretraining than to image-editing instruction tuning.

3.2 Instruct-180K: Task-Specific Interleaved Reasoning Data

Whereas Pretext-270K teaches the mechanics of visual states, this corpus teaches how to use them for downstream tasks. It contains 180,490 traces from **nine construction pipelines**, of which 62,100 target basic visual perception, 15,144 static spatial understanding, and 103,246 mental world

modeling. Each example includes an input image, a question, reasoning text, one or more intermediate or final image segments, and an answer. Figure 8 in the Appendix shows the per-task distribution.

Task coverage. Basic visual perception tasks teach the model to locate and inspect visible evidence, including visual search [94] and Jigsaw QA [90]. Static spatial tasks require organizing observed evidence into spatial structure, including spatial counting [32] and depth reasoning [25, 119] from a single image. Mental world-modeling tasks require constructing new visual states beyond the original input, including bird's-eye view synthesis [107], panorama [9, 10, 23, 48, 104], alternate-view rendering [68], object rotation [109, 112], and ego–exo perspective transfer [29].

Task-specific pipelines. Each task family uses a dedicated pipeline (Figure 2), since supervision comes from different sources. Visual search uses object-dense scenes with target regions highlighted by zooms or boxes. Jigsaw QA reuses a prior pipeline [30], and counting derives gold counts from LVIS instance masks [32]. Depth reasoning uses depth annotations or estimates. Viewpoint-centered tasks rely on geometry, including yaw and field-of-view metadata for panoramas, reconstruction and rendering for bird's-eye views, paired ego–exo frames or 3D lifting for perspective transfer, and calibrated multi-view data for object rotation. Whenever possible, the ground-truth answers are derived from geometry, annotations, or metadata rather than being synthesized by a language model.

Reasoning trace synthesis and quality control. After constructing the input, intermediate visual state, and ground-truth answer, we synthesize traces in a shared interleaved format. Each visual state is placed before the reasoning step that uses it, enforcing the intended causal order of first constructing or inspecting the state and then answering from it. We apply task-specific quality checks, such as yaw/FOV validation for panoramas, coverage checks for bird's-eye-view renders, IoU-based deduplication for segmentation, and multi-model judging for visual search. The shared synthesis prompt further prevents premature answer leakage, requires grounded references to generated images, and disallows describing an intermediate image before it appears.

4 Experiments

4.1 Setup

Training. We build ThinkMorph-SSL on BAGEL-7B-MoT [17] and InternVL-U-4B [77]. Both backbones generate text and images. Stage 1 trains on Pretext-270K, and Stage 2 continues from the Stage-1 checkpoint on Instruct-180K. Both stages use the same system prompt with a `<gen_image>` delimiter. Appendix G gives the full details.

Evaluation. We evaluate on twelve benchmarks at three levels: **L1** basic visual perception (V* [94], HR-4K, HR-8K [86], MME-RW-L [116], RealWorldQA [98]), **L2** static spatial understanding (BLINK [25], CV-Bench [78], SAT [67]), and **L3** mental world modeling (MMSI [104], MindCube [107], OmniSpatial [39], SpatialViz [85]). Only V*, MMSI, and MindCube share a source dataset or task family with Instruct-180K, so the other nine are out-of-domain. Benchmark questions and templates never appear in our training data. We score every model with VLMEvalKit [21] and use GPT-5 [59] as the LLM judge. Appendix G.2 gives the full protocol.

Baselines. We compare against closed-source APIs (GPT-5 [59], Claude Sonnet 4.6 [3], Gemini 2.5 Flash [28]), open-source vision–language models (Qwen2.5-VL-7B/32B [4]), and open-source unified models (Janus-Pro [13], Chameleon [75], SenseNova-U1-8B [61], LongCat-Next-A3B [76], Show-o2-7B [100]). We also run the untrained BAGEL backbone under several inference settings.

4.2 Inference-time Effort Does Not Yield Effective Interleaved Reasoning

A unified model might reason across text and images if it simply tries harder at inference time. We test this with more text effort, then more visual effort, and finally a ground-truth image to separate image quality from image use. The results suggest that training must teach two abilities: producing and reading intermediate images, and using them to advance reasoning.

Table 1: More text reasoning effort at inference time. Accuracy (%) averaged over each level. Parentheses give the change from the row above.

Setting	L1	L2	L3
No-think	50.35	56.17	36.42
Think-4k	51.40 (+1.05)	64.67 (+8.50)	25.40 (−11.02)
Think-16k	53.86 (+2.46)	64.23 (−0.44)	26.83 (+1.43)

Table 2: More visual reasoning effort at inference time. Spatial is a random 500-question subset of Instruct-180K with ground-truth (GT) intermediate images.

Setting	V*	MMSI	Spatial
Text reasoning (16k)	61.26	29.00	47.40
Forced interleaved (16k)	61.26	27.20 (−1.80)	41.20 (−6.20)
+ GT image	–	–	42.00 (+0.80)

More text reasoning adds little. Adding text reasoning with a 4k token budget to direct answers helps L1 and L2 but hurts L3 (Table 1). Many L3 outputs end without a final answer and are scored as wrong. Quadrupling the budget to 16k changes the three averages little, and L3 remains 9.59 points below direct answers. Tables 3 and 4 report all three settings and measure gains against the baseline.

Forced image generation does not help. We next increase visual reasoning effort while keeping the text-token budget at 16k. In Table 2, the model must generate an image and then continue in text. This forced interleaved reasoning ties text reasoning on V* and loses on MMSI. Visual reasoning effort at inference time does not improve on text reasoning.

Even adding a ground-truth image does not help. The previous comparison does not separate image quality from image use. We replace the generated image with a verified ground-truth image on 500 spatial questions from Instruct-180K, as if the model had produced it. Forcing an image costs 6.20 points on these questions, and the ground-truth image wins back only 0.80 points. These results suggest that **following the interleaved format is not enough**. The model must also learn to use what an intermediate image shows to advance its later text reasoning.

4.3 Main Results

We compare its performance with the baseline and other models, examine whether its answers rely on the generated images, and measure the cost of interleaved reasoning at inference time.

Table 3: Level 1 benchmarks (Basic Visual Perception). HR-4K / HR-8K are HRBench at 4K / 8K resolutions; MME-RW-L is the average of the Reasoning and Perception sub-scores. Δ is computed against BAGEL think-4k.

	Level 1: Basic Visual Perception					Avg
Model	V*	HR-4K	HR-8K	MME-RW-L	RealWorldQA	L1
Closed-source						
GPT-5	69.63	67.50	65.50	56.80	73.86	66.66
Claude Sonnet 4.6	66.49	72.50	64.50	46.95	71.50	64.39
Gemini 2.5 Flash	76.96	82.75	78.50	51.02	75.16	72.88
Open-source						
Qwen2.5-VL-7B	71.20	61.38	58.75	43.10	61.96	59.28
Qwen2.5-VL-32B	75.92	66.62	64.38	45.28	63.01	63.04
Unified						
Janus-Pro	38.22	44.88	39.63	23.19	41.83	37.55
Chameleon	28.27	26.25	24.88	24.13	36.21	27.95
SenseNova-U1-8B	61.26	56.38	52.25	39.52	69.67	55.82
LongCat-Next-A3B	76.44	65.38	62.00	40.78	71.76	63.27
Show-o2-7B	42.93	49.75	43.25	41.01	64.05	48.20
BAGEL inference variants						
BAGEL no-think	52.88	46.88	39.50	41.88	70.59	50.35
BAGEL think-4k	58.64	59.38	48.88	25.64	64.44	51.40
BAGEL think-16k	61.26	63.00	50.50	25.64	68.89	53.86
ThinkMorph-SSL (Ours)	68.06	67.00	55.38	45.28	70.46	61.24
Δ baseline	**+9.42**	**+7.62**	**+6.50**	**+19.64**	**+6.02**	**+9.84**

Table 4: Level 2 (Static Spatial Understanding) and Level 3 (Mental World Modeling) benchmarks. CV-Bench is split into CV-2D and CV-3D. Δ is computed against BAGEL think-4k.

	Level 2: Static Spatial Understanding				Level 3: Mental World Modeling				Avg	
Model	BLINK	CV-2D	CV-3D	SAT	MMSI	MindCube	OmniSpat	SpatViz	L2	L3
Closed-source										
GPT-5	67.70	77.68	87.58	81.33	42.50	42.33	57.34	53.47	78.57	48.91
Claude Sonnet 4.6	63.28	69.25	82.33	44.67	32.70	43.18	50.29	44.58	64.88	42.69
Gemini 2.5 Flash	67.23	74.21	80.75	65.33	35.90	42.20	53.62	29.66	71.88	40.35
Open-source										
Qwen2.5-VL-7B	54.87	69.97	64.00	45.33	28.80	30.95	41.88	30.76	58.54	33.10
Qwen2.5-VL-32B	59.55	74.28	83.33	43.33	27.60	33.10	47.75	35.25	65.12	35.93
Unified										
Janus-Pro	38.51	68.74	66.75	22.00	27.80	37.50	28.70	18.64	49.00	28.16
Chameleon	16.52	27.36	47.50	18.67	5.10	28.85	22.90	21.53	27.51	19.60
SenseNova-U1-8B	68.60	81.45	90.33	72.00	38.70	64.13	45.66	31.27	78.10	44.94
LongCat-Next-A3B	56.34	73.17	80.75	27.33	28.80	39.90	40.18	28.56	59.40	34.36
Show-o2-7B	43.35	67.95	74.08	26.00	29.00	40.00	40.51	30.68	52.84	35.05
BAGEL inference variants										
BAGEL no-think	63.91	70.92	43.17	46.67	31.60	34.71	45.79	33.56	56.17	36.42
BAGEL think-4k	53.39	72.81	85.17	47.33	27.40	29.66	31.57	12.97	64.67	25.40
BAGEL think-16k	52.97	74.30	83.00	46.67	29.00	29.66	35.09	13.56	64.23	26.83
ThinkMorph-SSL (Ours)	58.02	74.79	86.75	56.00	34.40	60.38	45.27	33.22	68.89	43.32
Δ baseline	**+4.63**	**+1.98**	**+1.58**	**+8.67**	**+7.00**	**+30.72**	**+13.70**	**+20.25**	**+4.22**	**+17.92**

Gains are consistent across all twelve benchmarks. Tables 3 and 4 show that ThinkMorph-SSL improves on the baseline on every benchmark. The improvement spans visual perception and spatial reasoning. L3 has the largest average gain at 17.92 points, compared with 9.84 on L1 and 4.22 on L2. The larger L3 gain may reflect a division of labor between generation and perception. A unified model can construct the unseen state, then inspect it visually, turning part of mental simulation into a perceptual judgment. The gains extend beyond the training task families. The three in-domain benchmarks gain 15.71 points on average. All **nine out-of-domain benchmarks** also improve, by 9.87 points on average. At the benchmark level, MindCube gains the most at 30.72 points. SpatialViz has the largest gain among out-of-domain benchmarks at 20.25 points. These gains combine pretext reasoning and instruction tuning.

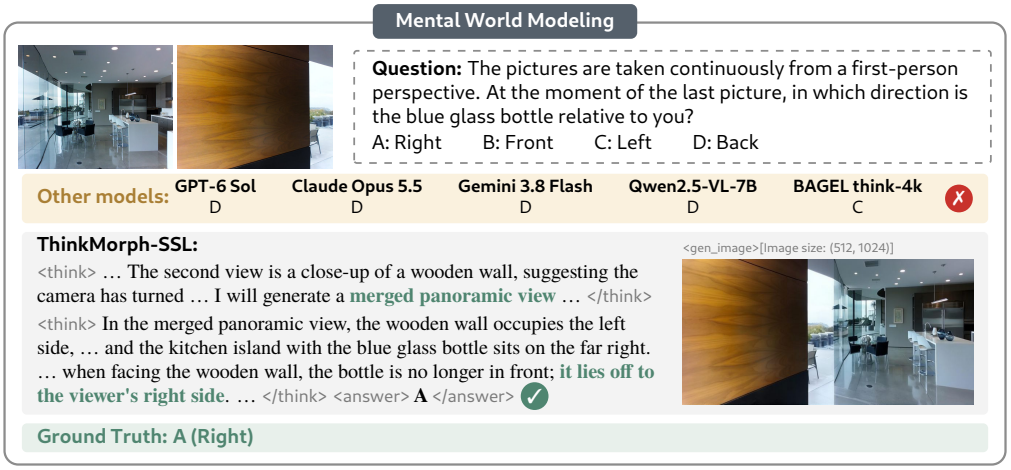


Figure 3: A qualitative example from MMSI.

ThinkMorph-SSL rivals larger models on spatial reasoning. Figure 3 presents a qualitative example from MMSI. The final frame faces a wooden wall, while the bottle appears in the earlier view. A unified backbone, an open-source VLM, and three closed-source models answer left or back. Our model instead merges the views into a panorama and answers right. The panorama places the wall and bottle in one view, making their spatial relation available to the next reasoning step.

Interleaved reasoning here combines both abilities: producing and reading an intermediate image, then using its content to derive the answer.

The advantage extends beyond this example to the spatial benchmark averages in Table 4. Among unified models, our model exceeds four competitors on both spatial levels. The lead also holds against open-source VLMs, including both the 7B and 32B versions of Qwen2.5-VL. The 32B model trails by 7.39 points on mental world modeling. The comparison with closed-source models is mixed. Our model outperforms Claude Sonnet 4.6 on both spatial levels. Its lead on static spatial understanding is 4.01 points. It also exceeds Gemini 2.5 Flash by 2.97 points on mental world modeling. On the in-domain MindCube benchmark, our model exceeds GPT-5 by 18.05 points.

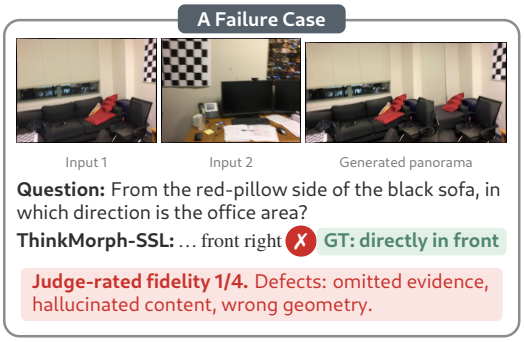


Figure 4: A low-fidelity panorama leads to an error. Key objects are missing or duplicated.

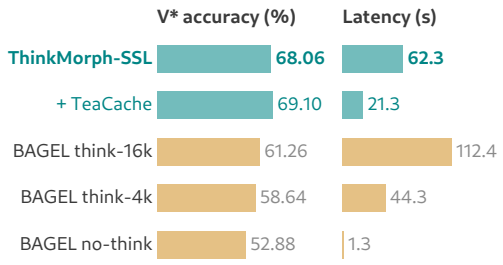


Figure 5: Accuracy and latency on V*. TeaCache reduces latency without lowering accuracy.

The model leverages its own generated images for reasoning. We analyze sampled traces from both splits in Table 5, comparing image use, image defects, and answer accuracy. We first check whether traces with images are more accurate. The model chooses whether to generate an image. Higher accuracy alone cannot show that the image caused the gain. For questions with both image and image-free traces, we compare accuracy between the two groups. Accuracy is 37.2 points higher with images on the training subset and 12.8 on MMSI. We then check whether image defects accompany answer errors. A judge flags critical defects without seeing the final answer. Traces with defective images have higher error rates on both splits. On MMSI, the gap is 20.4 points.

Table 5: Does the image matter? Spatial (train): 150 spatial problems from Instruct-180K. MMSI (test): 150 test problems. Eight traces per problem. Left: within-problem accuracy w/ vs. w/o images. Right: error rates w/ vs. w/o critical defects.

	Accuracy (%)			Error rate (%)	
Split	w/ img	w/o img	Δ	defect	none
Spatial (train)	69.4	32.2	+37.2	40.7	15.4
MMSI (test)	31.3	18.5	+12.8	67.8	47.4

The failure in Figure 4 further illustrates this link between image defects and answer errors. The panorama omits the desk and monitors from Input 2. It also duplicates the sofa, red pillow, and chair. The model answers front right instead of directly in front. The model appears to use the generated scene as evidence, making its answer vulnerable to missing or distorted spatial information.

The cost of interleaved reasoning can be reduced without losing accuracy. More inference-time effort alone did not yield effective interleaved reasoning in Tables 1 and 2. After training, we apply inference-time acceleration to reduce the model's generation cost. Figure 5 compares accuracy and end-to-end latency on V*. The uncached model is more accurate but slower than the baseline. Image generation accounts for nearly all its FLOPs, making it the main optimization target. We apply TeaCache [53] to reuse computation across image-denoising steps. This reduces total FLOPs by **77%** without lowering accuracy. Latency drops by 41.0 seconds to 21.3 seconds, below the baseline's 44.3 seconds. ThinkMorph-SSL can incorporate compatible accelerators for both text and image generation, with TeaCache serving as one initial example.

4.4 Does Pretext Reasoning Improve Interleaved Reasoning?

The main results combine pretext reasoning and instruction tuning. Here, we isolate the contribution of pretext reasoning, then examine the training schedule and the use of instructions in Stage 1.

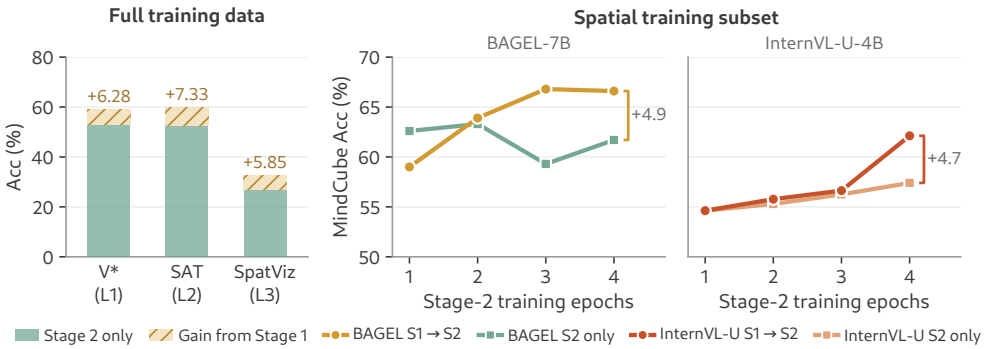


Figure 6: Ablations of pretext reasoning. Left: Stage-2-only accuracy (solid) and the gain from Stage 1 (hatched). Middle and right: MindCube accuracy with and without Stage 1 on BAGEL-7B and InternVL-U-4B.

Pretext reasoning helps at the same instruction-tuning budget. Figure 6 compares the two-stage training recipe of ThinkMorph-SSL with Stage 2 alone in two settings. **(1) Full training data.** We evaluate one representative benchmark per level. The staged run starts from the same Stage-1 checkpoint used for the main results in Tables 3 and 4. Both runs use Instruct-180K with matched Stage-2 training budgets. Pretext reasoning helps across all three task levels, with an average gain of **6.49 points**. **(2) Across backbones.** We compare BAGEL-7B and InternVL-U-4B using identical data and training settings. Stage 2 uses the spatial training subset of Instruct-180K, with the Stage-1 pretext-training budget unchanged. On MindCube, the staged BAGEL run leads from the second epoch and finishes 4.9 points ahead. On InternVL-U, the gain stays below 0.5 points for the first three epochs, then reaches 4.72 points in the fourth. These results suggest that pretext reasoning helps across backbones, with its benefit emerging as instruction tuning progresses.

Pretext reasoning should come before instruction tuning. Following the spatial-subset setting above, Figure 7 compares staged training with mixing both corpora at the same instruction-tuning budget. The mixed run stays near 57%, while staged training reaches 66.6%. Pretext reasoning therefore provides better preparation when learned before instruction tuning. These results suggest complementary roles for the two stages. Stage 1 practices producing and reading intermediate images. Stage 2 then teaches the model to use their content to advance text reasoning.

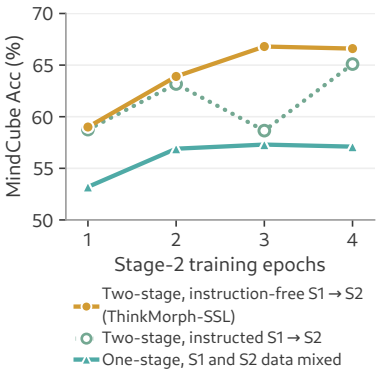


Figure 7: Effects of training order and pretext instructions. MindCube accuracy under staged versus mixed training and instruction-free versus instructed pretext training, with matched instruction-tuning budgets.

Instruction-free traces transfer better than instructed ones. Stage-1 inputs provide neither a question nor an instruction. This makes the task harder: the model must infer the pretext task from the image, without a text instruction specifying the transformation. Following the same spatial-subset setting, we add instructions only to Stage-1 inputs and keep Stage 2 unchanged. Figure 7 compares the two variants. Surprisingly, the instruction-free run leads at every checkpoint, with a modest mean advantage of 2.7 points. This suggests that Stage 1 can build the ability to produce and read intermediate images without explicit task instructions. This instruction-free preparation transfers slightly better to Stage 2, where the model learns to use image content to advance its text reasoning.

5 Related Work

Our work connects three lines of research. First, recent unified multimodal models support both visual understanding and image generation within a shared interface, using mixed-modal autoregression, diffusion- or continuous-token generation, decoupled visual pathways, or modality-bridging mechanisms [11, 13, 17, 22, 40, 56, 63, 75, 79, 80, 88, 93, 97, 99, 100]. These works focus primarily on building the unified interface itself; in contrast, we study how to convert a pretrained unified model into an interleaved reasoner.

Second, self-supervised vision pretraining has long used image-derived supervision, including rotation prediction, jigsaw puzzles, colorization, inpainting, masked image modeling, self-distillation, and image–text alignment [5, 8, 26, 34, 58, 62, 64, 65, 101, 111, 115, 120]. These methods typically learn representations, tokenizers, or alignment spaces. Pretext reasoning instead recasts these tasks as autoregressive Text–Image–Text traces, so the model learns to produce and consume visual states within its own reasoning trajectory.

Third, interleaved multimodal reasoning has explored visual grounding for chain-of-thought, external sketchpads and tools, latent visual thoughts, supervised text-image reasoning traces, and benchmarks that expose failures in visual simulation and spatial reasoning [6, 14, 15, 37, 38, 47, 49, 55, 57, 66, 69, 71, 72, 74, 84, 95, 96, 105, 106, 113, 118]. Zebra-CoT and ThinkMorph develop supervised interleaved traces that train models to generate and reason from intermediate images [30, 46]. Most of this work depends on task-specific supervised traces or predefined tools. We add a stage before such supervision, in which traces built from unlabeled images teach the model to construct visual states and continue from them. Instruction tuning then adapts these abilities to question-driven reasoning. We provide a fuller discussion in Appendix B.

6 Discussion and Conclusion

We introduced ThinkMorph-SSL, a two-stage recipe for scaling interleaved multimodal reasoning through self-supervised visual practice. Pretext-270K provides instruction-free traces for producing and interpreting intermediate images, while Instruct-180K supplies automatically constructed demonstrations for question-driven instruction tuning. Pretext training improves subsequent learning at a fixed instruction-tuning budget, and the full recipe improves performance across perception, spatial understanding, and mental world modeling. These findings show that supervision need not be designed around downstream questions to benefit interleaved reasoning. Images themselves can provide scalable training targets that make learning from task-specific demonstrations more effective.

Limitations. Our pretext pipeline constructs supervision automatically, but we have not systematically measured how performance changes with more pretext data, training compute, or larger models. We study seven visual pretext tasks and evaluate mainly perception and spatial reasoning; whether the recipe extends to other tasks and domains remains open. Interleaved reasoning also has substantial inference costs, which may limit its use when compute or latency is constrained. Finally, the base model's pretraining data are not public, so overlap with our source data cannot be ruled out.

Broader impact. We openly release the datasets, model checkpoints, and code to support research on interleaved multimodal reasoning. Better spatial reasoning could help assistive systems describe scenes and explain layouts. The model may also reproduce biases in its pretrained backbone and the public datasets used here; we have not audited those biases.

Acknowledgments and Disclosure of Funding

This project was partially funded by Samsung and by SRC CoCoSys.

References

[1] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[2] Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Didi Zhu, et al. LLaVA-OneVision-1.5: Fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*, 2025.

[3] Anthropic. Introducing Claude Sonnet 4.6. https://www.anthropic.com/news/claude-sonnet-4-6, February 2026. Anthropic blog.

[4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao

Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[5] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[6] 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. *arXiv preprint arXiv:2412.03548*, 2024.

[7] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9630–9640. IEEE, 2021.

[9] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[10] Hao Chen, Yuqi Hou, Chenyuan Qu, Irene Testini, Xiaohan Hong, and Jianbo Jiao. 360+x: A panoptic multi-modal scene understanding dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19373–19382, 2024.

[11] Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, Le Xue, Caiming Xiong, and Ran Xu. BLIP3-o: A family of fully open unified multimodal models–architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025.

[12] Leon Liangyu Chen, Haoyu Ma, Zhipeng Fan, Ziqi Huang, Animesh Sinha, Xiaoliang Dai, Jialiang Wang, Zecheng He, Jianwei Yang, Chunyuan Li, Junzhe Sun, Chu Wang, Serena Yeung-Levy, and Felix Juefei-Xu. UniT: Unified multimodal chain-of-thought test-time scaling. *arXiv preprint arXiv:2602.12279*, 2026.

[13] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-Pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

[14] Xinyan Chen, Renrui Zhang, Dongzhi Jiang, Aojun Zhou, Shilin Yan, Weifeng Lin, and Hongsheng Li. MINT-CoT: Enabling interleaved visual tokens in mathematical chain-of-thought reasoning. *arXiv preprint arXiv:2506.05331*, 2025.

[15] Zhenfang Chen, Qinhong Zhou, Yikang Shen, Yining Hong, Zhiqing Sun, Dan Gutfreund, and Chuang Gan. Visual chain-of-thought prompting for knowledge-based visual reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[16] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, Yen-Sung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi,

Ross B. Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and PixMo: Open weights and open data for state-of-the-art vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 91–104. Computer Vision Foundation / IEEE, 2025.

[17] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Shi Guang, and Haoqi Fan. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

[18] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1422–1430. IEEE Computer Society, 2015.

[19] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2014.

[20] Chengqi Duan, Kaiyue Sun, Rongyao Fang, Manyuan Zhang, Yan Feng, Ying Luo, Yufang Liu, Ke Wang, Peng Pei, Xunliang Cai, Hongsheng Li, Yi Ma, and Xihui Liu. CodePlot-CoT: Mathematical visual reasoning by thinking with code-driven images. *arXiv preprint arXiv:2510.11718*, 2025.

[21] Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. VLMEvalKit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM international conference on multimedia*, pages 11198–11201, 2024.

[22] Lijie Fan, Luming Tang, Siyang Qin, Tianhong Li, Xuan Yang, Siyuan Qiao, Andreas Steiner, Chen Sun, Yuanzhen Li, Tao Zhu, Michael Rubinstein, Michalis Raptis, Deqing Sun, and Radu Soricut. Unified autoregressive visual generation and understanding with continuous tokens. *arXiv preprint arXiv:2503.13436*, 2025.

[23] Haoran Feng, Dizhe Zhang, Xiangtai Li, Bo Du, and Lu Qi. DiT360: High-fidelity panoramic image generation via hybrid training. *arXiv preprint arXiv:2510.11712*, 2025.

[24] Deqing Fu, Ghazal Khalighinejad, Ollie Liu, Bhuwan Dhingra, Dani Yogatama, Robin Jia, and Willie Neiswanger. IsoBench: Benchmarking multimodal foundation models on isomorphic representations. *arXiv preprint arXiv:2404.01266*, 2024.

[25] 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. *arXiv preprint arXiv:2404.12390*, 2024.

[26] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[27] Vinod Goel. *Sketches of Thought*. MIT press, 1995.

[28] Google. Gemini 2.5 Flash is now in preview. https://blog.google/products-and-platforms/products/gemini/gemini-2-5-flash-preview/, April 2025. Google blog.

[29] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, Eugene Byrne, Zachary Chavis, Joya Chen, Feng Cheng, Fu-Jen Chu, Sean Crane, Avijit Dasgupta, Jing Dong, María Escobar, Cristhian Forigua, Abrham Gebreselasie, Sanjay Haresh, Jing Huang, Md Mohaiminul Islam, Suyog Dutt Jain, Rawal Khirodkar, Devansh Kukreja, Kevin J. Liang, Jia-Wei Liu, Sagnik Majumder, Yongsen Mao, Miguel Martin, Effrosyni Mavroudi, Tushar Nagarajan, Francesco Ragusa, Santhosh Kumar Ramakrishnan, Luigi Seminara, Arjun Somayazulu, Yale Song, Shan Su, Zihui Xue, Edward Zhang, Jinxu Zhang, Angela Castillo,

Changan Chen, Xinzhu Fu, Ryosuke Furuta, Cristina González, Prince Gupta, Jiabo Hu, Yifei Huang, Yiming Huang, Weslie Khoo, Anush Kumar, Robert Kuo, Sach Lakhavani, Miao Liu, Mi Luo, Zhengyi Luo, Brighid Meredith, Austin Miller, Oluwatumininu Oguntola, Xiaqing Pan, Penny Peng, Shraman Pramanick, Merey Ramazanova, Fiona Ryan, Wei Shan, Kiran K. Somasundaram, Chenan Song, Audrey Southerland, Masatoshi Tateno, Huiyu Wang, Yuchen Wang, Takuma Yagi, Mingfei Yan, Xitong Yang, Zecheng Yu, Shengxin Cindy Zha, Chen Zhao, Ziwei Zhao, Zhifan Zhu, Jeff Zhuo, Pablo Arbeláez, Gedas Bertasius, Dima Damen, Jakob J. Engel, Giovanni Maria Farinella, Antonino Furnari, Bernard Ghanem, Judy Hoffman, C. V. Jawahar, Richard A. Newcombe, Hyun Soo Park, James M. Rehg, Yoichi Sato, Manolis Savva, Jianbo Shi, Mike Zheng Shout, and Michael Wray. Ego-Exo4D: Understanding skilled human activity from first- and third-person perspectives. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 19383–19400. IEEE, 2024.

[30] Jiawei Gu, Yunzhuo Hao, Huichen Will Wang, Linjie Li, Michael Qizhe Shieh, Yejin Choi, Ranjay Krishna, and Yu Cheng. ThinkMorph: Emergent properties in multimodal interleaved chain-of-thought reasoning. *arXiv preprint arXiv:2510.27492*, 2025.

[31] Zirun Guo, Feng Zhang, Kai Jia, and Tao Jin. LLM-I: LLMs are naturally interleaved multimodal creators. *arXiv preprint arXiv:2509.13642*, 2025.

[32] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[33] 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.

[34] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 15979–15988. IEEE, 2022.

[35] Mary Hegarty. Mental animation: Inferring motion from static displays of mechanical systems. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 18(5):1084–1102, 1992.

[36] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[37] 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. *arXiv preprint arXiv:2406.09403*, 2024.

[38] Zican Hu, Xuyang Hu, Yiming Liu, Zuwei Long, Wei Liu, Yunzhuo Hao, Jiawei Gu, Linjie Li, Yu Cheng, Zhenhong Sun, Weibo Gu, Xing Sun, and Zhi Wang. Bridging interleaved multi-modal reasoning as a unified decision process. arXiv preprint arXiv:2607.03748, 2026.

[39] Mengdi Jia, Zekun Qi, Shaochen Zhang, Wenyao Zhang, Xinqiang Yu, Jiawei He, He Wang, and Li Yi. OmniSpatial: Towards comprehensive spatial reasoning benchmark for vision language models. *arXiv preprint arXiv:2506.03135*, 2026.

[40] Jiachun Jin, Zetong Zhou, Xiao Yang, Hao Zhang, Pengfei Liu, Jun Zhu, and Zhijie Deng. LatentUM: Unleashing the potential of interleaved cross-modal reasoning via a latent-space unified model. *arXiv preprint arXiv:2604.02097*, 2026.

[41] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[42] David Kirsh. Thinking with external representations. *AI & Society*, 25(4):441–454, 2010.

[43] Stephen M. Kosslyn. Measuring the visual angle of the mind's eye. *Cognitive Psychology*, 10(3):356–389, 1978.

[44] Stephen M. Kosslyn. *Image and Mind*. Harvard University Press, 1980.

[45] Jill H. Larkin and Herbert A. Simon. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11(1):65–100, 1987.

[46] Ang Li, Charles Wang, Kaiyu Yue, Zikui Cai, Ollie Liu, Deqing Fu, Peng Guo, Wang Bill Zhu, Vatsal Sharan, Robin Jia, Willie Neiswanger, Furong Huang, Tom Goldstein, and Micah Goldblum. Zebra-CoT: A dataset for interleaved vision language reasoning. *arXiv preprint arXiv:2507.16746*, 2025.

[47] Linjie Li, Mahtab Bigverdi, Jiawei Gu, Zixian Ma, Yinuo Yang, Ziang Li, Yejin Choi, and Ranjay Krishna. Unfolding spatial cognition: Evaluating multimodal models on visual simulations. *arXiv preprint arXiv:2506.04633*, 2025.

[48] Linyuan Li, Yan Wu, Xi Li, Lingli Wang, Tong Rao, Jie Zhou, Cihui Pan, and Xinchen Hui. Realsee3D: A large-scale multi-view RGB-D dataset of indoor scenes (version 1.0). https://doi.org/10.5281/zenodo.17826243, 2025. Zenodo.

[49] Ming Li, Jike Zhong, Shitian Zhao, Haoquan Zhang, Shaoheng Lin, Yuxiang Lai, Chen Wei, Konstantinos Psounis, and Kaipeng Zhang. TIR-Bench: A comprehensive benchmark for agentic thinking-with-images reasoning. *arXiv preprint arXiv:2511.01833*, 2025.

[50] Chao Liao, Liyang Liu, Xun Wang, Zhengxiong Luo, Xinyu Zhang, Wenliang Zhao, Jie Wu, Liang Li, Zhi Tian, and Weilin Huang. Mogao: An omni foundation model for interleaved multi-modal generation. *arXiv preprint arXiv:2505.05472*, 2025.

[51] Jiaqi Liao, Zhengyuan Yang, Linjie Li, Dianqi Li, Kevin Lin, Yu Cheng, and Lijuan Wang. ImageGen-CoT: Enhancing text-to-image in-context learning with chain-of-thought reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

[52] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common objects in context. *arXiv preprint arXiv:1405.0312*, 2015.

[53] Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. arXiv preprint arXiv:2411.19108, 2024.

[54] Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-GPS: Interpretable geometry problem solving with formal language and symbolic reasoning. In *The 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

[55] Wenxin Ma, Chenlong Wang, Ruisheng Yuan, Hao Chen, Nanru Dai, Yijun Yang, Chengxin Qian, Zhao-Yang Wang, Alan Yuille, and Jieneng Chen. CausalSpatial: A benchmark for object-centric causal spatial reasoning. *arXiv preprint arXiv:2601.13304*, 2026.

[56] Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Xingkai Yu, Liang Zhao, Yisong Wang, Jiaying Liu, and Chong Ruan. JanusFlow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation. *arXiv preprint arXiv:2411.07975*, 2024.

[57] Zixian Ma, Jianguo Zhang, Zhiwei Liu, Jieyu Zhang, Juntao Tan, Manli Shu, Juan Carlos Niebles, Shelby Heinecke, Huan Wang, Caiming Xiong, Ranjay Krishna, and Silvio Savarese. LATTE: Learning to think with vision specialists. *arXiv preprint arXiv:2412.05479*, 2024.

[58] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI*, Lecture Notes in Computer Science, pages 69–84. Springer, 2016.

[59] OpenAI. Introducing GPT-5. https://openai.com/index/introducing-gpt-5/, August 2025. OpenAI blog.

[60] OpenAI. Thinking with images. https://openai.com/index/thinking-with-images/, 2025.

[61] OpenSenseNova. SenseNova-U1. https://github.com/OpenSenseNova/SenseNova-U1, 2026. GitHub repository.

[62] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[63] Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Jiuhai Chen, Kunpeng Li, Felix Juefei-Xu, Ji Hou, and Saining Xie. Transfer between modalities with MetaQueries. *arXiv preprint arXiv:2504.06256*, 2025.

[64] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2536–2544. IEEE Computer Society, 2016.

[65] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Proceedings of Machine Learning Research, pages 8748–8763. PMLR, 2021.

[66] Arijit Ray, Ahmed Abdelkader, Chengzhi Mao, Bryan A. Plummer, Kate Saenko, Ranjay Krishna, Leonidas Guibas, and Wen-Sheng Chu. Mull-Tokens: Modality-agnostic latent thinking. *arXiv preprint arXiv:2512.10941*, 2025.

[67] Arijit Ray, Jiafei Duan, Ellis Brown, Reuben Tan, Dina Bashkirova, Rose Hendrix, Kiana Ehsani, Aniruddha Kembhavi, Bryan A. Plummer, Ranjay Krishna, Kuo-Hao Zeng, and Kate Saenko. SAT: Dynamic spatial aptitude training for multimodal language models. *arXiv preprint arXiv:2412.07755*, 2025.

[68] SAM 3D Team, Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J. Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, Aohan Lin, Jiawei Liu, Ziqi Ma, Anushka Sagar, Bowen Song, Xiaodong Wang, Jianing Yang, Bowen Zhang, Piotr Dollár, Georgia Gkioxari, Matt Feiszli, and Jitendra Malik. SAM 3D: 3Dfy anything in images. *arXiv preprint arXiv:2511.16624*, 2025.

[69] Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual CoT: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[70] Roger N. Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971.

[71] Mingyang Song, Haoyu Sun, Jiawei Gu, Linjie Li, Luxin Xu, Ranjay Krishna, and Yu Cheng. AdaReasoner: Dynamic tool orchestration for iterative visual reasoning. *arXiv preprint arXiv:2601.18631*, 2026.

[72] Zhaochen Su, Linjie Li, Mingyang Song, Yunzhuo Hao, Zhengyuan Yang, Jun Zhang, Guanjie Chen, Jiawei Gu, Juntao Li, Xiaoye Qu, and Yu Cheng. OpenThinkIMG: Learning to think with images via visual tool reinforcement learning. *arXiv preprint arXiv:2505.08617*, 2025.

[73] Haoxiang Sun, Lizhen Xu, Bing Zhao, Wotao Yin, Wei Wang, Boyu Yang, Rui Wang, and Hu Wei. DeepVision-103K: A visually diverse, broad-coverage, and verifiable mathematical dataset for multimodal reasoning. *arXiv preprint arXiv:2602.16742*, 2026.

[74] Dídac Surís, Sachit Menon, and Carl Vondrick. ViperGPT: Visual inference via Python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

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

[76] Meituan LongCat Team, Bin Xiao, Chao Wang, Chengjiang Li, Chi Zhang, Chong Peng, Hang Yu, Hao Yang, Haonan Yan, Haoze Sun, et al. LongCat-Next: Lexicalizing modalities as discrete tokens. *arXiv preprint arXiv:2603.27538*, 2026.

[77] Changyao Tian, Danni Yang, Guanzhou Chen, Erfei Cui, Zhaokai Wang, Yuchen Duan, Penghao Yin, Sitao Chen, Ganlin Yang, Mingxin Liu, Zirun Zhu, Ziqian Fan, Leyao Gu, Haomin Wang, Qi Wei, Jinhui Yin, Xue Yang, Zhihang Zhong, Qi Qin, Yi Xin, Bin Fu, Yihao Liu, Jiaye Ge, Qipeng Guo, Gen Luo, Hongsheng Li, Yu Qiao, Kai Chen, and Hongjie Zhang. Internvl-u: Democratizing unified multimodal models for understanding, reasoning, generation and editing. arXiv preprint arXiv:2603.09877, 2026.

[78] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. *arXiv preprint arXiv:2406.16860*, 2024.

[79] Shengbang Tong, David Fan, Jiachen Li, Jiachen Zhu, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. MetaMorph: Multimodal understanding and generation via instruction tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

[80] Shengbang Tong, David Fan, John Nguyen, Ellis Brown, Gaoyue Zhou, Shengyi Qian, Boyang Zheng, Théophane Vallaeys, Junlin Han, Rob Fergus, Naila Murray, Marjan Ghazvininejad, Mike Lewis, Nicolas Ballas, Amir Bar, Michael Rabbat, Jakob Verbeek, Luke Zettlemoyer, Koustuv Sinha, Yann LeCun, and Saining Xie. Beyond language modeling: An exploration of multimodal pretraining. *arXiv preprint arXiv:2603.03276*, 2026.

[81] Barbara Tversky and Masaki Suwa. Thinking with sketches. In Arthur Markman and Kristin Wood, editors, *Tools for Innovation*. Oxford University Press, 2009.

[82] Barbara Tversky, Masaki Suwa, Maneesh Agrawala, Julie Heiser, Chris Stolte, Pat Hanrahan, Doantam Phan, Jeff Klingner, Marie-Paule Daniel, Paul Lee, et al. Sketches for design and design of sketches. *Human Behaviour in Design: Individuals, Teams, Tools*, pages 79–86, 2003.

[83] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In William W. Cohen, Andrew McCallum, and Sam T. Roweis, editors, *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, ACM International Conference Proceeding Series, pages 1096–1103. ACM, 2008.

[84] Haozhe Wang, Alex Su, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. *arXiv preprint arXiv:2505.15966*, 2025.

[85] Siting Wang, Minnan Pei, Luoyang Sun, Cheng Deng, Yuchen Li, Kun Shao, Zheng Tian, Haifeng Zhang, and Jun Wang. SpatialViz-Bench: A cognitively-grounded benchmark for diagnosing spatial visualization in MLLMs. *arXiv preprint arXiv:2507.07610*, 2026.

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

[87] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[88] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Yang, Qiying Sun, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

[89] Zehan Wang, Ziang Zhang, Jiayang Xu, Jialei Wang, Tianyu Pang, Chao Du, Hengshuang Zhao, and Zhou Zhao. Orient anything V2: Unifying orientation and rotation understanding. *arXiv preprint arXiv:2601.05573*, 2026.

[90] Zifu Wang, Junyi Zhu, Bo Tang, Zhiyu Li, Feiyu Xiong, Jiaqian Yu, and Matthew B. Blaschko. Jigsaw-R1: A study of rule-based visual reinforcement learning with jigsaw puzzles. *TMLR*, 2025.

[91] Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, et al. CharXiv: Charting gaps in realistic chart understanding in multimodal LLMs. *Advances in Neural Information Processing Systems*, 37:113569–113697, 2024.

[92] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Shengming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-Image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

[93] Chengyue Wu et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint arXiv:2410.13848*, 2024.

[94] Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal LLMs. *arXiv preprint arXiv:2312.14135*, 2023.

[95] Qiong Wu, Xiangcong Yang, Yiyi Zhou, Chenxin Fang, Baiyang Song, Xiaoshuai Sun, and Rongrong Ji. Grounded chain-of-thought for multimodal large language models. *arXiv preprint arXiv:2503.12799*, 2025.

[96] Xuecheng Wu, Jiaxing Liu, Danlei Huang, Xiaoyu Li, Yifan Wang, Chen Chen, Liya Ma, Xuezhi Cao, and Junxiao Xue. ViC-Bench: Benchmarking visual-interleaved chain-of-thought capability in MLLMs with free-style intermediate state representations. *arXiv preprint arXiv:2505.14404*, 2025.

[97] Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, Song Han, and Yao Lu. VILA-U: A unified foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429*, 2024.

[98] xAI. Grok-1.5 vision preview. https://x.ai/news/grok-1.5v, April 2024. Introduces the RealWorldQA benchmark.

[99] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

[100] Jinheng Xie, Zhenheng Yang, and Mike Zheng Shou. Show-o2: Improved native unified multimodal models. *arXiv preprint arXiv:2506.15564*, 2025.

[101] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. SimMIM: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[102] Jinbo Xing, Zeyinzi Jiang, Yuxiang Tuo, Chaojie Mao, Xiaotang Gai, Xi Chen, Jingfeng Zhang, Yulin Pan, Zhen Han, Jie Xiao, Keyu Yan, Chenwei Xie, Chongyang Zhong, Kai Zhu, Tong Shen, Lianghua Huang, and Yu Liu. Wan-Weaver: Interleaved multi-modal generation via decoupled training. *arXiv preprint arXiv:2603.25706*, 2026.

[103] Shusheng Yang, Jihan Yang, Pinzhi Huang, Ellis L Brown II, Zihao Yang, Yue Yu, Shengbang Tong, Zihan Zheng, Yifan Xu, Muhan Wang, et al. Cambrian-S: Towards spatial supersensing in video. In *The Fourteenth International Conference on Learning Representations*, 2026.

[104] Sihan Yang, Runsen Xu, Yiman Xie, Sizhe Yang, Mo Li, Jingli Lin, Chenming Zhu, Xiaochen Chen, Haodong Duan, Xiangyu Yue, Dahua Lin, Tai Wang, and Jiangmiao Pang. MMSI-Bench: A benchmark for multi-image spatial intelligence. *arXiv preprint arXiv:2505.23764*, 2025.

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

[106] Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Lijuan Zeng, and Lijuan Wang. MM-REACT: Prompting ChatGPT for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023.

[107] Baiqiao Yin, Qineng Wang, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, Saining Xie, Manling Li, Jiajun Wu, and Li Fei-Fei. Spatial mental modeling from limited views. In *Structural Priors for Vision Workshop at ICCV'25*, 2025.

[108] Youngjin Yoon, Hae-Gon Jeon, Donggeun Yoo, Joon-Young Lee, and In So Kweon. Learning a deep convolutional network for light-field image super-resolution. In *2015 IEEE International Conference on Computer Vision Workshop, ICCV Workshops 2015, Santiago, Chile, December 7-13, 2015*, pages 57–65. IEEE Computer Society, 2015.

[109] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Tianyou Liang, Guanying Chen, Shuguang Cui, and Xiaoguang Han. MVImgNet: A large-scale dataset of multi-view images. In *CVPR*, 2023.

[110] Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. MMMU-Pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15134–15186, 2025.

[111] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[112] Xiaoyu Zhan, Wenxuan Huang, Hao Sun, Xinyu Fu, Changfeng Ma, Shaosheng Cao, Bohan Jia, Shaohui Lin, Zhenfei Yin, Lei Bai, Wanli Ouyang, Yuanqi Li, Jie Guo, and Yanwen Guo. Actial: Activate spatial reasoning ability of multimodal large language models. *arXiv preprint arXiv:2511.01618*, 2025.

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

[114] Jiajie Zhang and Donald A. Norman. Representations in distributed cognitive tasks. *Cognitive Science*, 18(1):87–122, 1994.

[115] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III*, Lecture Notes in Computer Science, pages 649–666. Springer, 2016.

[116] Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, et al. MME-RealWorld: Could your multimodal LLM challenge high-resolution real-world scenarios that are difficult for humans? *arXiv preprint arXiv:2408.13257*, 2024.

[117] Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, Yandong Guo, and Lei Zhang. Recognize anything: A strong image tagging model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024 - Workshops, Seattle, WA, USA, June 17-18, 2024*, pages 1724–1732. IEEE, 2024.

[118] Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. DeepEyes: Incentivizing "thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025.

[119] Enshen Zhou, Jingkun An, Cheng Chi, Yi Han, Shanyu Rong, Chi Zhang, Pengwei Wang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, et al. RoboRefer: Towards spatial referring with reasoning in vision-language models for robotics. *arXiv preprint arXiv:2506.04308*, 2025.

[120] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT: Image BERT pre-training with online tokenizer. In *International Conference on Learning Representations*, 2022.

[121] Yiyang Zhou, Haoqin Tu, Zijun Wang, Zeyu Wang, Niklas Muennighoff, Fan Nie, Yejin Choi, James Zou, Chaorui Deng, Shen Yan, Haoqi Fan, Cihang Xie, Huaxiu Yao, and Qinghao Ye. When visualizing is the first step to reasoning: MIRA, a benchmark for visual chain-of-thought. *arXiv preprint arXiv:2511.02779*, 2025.

[122] Muzhi Zhu, Hao Zhong, Canyu Zhao, Zongze Du, Zheng Huang, Mingyu Liu, Hao Chen, Cheng Zou, Jingdong Chen, Ming Yang, and Chunhua Shen. ACTIVE-O3: Empowering multimodal large language models with active perception via GRPO. In *International Conference on Machine Learning*, 2026.

A Overview of the Appendix

This appendix is organized as follows:

- Section B extends the related-work discussion.
- Section C reports the composition of Pretext-270K and Instruct-180K.
- Section D details the task-specific Instruct-180K pipelines and the shared trace synthesis prompt.
- Section E shows example traces per Instruct-180K task, and Section F shows one Text–Image–Text trace per pretext task.
- Section G reports training and evaluation details: hyperparameters of the main and ablation runs, the loss, system prompts, compute, the VLMEvalKit configuration, per-benchmark scoring, and judge prompts.
- Section H presents qualitative good cases and failure modes of ThinkMorph-SSL, data overlap, and alternative routes without visual-state training.

B Extended Related Work

Unified multimodal models. Recent unified multimodal models support visual understanding and image generation within a shared interface. Existing approaches vary in how they connect the two capabilities, from early-fusion mixed-modal modeling [75] to autoregressive token-based unification [88, 97, 99, 100] and continuous-token or diffusion-based variants [11, 22]; another line decouples or harmonizes visual pathways [13, 56, 93] or introduces latent or query-based bridges between modalities [40, 63], and a complementary line studies adaptation- or pretraining-based recipes [79, 80]. We build on BAGEL, which shows that large-scale interleaved pretraining can yield emergent unified capabilities [17]. These works primarily address how to build or adapt the unified interface itself. In contrast, ThinkMorph-SSL studies the orthogonal training bridge from generic unified competence to interleaved reasoning behavior: pretext reasoning is inserted between unified pretraining and question-driven instruction tuning to teach the model the mechanics of Text–Image–Text continuation before supervised reasoning.

Self-supervised visual pretraining. A long line of work learns visual representations without human labels by constructing supervision from the image itself. Early pretext tasks include instance and rotation discrimination [19, 26], spatial-context and jigsaw puzzles [18, 58], color and inpainting reconstructions [64, 115], and motion-based signals from video [1, 87]. These ideas later scaled through masked image modeling [5, 34, 101, 120], self-distillation [8, 62], and image–text alignment [65, 111]. Despite their diversity, these objectives primarily produce an encoder, a tokenizer, or an alignment space; the model is not trained to use visual generation as part of its own computation. Pretext reasoning revisits this paradigm but changes the interface: each restoration task is reformulated as an autoregressive Text–Image–Text trace. The resulting supervision trains a unified model to continue interleaved perception and generation trajectories rather than merely produce a visual representation.

Interleaved multimodal reasoning. Cognitive science has long argued that human spatial reasoning often relies on mental imagery and visual simulation rather than purely verbal deduction [44, 70]. This motivates models that can reason through intermediate visual states instead of expressing every step only in text. Prior work brings visual content into reasoning in three main ways. The first augments textual chain-of-thought with visual grounding [15, 69, 95] or with external programmatic[20], agent-style, and sketchpad tools [37, 74, 106]; recent work further trains models to invoke vision specialists or pixel-space operations adaptively [57, 71, 72], often through reinforcement learning over images [60, 84, 118, 122]. These methods improve evidence gathering and interpretability, but the reasoning trace remains primarily textual or tool-mediated, and its visual operations are bounded by predefined tools or experts.

The second line incorporates visual state directly into the reasoning trajectory. Some methods train on supervised interleaved text-image reasoning traces [14, 30, 46]; others introduce latent visual tokens as task-specific intermediates, mental imagery, internal sketchpads, or modality-agnostic thought tokens

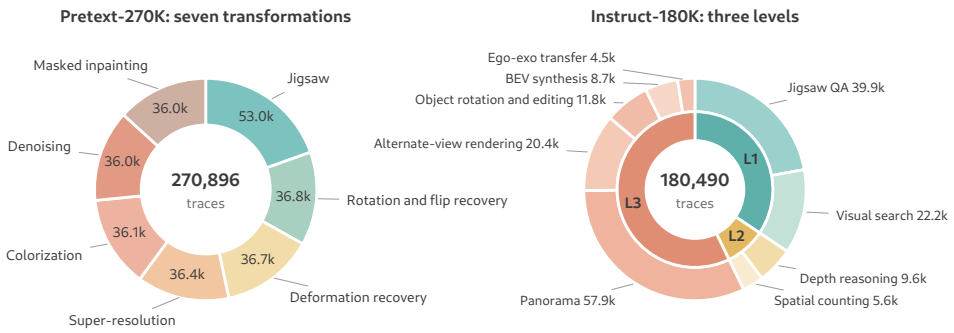


Figure 8: Corpus composition. Left: traces per pretext transformation in Pretext-270K. Right: traces in Instruct-180K per level (inner ring; L1 basic visual perception, L2 static spatial understanding, L3 mental world modeling) and per construction pipeline (outer ring; nine pipelines).

[6, 66, 105, 113]. Complementary work studies test-time scaling and representation bottlenecks for such multimodal reasoning processes [12, 40]. Recent benchmarks further motivate this direction, showing that current models struggle when answers depend on intermediate visual states, visual simulation, or causal-spatial structure rather than language-only reasoning [47, 49, 55, 96, 121].

The third line studies interleaved multimodal generation, where mixed image-text content is the final artifact rather than an intermediate reasoning process [31, 50, 51, 102]. This direction is complementary to our goal, since we use generated visual states as intermediate thoughts to improve multimodal understanding and reasoning. Across these lines, the dominant recipe relies on supervised reasoning traces, which are costly to scale and limited in coverage. We target the stage that prior work largely skips. Pretext reasoning provides instruction-free training, before any question-driven supervision, that teaches a unified model the mechanics of Text–Image–Text continuation, while Instruct-180K then fits within the supervised interleaved-reasoning lineage as a question-driven corpus.

C Data Composition

Figure 8 shows the composition of both corpora. Pretext-270K draws its images from SA-1B, PixMo-Cap, VSI-590K, LLaVA-OneVision-1.5-Instruct-Data, and DeepVision-103K for all seven transformations, and additionally from Realsee3D, 360+x, and Matterport3D for Jigsaw.

D Implementation Details of the Instruct-180K Curation Pipeline

Table 6 maps each level of Instruct-180K to its task families and to the kind of intermediate image its traces construct.

Table 6: Role of the intermediate visual state at each level of Instruct-180K.

Level	Role of the visual state	Task families	Intermediate image
L1: Basic visual perception	Focus on visible evidence	Visual search, Jigsaw QA	Target boxed or zoomed in the scene; unscrambled image
L2: Static spatial understanding	Organize visible spatial relations	Spatial counting, depth reasoning	Instance crops; depth map
L3: Mental world modeling	Reconstruct unseen or transformed states	Bird's-eye view synthesis, panorama, alternate-view rendering, object rotation, ego–exo perspective transfer	Bird's-eye view; panorama; alternate view; rotated object; another agent's view

Mental world modeling further splits into viewpoint synthesis, where the camera moves while the scene stays fixed, and object manipulation, where the camera stays while the object changes. Within viewpoint synthesis the subsections are ordered by how far the synthesized view departs from the input egocentric frame. Panorama stays at the same camera position with a wider field of view. Bird's-eye view synthesis leaves the agent for a world-anchored overhead. Ego–exo perspective

transfer and alternate-view rendering both adopt another entity's viewpoint; the first learns from real paired ego/exo captures and the second from single-image 3D lifts, since no single source covers both.

D.1 Interleaved chain-of-thought generation

Each row arrives with the question stem, the gold label, input images, and zero or more reasoning images that have already been rendered or photographed. A separate vision-language pass writes the trace a student model is trained to mimic. For N attached reasoning images the writer returns $N + 1$ text blocks alternating with those images. The first block is written before any generated view is consulted: it inspects the inputs, states what visualization will come next, and explains why. Each later block is written after its corresponding image: it describes what that view makes explicit, ties it back to the inputs, and either plans the next view or concludes.

The writer is invoked through two cascaded prompts. An optional first pass paraphrases the question and answer to diversify surface form without changing semantics. The main pass casts the model as an agent that generates the intermediate views during reasoning, under three constraints: it must not refer to the views as externally supplied, it must not leak the gold answer before the final block, and it must use descriptive view names (bird's-eye view, depth map) rather than ordinal labels. The model sees the stem, the answer, a per-task suggested reasoning approach, and the input and reasoning images in alternation. When per-row metadata indicates which reasoning view corresponds to which input, that correspondence is attached as a hidden hint usable only for internal consistency. Outputs that omit any of the $N + 1$ expected blocks are rejected and re-run.

Tasks that do not commit to a visual reasoning sequence skip reasoning images. The model returns a single continuous-prose chain under a lighter system prompt that forbids bullet inventories and mechanical enumeration.

Per-task templates supply both the suggested reasoning approach and the short captions used to label the attached reasoning images. The fixed system roles below are shared across tasks. The template excerpts that follow are representative rather than exhaustive.

`REWRITE_SYSTEM_PROMPT` (optional question–answer paraphrase).

```
Rewrite the given question and answer with diverse wording and sentence
structure while preserving correctness.

Rules:
1. Preserve the original meaning -- do not alter spatial relations,
   change counts, swap entities, or introduce ambiguity. All entities,
   constraints, and spatial details must be retained.
2. Vary vocabulary, sentence structure, and phrasing style freely. Do
   not add hints toward the answer.
3. For multiple-choice questions, keep the option order unchanged; you
   may rephrase option text if the meaning is preserved.
4. Output a JSON object with keys: "question", "answer".
```

`CORE_SYSTEM_PROMPT` (interleaved reasoning images).

```
You are an advanced spatial reasoning AI with a unique capability: you
can generate intermediate visualizations (e.g., bird's-eye views, side
profiles, depth maps, rotated perspectives, or step-by-step
constructions) to help understand spatial relationships.

Your Task:
You will be given:

1. Input Images: One or more images related to the task (e.g.,
   multiple viewpoints of a scene, a puzzle layout, a diagram, etc.).
2. Intermediate Generated Views: A sequence of visual aids you
   "generate" during reasoning (e.g., new perspectives, step-by-step
```

```
   assembly stages, annotated overlays, etc.).
3. A spatial reasoning question and its correct answer.

Your Goal:
Generate textual reasoning that connects these images step-by-step.
Since you have the capability to "generate" these intermediate views,
you must explain why you need them and what they show.

For N intermediate views, you need N+1 reasoning blocks:

* reasoning_thought_0: BEFORE generating any view
  - Analyze the initial Input Images and questions.
  - Explicitly state what type of view you will generate next and why.

* reasoning_thought_1: AFTER generating the first view
  - Now you can see the generated image.
  - Describe the type of view you generated (e.g., "This top-down
    perspective reveals...") and analyze its content.
  - Synthesize this with the original inputs.
  - State what you will generate next (if needed).

* reasoning_thought_2: AFTER generating the second view
  - Describe the new view and what information it provides.
  - Synthesize all observations to derive the final answer.

* (Continue this pattern for more views if needed.)

Output Format: Return your reasoning as a JSON object with the
required keys.

Important Notes:

1. Never describe a view before generating it. In reasoning_thought_0
   you only state your intention to generate a specific type of view.
2. Dynamic Terminology: Do not use generic labels like "Image 1".
   Instead, use descriptive terms based on the view type, such as
   "the bird's-eye view", "the depth map", "the side-angle
   perspective", or "the zoomed-in visualization".
3. Cross-Referencing: Intermediate views help with structure but may
   lack specific details. If you need, please refer back to the Input
   Images for fine-grained object identification.
4. Step-by-Step Deduction: Your conclusion must be the logical result
   of combining the input images with your generated visualizations.
5. Do not state, hint at, or assume the correct answer in the
   reasoning process. Do not refer to intermediate views as "provided"
   or "given"; describe them as views you generate during reasoning.
```

TEXT_ONLY_SYSTEM_PROMPT (no intermediate images).

```
You are an advanced spatial reasoning AI.

Your Task:
You will be given:

1. Input Images: One or more images related to the task.
2. A spatial reasoning question and its correct answer.

Your Goal:
Generate a natural chain-of-thought reasoning that explains how to
arrive at the correct answer by analyzing the input images.

Output Format: Return your reasoning as a JSON object with exactly one
```

```
key: "reasoning".

Important Notes:

1. Think naturally. Write your reasoning as a continuous, flowing
   thought process -- as if you are talking through the problem.
   Avoid rigid numbered lists, bullet-by-bullet image inventories, or
   mechanical enumeration patterns like "Count = 1, Count = 2".
2. Build toward the answer progressively. Do not state or assume the
   correct answer at the beginning. Let your conclusion emerge from
   your observations and spatial analysis.
```

Counting with mask overlay (`lvis_mask_counting`).

```
Task: Count how many instances of the queried object category are in
the image.
reasoning_thought_0 (BEFORE generating any view): Observe the input
image, identify the queried object category, and note any challenges
for counting (occlusion, clutter, similar-looking objects). Explain
why you will generate a mask overlay that isolates only the target
instances on a white background.
reasoning_thought_1 (AFTER generating the mask overlay): The generated
mask overlay shows only the target instances on a white background
with black contours around each instance. Count each distinct instance
visible in the overlay carefully and state the final count.
```

Top-down counting on multi-view scenes (`topdown_counting`).

```
Task: Count how many instances of the queried object category are in
the full scene.
reasoning_thought_0: Briefly inspect the multi-view input images, and
explain why you will generate a top-down view.
reasoning_thought_1: Describe what the generated top-down view
reveals, count the queried category carefully in that view, and
finish with the final count consistent with the question.
```

RefSpatial depth with two generated views (`refspatial_point_depth`).

```
Task: Answer a RefSpatial depth question using the available generated
depth-support view or views. When there are two generated views, use
the first to localize the correct point or points and the second to
make the final depth judgment.
For reasoning_thought_0, briefly inspect the input image and
question, explain what depth relation must be resolved, and state
what kind of depth-support view you will form next. Do not answer yet.
```

Ego–exo perspective transfer (`egoexo_perspective_taking`).

```
Task: Answer a perspective-taking spatial question from a third-person
scene image.
The input image shows the scene from an outside view, while the
generated view represents the queried person's first-person
perspective.
Do not mention raw field names such as ‘images‘, ‘reasoning_image_0‘,
or any hidden metadata.
Do not mention markers, highlights, or annotations that are not
visible to the final solver.
Instead, refer naturally to the outside scene view and the generated
first-person view.
```

```
reasoning_thought_0 (BEFORE generating any view): Inspect the
third-person image, identify which person the question refers to,
locate the relevant object or objects mentioned in the question, and
explain why the answer cannot be judged directly from the outside
camera.
State that you will generate a first-person view from that person's
position and facing direction so the relation can be judged from
their own perspective.
Do not reveal the final answer yet.
reasoning_thought_1 (AFTER generating the first view): Analyze the
generated first-person view from that person's perspective.
Use it to determine the asked relation precisely, such as left versus
right, above versus below, or front versus behind.
Keep the reasoning grounded in visible scene evidence and perspective
transformation only, then finish with the correct answer.
```

Panorama merge (`panorama_spatial`).

```
Task: Answer a spatial reasoning question about an indoor or outdoor
scene captured from two partially overlapping viewpoints (left view
and right view from the same camera position).
The merged panoramic view provides a wider field of view that reveals
the full spatial layout.
reasoning_thought_0 (BEFORE generating any view): Inspect the two
input images carefully.
Identify the key objects, scene areas, and spatial cues mentioned in
the question.
Note how the two views overlap at the center (shared region) and what
each view uniquely shows on its periphery.
Explain that you will generate a merged panoramic view to see the
complete spatial layout and resolve the relationship asked. Do not
answer yet.
reasoning_thought_1 (AFTER generating the merged view): Analyze the
generated merged panoramic view.
Use it to determine the full spatial layout -- identify where objects
and scene areas are located relative to each other and relative to
the camera.
Cross-reference with the original input images for fine-grained
object identification.
Derive the final answer through step-by-step spatial reasoning
grounded in the visual evidence.
```

Mindcube-style left/right with two top-downs (`template_02_directional_query`).

```
Task: Answer a left/right query for the target object from one
queried input viewpoint.
Do not mention raw field names such as ‘reasoning_image_0‘,
‘reasoning_image_1‘, ‘prompt_metadata‘, or
‘reasoning_image_1_matches_image_index‘ in the output reasoning.
Do not mention any hidden hint, metadata, annotation, field name, or
externally provided correspondence in the output reasoning.
Instead, refer to the visual evidence naturally as the first
generated top-down view, the rotated top-down view, or the
corresponding original image.
reasoning_thought_0 (BEFORE generating any view): Inspect the queried
input image and the other input images briefly. State that you will
first form a bird's-eye view of the queried viewpoint, which helps
convert the egocentric left/right question into a rotation on the
top-down layout. Do not answer yet.
reasoning_thought_1 (AFTER generating the first view): Analyze that
first bird's-eye view, then apply the 90-degree rotation rule
```

```
(clockwise for left questions, anticlockwise for right questions in
this dataset) to explain why you need the second generated view.
Keep this step focused on forming the rotated view itself, without
yet naming which original image it matches.
reasoning_thought_2 (AFTER generating the second view): State which
original input image has the same orientation as the rotated view,
then use that matched original input image to identify what lies in
front in the aligned view, because that matched original image is
the normal-perspective version of the rotated view.
So the actual object judgment should be made from the matched
original input image, then translated back into the answer for the
queried viewpoint.
Use the original images to confirm object identity and appearance
details, and keep the explanation grounded in visual reasoning only.
```

Downstream, modality-specific post-filters (for example the panorama CoT consistency check in Section D.4.1) assume these fields are already present on each row.

D.2 Basic visual perception

D.2.1 Visual search

Visual search targets a single object inside an object-dense scene. Images must contain at least 135 annotated bounding boxes to ensure sufficient object density for unambiguous target selection. For each candidate target, a templated GPT call conditioned on the full image with a red bounding box drawn around the target or a high-resolution crop of that object generates a four-option multiple-choice question spanning fine-grained attribute recognition, spatial relationships, OCR, and visual reasoning. The correct answer must be verifiable from the cropped view alone, so the intermediate image—the full scene with the red bounding box rendered—is by construction the visual artifact required to answer. A four-model judge ensemble (GPT-4.1, Gemini-2.5 Pro, GPT-5.1, and Gemini-3-Pro-Preview) then rejects items on any of four failure criteria: visual mismatch or ambiguity (`VIS_FAIL`), poor question logic (`LOGIC_FAIL`), incorrect ground-truth answer (`ANS_ERR`), or absence of visual evidence in the crop (`NO_EVID`).

D.3 Static spatial understanding

D.3.1 Spatial counting

Object-counting supervision is sampled from LVIS [32], which provides high-quality instance segmentation masks for a large open vocabulary. For each image we enumerate instances whose mask area clears a minimum pixel threshold, crop each instance with a small margin from its axis-aligned box, and pair the crops with questions whose gold counts come from the mask inventory alone, with no language-model labelling of the number. The task tests whether the model can aggregate fragmented visual evidence into a verifiable count.

D.3.2 Depth reasoning

Depth-oriented questions are subsampled from the RefSpatial corpus introduced in RoboRefer [119], whose annotations include depth-related probes. We keep the original question and answer semantics but replace the depth or disparity visualization: scalar maps are re-coloured with the viridis colormap for perceptual uniformity and clearer contrast in printed traces, without changing the underlying geometry.

D.4 Mental world modeling

D.4.1 Viewpoint synthesis

Panorama. The smallest departure from the input is to stay at the same camera position and widen the field of view. Supervision for this comes from equirectangular panoramas (Matterport3D [9, 23], RealSee-3D [48], 360+x [10] scans). Because we can synthesize pairs of perspective crops at

controlled yaw, pitch, and field of view, the yaw separation between the two views is known exactly, which lets us derive ground-truth answers for many spatial questions analytically rather than asking a vision-language model to read them off pixels. Each panorama becomes a stream of multiple-choice questions whose images are two perspective crops and whose reasoning aid is the side-by-side composite of those crops.

We synthesize two question families on the same view-extraction frontend, distinguished by who decides the answer. A template-driven family covers cases where view metadata does not pin down the answer (compass premises, measurement, object identification, scene navigation, camera motion); a vision-language model writes the question and picks the gold option, gated by metadata-consistency checks on the subset where geometry is verifiable. For the geometric subset where metadata determines the answer, we use an analytical generation pipeline: the model only names objects and writes prose around an analytically fixed direction. The pipeline runs in four stages.

- **Step 1: Perspective view extraction.** For each panorama we sample a base pair (left, right) sharing the camera position but offset in yaw. Per-view FOV is drawn from $U(60^\circ, 90^\circ)$, the desired angular overlap from $U(-5^\circ, 50^\circ)$ (negative values indicate a gap), and a small pitch jitter from $U(-5^\circ, 5^\circ)$. The yaw separation that realizes the requested overlap is

  $$\mathrm{yaw}_{\mathrm{sep}} = \max\Big(0,\ \tfrac{1}{2}(\mathrm{fov}_L + \mathrm{fov}_R) - \mathrm{overlap}\Big),$$

  placed symmetrically about the panorama's reference yaw. Each crop is rendered by direct equirectangular-to-perspective resampling: pixels are lifted to camera-frame rays, rotated by the requested yaw and pitch, converted back to spherical coordinates, and bilinearly sampled from the panorama. No distortion model is needed because the source is already rectified to equirectangular convention.

  To support multi-image mental spatial reasoning and cannot-be-determined questions, we extend each scene with additional crops on both sides of the base pair. From each base view, we sample a chain of crops with random yaw offsets, stopping when a candidate overlaps with the opposite base view above a small threshold. The resulting auxiliary views tile the half-sphere away from the base pair without re-entering it. We save the side-by-side composite of the left and right chains as the reasoning-aid image attached to every question for the scene.
- **Step 2: Question synthesis.** The template catalog covers photographer-relative directions, compass premises, object–object relations, camera motion (translation and rotation), measurement, and object identification, plus a reserved cannot-be-determined category for non-overlapping pairs. Templates are sampled per scene with category-specific weights. For every scene we compute camera deltas from the view metadata,

  $$\Delta\mathrm{yaw} = (\mathrm{yaw}_2 - \mathrm{yaw}_1 + 180) \bmod 360 - 180, \quad \Delta\mathrm{pitch} = \mathrm{pitch}_2 - \mathrm{pitch}_1,$$

  and inject them into the prompt in plain English ("the viewpoint shifts to the right by $|\Delta\mathrm{yaw}|^\circ$") so the model has ground-truth camera motion when answering rotation or translation questions. A forbidden-token guard rejects any output mentioning stitching, panoramas, yaw, or file paths. Compass questions are further constrained by a rule that a cardinal premise on view 1 fixes only one axis: the perpendicular axis is determinable only with an orthogonal reference, and "in front of a wall" is depth rather than a cardinal direction.

  For samples whose answer can be derived from yaw alone under the equal-distance assumption that anchor and target sit at the centers of their respective views at the same depth from the panorama's optical center. Setting $\theta_A = 0$ and $\theta_B = (\mathrm{yaw}_B - \mathrm{yaw}_A) \bmod 360^\circ$ normalized into $[-180^\circ, 180^\circ)$, the world-frame direction from anchor to target is

  $$\mathrm{direction}_{\mathrm{world}} = \mathrm{atan2}\big(\sin\theta_B - \sin\theta_A,\ \cos\theta_B - \cos\theta_A\big),$$

  which is rotated into the anchor's local frame using a fixed lookup of forward directions for a whitelist of oriented objects (TV, bed, sofa, sink, kitchen island). Cardinal-chain templates instead carry a cardinal premise from view A to view B by adding the clockwise yaw delta and snapping to the nearest 45° (8-direction) or 90° (4-direction) bin.

  Pure yaw-based direction is fragile when anchor and target sit on different walls. A yaw-clockwise target on the back wall is geometrically back from the camera rather than to the right. For relational templates that ask about a target in view B relative to an anchor in view A, we replace the bearing rule with the depth-and-yaw hybrid in Table 7. The model is asked only for a 3-way depth comparison (closer, same, or further), which we combine with the sign of $\Delta\mathrm{yaw}$ to produce an 8-direction label. The 25° dead-zone reflects that base pairs are typically separated by 20–80°:

Table 7: Depth–yaw mapping to 8-direction labels for the relational templates. The 25° dead-zone rejects ambiguous pairs.

Depth	$\lvert\Delta\mathrm{yaw}\rvert < 25^\circ$	$\Delta\mathrm{yaw} > 0$	$\Delta\mathrm{yaw} < 0$
closer	front	front-right	front-left
same	reject	right	left
further	back	back-right	back-left

pairs below the dead-zone with a clear depth signal read as straight front or back, while pairs in the dead-zone with no depth signal yield no usable direction and are rejected.

A first pass typically leaves coverage gaps in some categories, which we patch with targeted regeneration that re-runs the scene loop on a restricted template pool. The regeneration shares the per-scene RNG seeding, so its output concatenates with the main pass without overlap.

- **Step 3: Verification.** Every candidate clears four cumulative gates before it is written. The forbidden-leak guard disqualifies any stem or option that mentions stitching, panoramas, yaw, or file paths. The direction–metadata consistency check parses the answer text of verifiable templates (camera rotation, translation, photographer-relative direction) for direction keywords and compares them against $\Delta\mathrm{yaw}$ and $\Delta\mathrm{pitch}$: yaw shifts above 3° require a horizontal direction with the correct sign and reject answers that are purely vertical, and pitch shifts work analogously. Uniqueness grounding sends every named noun phrase from the questions back to the model in a separate call asking it to count visible candidate matches in its image, and only objects flagged as uniquely present survive; we found object naming from a single forward pass to be the largest source of unverifiable references. Finally, a reverse-solve gate re-runs each question blind against a fresh model call with the exact images and options under a prompt that requires explicit object localization, angular and depth judgment, 8-direction classification, and option matching, and items where the reverse-solver disagrees with the generator are dropped.
- **Step 4: Post-processing.** After CoT generation we apply a conservative line-by-line filter as a final defense against CoT-level inconsistencies that the per-question gates in Step 3 cannot see. The filter drops rows whose reasoning chain contradicts the rotation direction encoded in the gold option, whose CoT is empty or under twenty whitespace tokens, or whose gold letter does not extract to a non-empty option text in the prompt.

Bird's-eye view synthesis. The next departure leaves the input agent's position altogether and synthesizes a world-anchored overhead view. Source scenes come from Mindcube [107]. An overhead view makes object positions, distances, and ego-relative directions directly readable, so the model can consult it as an intermediate image during reasoning. To be usable as a visual thought, the synthesized map must be geometrically faithful so that the original answer survives the view substitution, stable under sparse-view inputs, and realistic enough that a unified VLM treats it as a photograph rather than a diagram.

To our knowledge no public tool produces a sparse-view top-down image of an indoor scene under these constraints. We tried three natural candidates from adjacent fields, summarized in Table 8 and illustrated in Figure 9.

Candidates that are strong on appearance tend to be weak on geometric consistency, and candidates that are strong on geometry produce outputs that do not look photographic. We split the problem along that boundary. A reconstruction stage commits to where objects are and how they are oriented relative to a horizontal support plane, and a generative stage paints that layout to look photographic. Steps 1–3 build the geometric layout, and Step 4 renders it and paints in the texture.

- **Step 1: Filtered point-cloud reconstruction.** Let $\mathcal{I} = \{I_t\}_{t=1}^{T}$ denote the input views. Running Pi3 on $\mathcal{I}$ produces dense 3D points with per-point confidence scores $w_i \in [0, 1]$, RGB colors copied from the source pixels, and camera-to-world matrices. We discard low-confidence points and points adjacent to depth discontinuities. The latter removes the boundary halos Pi3 produces around silhouettes, which would otherwise bleed into the top-down rendering. The result is a colored point cloud

$$\mathcal{P} = \{(\mathbf{x}_i,\, \mathbf{c}_i,\, w_i)\}_{i=1}^{N},$$

which we subsample with probability proportional to w_i to bias survival toward high-confidence geometry.

Table 8: Off-the-shelf candidates we considered for sparse-view top-down construction, and the way each one fails.

Candidate	What it does	Why it fails in our setting
SAM3 + SAM3D	Detect objects, recover per-object 3D shape and pose, composite into a top-down view.	Requires an explicit object vocabulary, ingests views one at a time, and the rendered composite is visibly synthetic.
Image-edit / view-conversion (Qwen-Edit, Nano Banana Pro)	Treat the top-down view as image-to-image editing.	Object orientations drift, items disappear, and the camera angle is inconsistent across runs. Fine-tuning Qwen-Edit on synthetic top-down pairs improves in-distribution scenes but still hallucinates content, and inference slows as the input view count grows.
Novel-view synthesis (Gaussian Splatting, Stable Virtual Camera)	Reconstruct or interpolate views from the input set.	Built for small camera displacements; rotating to nadir is far outside their regime, and sparse inputs blur quickly under any large rotation.

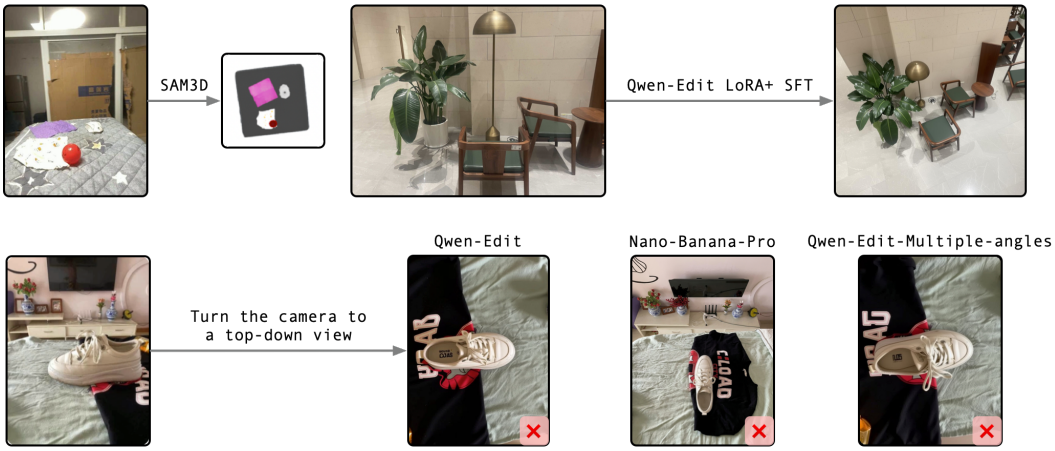


Figure 9: Representative failures of off-the-shelf and fine-tuned candidates on real scenes. *Top:* a SAM3D-based composite of the small objects on the bed yields a generic synthetic-looking platform, and a Qwen-Edit model further fine-tuned (LoRA + SFT) on synthetic top-down pairs only resamples a different angle of that fabricated scene rather than depicting the actual input. *Bottom:* given the prompt "turn the camera to a top-down view", Qwen-Edit, Nano-Banana-Pro, and a multi-angle Qwen-Edit fine-tune each return a wrong camera angle and drift in object orientation. The picture is consistent across many scenes.

- **Step 2: Scene anchor and rendering radius.** We need a center $\mathbf{o}$ for the overhead camera to look at and a radius R that sets its framing. Both are estimated coarse-to-fine so distant outliers cannot drag either off the scene.
  For the coarse center we take the least-squares intersection of the input camera rays,

$$\mathbf{q}_0 = \arg\min_{\mathbf{q}} \sum_t \left\| (\mathbf{I} - \mathbf{d}_t \mathbf{d}_t^\top)(\mathbf{q} - \mathbf{o}_t) \right\|_2^2,$$

  where $\mathbf{o}_t \in \mathbb{R}^3$ and $\mathbf{d}_t \in \mathbb{R}^3$ are the origin and unit direction of ray t, $\mathbf{I}$ is the 3×3 identity, and $\mathbf{I} - \mathbf{d}_t \mathbf{d}_t^\top$ is the projection onto the plane orthogonal to $\mathbf{d}_t$. The minimizer $\mathbf{q}_0$ is the approximate point of common visual focus. When camera poses are missing we fall back to the confidence-weighted centroid of the cloud.
  To separate foreground from background, we sort point distances to $\mathbf{q}_0$ and look for the first large gap relative to the median adjacent difference. The radius before that gap is taken as the foreground extent and the rest is discarded as distant background. We then refine the center to the confidence-weighted mean over the foreground subset $\mathcal{F}$ and set R to a fixed percentile of distances from $\mathcal{F}$ to the refined center. Using a percentile rather than the maximum keeps framing stable across scenes by ignoring the tail of weakly reconstructed points.
- **Step 3: Overhead camera frame.** We recover the top-down direction by fitting the dominant horizontal plane to the cloud with RANSAC (PCA fallback on degenerate clouds). When camera

poses are available we sharpen the fit with two pose-aware constraints: the candidate normal must roughly align with the average camera up direction, and most camera centers must lie on the same side of the plane. Together these push the fit toward the actual support surface and reject vertical structures such as walls. The normal's sign is resolved by flipping it toward the side that contains more cameras.

With the oriented normal $\mathbf{n}$ we place the overhead camera at $\mathbf{t}_{\text{top}} = \mathbf{o} + h\mathbf{n}$ with forward direction $-\mathbf{n}$ and height $h = sR$, optionally clamped so the camera clears any tall reconstructed geometry. We align its right axis with the reference camera when one is available, so all rendered top-downs share a consistent orientation across scenes; otherwise we derive the right axis from a canonical world up vector. Camera intrinsics are a standard pinhole, with the field of view widened on square outputs to avoid clipping the source aspect ratio.

- **Step 4: Rendering and photographic texturing.** We project the surviving points into the overhead camera using standard pinhole projection, drop points behind the camera, and resolve per-pixel collisions by keeping the nearest hit. The result is geometrically faithful but visually sparse, so it serves as structural conditioning rather than as the final image. As a cheap consistency check, we also render the point cloud from the original reference camera pose and discard the scene if the cloud and the pose disagree there, before paying the cost of texturing.

  A LoRA-fine-tuned Qwen-Image-Edit-2511 [92] turns the sparse rendering into a photographic top-down. We feed the sparse rendering as the image-conditioning input and train the LoRA [36] to reproduce a real overhead photograph of the same scene. The training set is a hand-collected corpus of 25 scene groups, each containing the multi-view inputs, the reconstructed point cloud, and a real overhead photograph, used as (sparse rendering, photograph) pairs. A narrow rank-32 LoRA combined with heavy resampling lets the model pick up our rendering style without erasing its pretraining. Full hyperparameters are in Table 9. At inference we feed the sparse rendering with a fixed top-down instruction prompt and read off the photographic image.

Table 9: LoRA training configuration for the photographic texturing model.

Setting	Value
Base model	Qwen-Image-Edit-2511 DiT (trainable); Qwen-Image text encoder and VAE (frozen)
Adapter	LoRA, rank 32
LoRA target modules (DiT)	attention q/k/v projections (including cross-modal variants for text–image fusion), output projections, MLP layers, modulation layers
Precision	bf16 with gradient checkpointing
Learning rate	peak 10^{-4}, cosine decay to 10^{-6}, 100-step warmup
Schedule	20 epochs; each example resampled 20 times per epoch
Input resolution	up to 1024^2 pixels
Hardware	4 GPUs, data-parallel

For each scene we save the photographic top-down, the sparse rendering as a structural reference, the colored point cloud, and a metadata file with the refined center, rendering radius, plane normal, and camera intrinsics and extrinsics, sufficient to project 3D annotations onto the top-down image without rerunning reconstruction.

Ego–exo perspective transfer (Ego-Exo4D). Ego-Exo4D [29] records every activity simultaneously with a head-mounted Aria camera and several stationary GoPro cameras, so the photographer's first-person view is paired frame-for-frame with several third-person views of the same scene. Given a third-person photograph in which the photographer is visible, the answer to "what is to your left from this person's perspective?" reduces to a direct ego-frame fact.

We convert each Ego-Exo4D take into multiple-choice perspective-taking questions. The third-person image is the model's input, and the photographer's ego frame is attached as a reasoning aid. The third-person image used as input is unmarked. We do draw a green marker on the photographer's projected head, but only on a parallel copy of the same exo frame, kept in metadata for reviewer inspection. Marking the input would let any model identify the photographer trivially from color and bypass the perspective-taking signal.

The pipeline runs in three steps. First, a per-take nine-pack exports three ego frames, three exo frames, and three exo-marked frames. Then a two-stage GPT pipeline writes one or two perspective-taking

questions per frame. A deterministic post-processor then selects a balanced subset across takes and answer axes.

- **Step 1: Nine-pack construction.** For every take we select three ego frames that jointly maximize in-focus quality and visual diversity. Per-frame sharpness is the variance of the Laplacian on a central $\frac{3}{4} \times \frac{3}{4}$ crop of the rectified ego frame. Visual diversity is one minus the maximum cosine similarity between 96×96 z-normalized appearance vectors. Picks are made greedily by maximizing $0.45 \cdot \mathrm{sharpness} + 0.55 \cdot \mathrm{novelty}$ subject to a minimum temporal gap between selected frames (default 120 frames at 30 fps), which prevents the top-k from collapsing onto a single sub-second cluster on temporally redundant activity videos.

  The Aria RGB stream is fisheye and is rectified to a regular perspective view before scoring. When the official Project Aria calibration is available we use it directly; otherwise we estimate fisheye disc parameters from the first frame and resample onto a perspective grid. The exporter runs the same rectification at two resolutions: a smaller one for frame scoring and a larger one for the on-disk image.

  Drawing the marker on the third-person frame requires projecting the photographer's 3D world position into the exo image at the selected ego frame. Using the per-take time synchronization, we map each ego-frame index to a trajectory time and read the device world position $\mathbf{p}_{\mathrm{world}}$ off the closed-loop trajectory. We then transform $\mathbf{p}_{\mathrm{world}}$ into the exo camera frame as $\mathbf{p}_{\mathrm{cam}} = \mathbf{R}_{\mathrm{cam,world}}(\mathbf{p}_{\mathrm{world}} - \mathbf{t}_{\mathrm{world,cam}})$ with the GoPro extrinsics, and project it through the Kannala–Brandt fisheye model. Writing $\mathbf{p}_{\mathrm{cam}} = (x, y, z)$, the model maps it to image coordinates (u, v) through

$$a = x/z, \quad b = y/z, \quad r = \sqrt{a^2 + b^2}, \quad \theta = \arctan r,$$

$$\theta_d = \theta + k_0\theta^3 + k_1\theta^5 + k_2\theta^7 + k_3\theta^9,$$

$$(u, v) = \big(f_x \cdot (\theta_d/r) \cdot a + c_x,\ f_y \cdot (\theta_d/r) \cdot b + c_y\big),$$

  where (f_x, f_y), (c_x, c_y), and $\{k_0, \ldots, k_3\}$ are the per-camera intrinsics. A point projects to a marker only when it lies in front of the camera ($z > 0$) and (u, v) falls inside the exo frame. Otherwise no marker is drawn and the frame is exported with the marker layer suppressed. Takes with missing or invalid timesync, trajectory, or calibration files are skipped by default.
- **Step 2: Two-stage VQA generation.** From each frame in the nine-pack we generate up to two perspective-taking questions. The single-object axis question picks one visible object near a single edge of the photographer's view, with the answer drawn from {left, right, above, below}. The object-pair relation question picks a target and a reference, with the answer drawn from {left, right, above, below, front, behind}. Both branches use a two-stage GPT pipeline that routes images so that the spatial fact and the question wording are decided in different stages from different views.

  In the candidate stage the model sees only the ego image and proposes the spatial fact: a target object (and, for the pair branch, a reference) plus the chosen direction label. Question wording is forbidden at this stage. In the finalize stage the model sees only the exo image with the marker on the photographer and rewrites the candidate fact into a perspective-taking question, naming the photographer naturally by clothing, role, or position, picking an opener from a small bank of templates, and forcing the either-or option form. The finalize stage never sees the ego image, so it cannot redo the spatial reasoning, only re-phrase it.

  This split removes two failure modes we saw when a single prompt did both: the model invented spatial answers to match the marker rather than the ego frame, and the question stem leaked the answer through phrases like "the person is on the left, so X must be on his left". With the split, the spatial fact is fixed by the candidate stage and the finalize stage is constrained to keep the same target and chosen axis. Each candidate's chosen axis fixes a binary either-or option pair (left/right, above/below, or front/behind), which determines the gold letter directly.
- **Step 3: Balanced subset selection.** The raw corpus is naturally imbalanced. Single-object axis questions skew toward whichever direction the activity faces in each take, and some takes contribute many more frames than others. A deterministic post-processor produces a balanced subset by greedy selection.

  We first reduce to at most one row per (take, frame) by random selection between the single-object and pair candidates of the same frame, so no frame contributes twice. We then pick rows in two phases. Phase 1 walks the takes in shuffled order and adds, from each take, the row whose addition

would most reduce the imbalance across the three axis pairs (left vs. right, above vs. below, front vs. behind), with under-represented pair totals and smaller absolute counts as tie-breakers. This yields at most one row per take in an order that keeps the answer distribution close to per-axis balanced. If the target subset size is larger than the number of distinct takes, Phase 2 pulls from the remaining pool with the same scoring rule but without the per-take constraint. The end result is a subset with near 50/50 splits on each axis pair and a maximally diverse take base for the requested size.

Alternate-view rendering (SAM3D). When only a single image is available, we cannot rely on a paired second camera. Instead we lift the image to 3D, pick any directional entity in the scene (a person, a chair, a TV) as the viewer, and render the rest of the scene from the viewer's inferred pose. The supervision target is a question like "from the chair's point of view, is the lamp to the left or right of the table?". Every gold answer is computed analytically from 3D vector geometry. The language model, when invoked at all, only plays validator and natural-language polisher.

The construction emits three layered reasoning images per question: a who-is-who image showing role-colored masks on the original image, an orientation image showing the viewer's basis pasted at the viewer's centroid, and the viewer-perspective render with the viewer hidden. A model trained to consume them works through them as an interleaved visual chain of thought during reasoning. The pipeline runs in four steps. Before describing them we note the front-end choice.

- **Detection and segmentation front end.** Before settling on RAM++ [117], SAM3 [7], and SAM3D-Objects [68] we tried several alternatives. Combining RAM++ with Grounding DINO or Florence-2 produced boxes that were too loose for clean single-object splats. SAM3 alone gave high mask quality when it fired, but recall was too low: many RAM++ tags never produced a usable instance, which blocked triplet discovery. A hybrid that piped RAM++ through WeDetect into SAM3 improved coverage somewhat, but WeDetect boxes were often clipped at the object boundary, and those truncated masks propagated into incomplete 3D geometry. The RAM++ $\rightarrow$ SAM3 $\rightarrow$ OriAnyV2 [89] stack we ship traded a small loss in vocabulary recall for tighter masks and reliable orientation, which matters more for downstream rendering.
- **Step 1: 2D perception.** For each input image we run RAM++ to produce a list of open-vocabulary tags, then call SAM3 once per tag with the tag as the segmentation prompt to obtain per-instance masks at the original image resolution. A tag-aware non-maximum suppression removes both within-tag duplicates (a single chair detected many times) and tag-internal hierarchical duplicates (a sofa mask containing its cushions). Any mask whose bounding box covers more than 60% of the image is dropped, since these are almost always background panels (walls, floors, sky) that the tagger picked up.

  For every surviving instance we predict its 3D orientation with OriAnyV2, which returns the object's canonical-front azimuth, elevation, and roll relative to the camera frame, together with a binary directional-front flag. Only instances flagged as directional (TVs, sofas, beds) can serve as viewers whose forward direction is read off the predicted orientation; non-directional instances (spheres, bowls, lamps) either play the target role or, if used as the viewer, fall back to a look-toward-the-targets view direction.
- **Step 2: Per-instance 3D reconstruction.** Each (image, instance mask) pair is reconstructed independently into a 3D Gaussian splat by SAM3D-Objects. For this branch of the corpus we use MS COCO 2017 [52] (train and val splits) as the only image source, so reconstruction statistics stay tied to a single domain. The model outputs a per-object Gaussian set in a local frame together with a rigid-plus-isotropic transform $(\mathbf{R}, \mathbf{t}, s)$ that takes the object into a shared canonical frame. We compose multiple objects into a scene by applying these transforms to the per-Gaussian centers, rotating each Gaussian's covariance quaternion accordingly, and concatenating the Gaussian buffers. Per-object scales are clamped to be at least $1.1\times$ the merged kernel size to absorb mild scale inconsistency between independent reconstructions. After composition we apply a fixed axis-swap to align the scene with a $+Y$-up convention and re-normalize so the active Gaussians' largest extent is unit length. Reconstructing per-instance rather than fitting a whole scene at once keeps each splat small and decouples reconstruction failures across unrelated regions, at the cost of mild per-object scale drift that the clamping and re-normalization absorb.
- **Step 3: Viewer-perspective rendering.** For each candidate viewer–target triplet (one viewer plus the two targets that will appear in the question) we place a virtual camera at the viewer's center, looking along the viewer's forward direction, and render the remaining objects through

that camera. For directional viewers the forward direction is the negative of the first column of the predicted orientation matrix (the camera looks out of the object's front); for non-directional viewers it is a look-at vector toward the centroid of the targets.

We jointly fit the camera position and field of view so that the targets fit with a small margin and the viewer does not clip the camera. A small ladder of backoff distances (multiples of the scene diagonal) and angular widths is swept, and we accept the smallest pair $(\text{backoff}, \text{fov})$ for which every target AABB corner $\mathbf{v} = \mathbf{p} - \mathbf{c}_{\text{cam}}$ satisfies

$$\text{atan2}\big(|\mathbf{v} \cdot \mathbf{r}|,\ \mathbf{v} \cdot \mathbf{f}\big) < \text{fov}/2 \quad \text{and} \quad \mathbf{v} \cdot \mathbf{f} > 0,$$

where $\mathbf{f}$ is the camera forward axis and $\mathbf{r} = \mathbf{f} \times \mathbf{u}$ its right axis built from the global up vector. The smallest fitting pair preserves the illusion that the camera sits at the viewer object. Larger backoffs and wider angular extents are only allowed when no smaller pair fits.

We hide the viewer in the render by zeroing the opacity of its Gaussians, so the viewer disappears from the image while the rest of the buffer stays intact.

Two sanity checks finalize the candidate. Each target's AABB must project to a non-empty bounding box in the rendered view (i.e., lie in front of the camera and inside the image), and the two target bboxes must not overlap in render coordinates, since overlap means the question is not visually answerable from the render alone. Candidates failing either check are dropped.

- **Step 4: Question synthesis.** For each accepted candidate we emit one or two multiple-choice questions from a small set of deterministic templates. The current templates ask (i) whether one target is to the left or right of the other from the viewer's standpoint, and (ii) which of the two targets the viewer is facing more toward. Both gold answers are computed analytically from the camera basis $(\mathbf{f}, \mathbf{r})$ and the relative vectors $\mathbf{rel}_X = \mathbf{c}_X - \mathbf{c}_{\text{viewer}}$.

  For the left/right question we project each target's offset onto the viewer's right axis,

  $$x_a = \mathbf{rel}_a \cdot \mathbf{r}, \quad x_b = \mathbf{rel}_b \cdot \mathbf{r},$$

  and pick "left" iff $x_a < x_b$, subject to a minimum normalized margin $|x_a - x_b|/(\|\mathbf{rel}_a\| + \|\mathbf{rel}_b\|) \geq \delta_{\text{LR}}$ so close-call configurations are rejected. For the facing question we measure each target's angular offset from the viewer's forward axis,

  $$\text{angle}_X = \arccos\big((\mathbf{rel}_X \cdot \mathbf{f})/\|\mathbf{rel}_X\|\big),$$

  and pick whichever target has the smaller angle, subject to a minimum angular margin between the two angles. Rejecting close-call candidates makes the gold answers safe to use as supervision, since a small reconstruction error nudging the two targets in opposite directions cannot silently flip the answer.

  A vision-language model is invoked twice, optionally. It can validate a candidate by viewing the three reasoning artefacts produced in Step 3 and either accepting it or rejecting it as visually ambiguous, under a system prompt that asks for conservative rejection. It can also polish the question stem into more natural language while the gold answer and option pair stay frozen. The model is never given authority over the answer.

  Object phrasings in the stem are disambiguated automatically. Viewer choice is restricted to instances the stem can refer to unambiguously, which means singletons of their tag, instances with a unique color within their tag group, or strictly leftmost or rightmost in bbox order. When multiple instances of the same tag remain, the chosen one is referred to as left, right, leftmost, rightmost, or middle by its centroid-x rank within the group. The stem therefore stays unambiguous even when several chairs sit in the same scene.

D.4.2 Object manipulation

The remaining family of mental world modeling keeps the camera fixed and transforms the object itself. We supervise the simplest case: rigid rotation of the object about a vertical axis, which the model must imagine and produce as a second view of the same object from another angle.

Object rotation (Viewpoint-100K). The model is given one image of an object and the supervision target is a second view of the same object after a horizontal rotation, with mostly the same subject framing. The supervision comes from Viewpoint-100K [112], which provides one hundred thousand calibrated pairs of multi-view frames of the same physical object, built from MVImgNet [109] (roughly 6.5M real multi-view frames with object masks, intrinsics, extrinsics, depth, and point clouds per subject). For each sample, two views of the same instance are drawn with horizontal separation between 20° and 100°, over 10,813 distinct object identities across 205 classes.

We repackage those pairs as edit-the-input examples. The first frame is the image to be transformed, the second is the supervision target or auxiliary visual thought, and the text prompt describes the relative rotation in natural language rather than exposing raw calibration tuples. The geometric relationship between the pair is unchanged from the public release; only the prompt and field layout are adapted to our trace format.

E Instruct-180K Examples

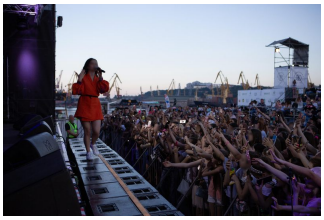
We show Text–Image–Text traces for each Instruct-180K task, organized by cognitive level: one per basic visual perception task and two per spatial task. Each card pairs the input scene with its rendered intermediate image; Thought 0 (under the input) shows the model's plan before generation, Thought 1 (under the intermediate) describes the synthesized view. The question stem and gold answer appear at the bottom of each card.

Level 1 (basic visual perception)

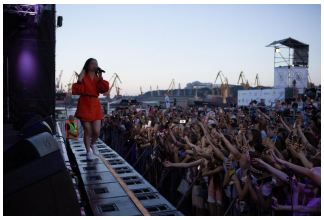
Tasks at this level test direct reading of single-image content—locating a target among distractors, or recovering a global layout from shuffled fragments.

Visual search (L1)

Visual search (L1) • *Example 1* bbox variant



INPUT



INTERMEDIATE IMAGE

THOUGHT 0 — PLAN

To answer the question, first look at the left side of the image where the stage structure is located. The performer in red is standing on a narrow platform along the front of the stage, and the question asks about another person standing near the stage on that same left side. Scan downward and slightly behind the performer, close to the dark stage wall and speaker equipment. There is a partially visible person positioned low on the left side near the barrier area. This person is wearing a high-visibility vest that stands out sharply against the darker stage surroundings. That bright vest is the key item that should be enclosed by the red bounding box.

THOUGHT 1 — INTERMEDIATE VIEW

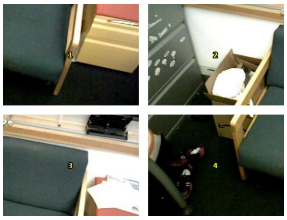
The highlighted region isolates the vest worn by the person standing near the stage on the left. Its color is a vivid fluorescent yellow-green, the kind commonly used for security or event staff, and it clearly does not appear dark blue, red, or white. The strong neon tone makes the correct choice A.

Q. What is the main color of the vest worn by the person standing near the stage on the left side of the image? (A) Bright yellow-green (B) Dark blue (C) Red (D) White Answer with the option's letter from the given choices directly.

A. A

Jigsaw QA (L1)

Jigsaw QA (L1) • *Example 1* in-place



INPUT



INTERMEDIATE IMAGE

THOUGHT 0 — PLAN

I start by identifying the main objects visible across the four scrambled quadrants. The image appears to be an indoor office or waiting-room corner viewed from above. The major elements are a blue upholstered chair, a light-wood cabinet or side table with papers on top, a cardboard box containing white circular items, a dark filing cabinet at the left, and dark carpet on the floor. There is also a windowsill or ledge running horizontally near the top of the scene. Piece 2 contains the dark filing cabinet on its left side, the white wall behind, the cardboard box near the center-right, and the left arm of the blue chair at the far right edge. This piece clearly belongs toward the upper-left region of the final image, because the filing cabinet looks like a left boundary object and the ledge runs along its top edge. Piece 3 shows the top of the blue chair occupying much of the left and center area, with the same horizontal ledge along the top. On the right side of this piece there is the light-wood cabinet with papers on top. This makes Piece 3 a strong candidate for the upper-right …

THOUGHT 1 — INTERMEDIATE VIEW

Rechecking the assembled image as a whole, the scene now reads as a single continuous overhead view of an office corner, which is exactly the kind of coherence I want from a correct reconstruction. The most prominent object is the blue upholstered chair placed centrally. Its backrest and seat align smoothly with no break in contour or texture, and the wooden arms descend naturally on both sides. That already gives a strong structural anchor for the arrangement. Along the top edge, the horizontal windowsill or ledge runs cleanly from left to right. On the left side of this upper band sits the dark filing cabinet, and toward the right side there is a stack of dark trays or papers. The ledge maintains a consistent height and perspective across the full width, which is a very important sign that the top row has been assembled properly. On the left side of the chair, the cardboard box fits convincingly into the narrow space between the chair and the filing cabinet. Its top opening and the white circular contents sit at a plausible angle relative to the wall and floor. The box does not …

Q. The image below is divided into four parts by white strips, forming a 2×2 jigsaw puzzle. The parts are labeled "1" (top-left), "2" (top-right), "3" (bottom-left), and "4" (bottom-right). These parts are from a single original image but have been shuffled. Your task is to determine the correct arrangement of the physically labeled parts to reconstruct the natural image. Select the correct statement from the following choices: (A) The top-left part should be Part 2; the top-right part should be Part 1; the bottom-left part should be Part 4; and the bottom-right part should be Part 3. (B) The …

A. C

Level 2 (static spatial understanding)

Tasks at this level reason over spatial relationships within the observed scene—object counts and relative depths between visible regions.

Spatial counting (L2)

Spatial counting (L2) • *Example 1* LVIS



INPUT



INTERMEDIATE IMAGE

THOUGHT 0 — PLAN

The street scene contains several large vehicles, and some buses are partially hidden behind the sightseeing bus and by the left edge of the frame. To avoid missing small or occluded bus portions, I will generate a mask overlay that isolates only the bus instances against a plain background, making each separate bus silhouette easier to count accurately.

THOUGHT 1 — INTERMEDIATE VIEW

The mask overlay cleanly separates the buses from the rest of the scene. I can identify one small partial bus at the far left edge, a second red double-decker just to its right, the large sightseeing bus in the center, a fourth bus segment visible behind the right side of the sightseeing bus, and a fifth bus at the far right edge. Counting these distinct isolated instances gives a total of 5 buses visible.

Q. What is the total number of buses visible in this image?

A. 5

Spatial counting (L2) • *Example 2* LVIS



INPUT



INTERMEDIATE IMAGE

THOUGHT 0 — PLAN

The showroom scene contains several wall-mounted screens arranged in a row, and some are smaller or farther back, which makes it easy to miss one when counting directly from the cluttered display. To avoid confusing televisions with the surrounding kiosks and signage, I will generate a mask overlay that isolates only the television sets on a plain background so each distinct screen can be counted cleanly.

THOUGHT 1 — INTERMEDIATE VIEW

The mask overlay isolates the televisions as separate dark-outlined screen shapes on a white background. Counting each distinct isolated set from left to right gives six televisions in total, including the small screens toward the back on the right. Therefore, the number of visible television sets is 6.

Q. What is the number of television sets visible in this image?

A. 6

Depth reasoning (L2)

Depth (L2) • *Example 1* RefSpatial



INPUT



DEPTH MAP

THOUGHT 0 — PLAN

I need to determine the front-back relationship between the grey knit dress in the center area and the matte black jacket on the left. From the original photo, both adults overlap slightly, so I will generate a depth-support view next to compare which person is closer to the camera based on their relative depth shading.

THOUGHT 1 — INTERMEDIATE ANALYSIS

The generated depth-support view shows the left person in the matte black jacket mostly in cooler teal tones, while the grey knit dress wearer appears in relatively warmer green tones over the torso and lower body. In this depth visualization, the warmer tones indicate being closer to the camera than the cooler tones. This means the grey knit dress is positioned in front of, not behind, the matte black jacket. Therefore the correct choice is (B) no.

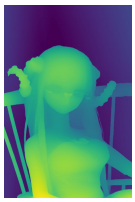
Q. Is grey knit dress at center behind matte black jacket at left? Options: (A) yes (B) no

A. B

Depth (L2) • *Example 2* RefSpatial



INPUT



DEPTH MAP

THOUGHT 0 — PLAN

I need to compare which candidate eye location is farther from the camera relative to the black velvet dress at the center. From the original photo, the dress sits on the torso in the foreground while the two blue eyes are near the center of the head. To judge the depth difference reliably, I will generate a depth-support view of the scene so I can compare the relative distance of the central dress region against the two candidate eye regions.

THOUGHT 1 — INTERMEDIATE ANALYSIS

The generated depth-support view uses color to encode distance: the dress at center appears in a warmer green-yellow range, while the eye region near the center is slightly cooler/greener, indicating it is farther back than the dress. Comparing the two candidate eye points, the blue glossy eye at center is the farther one relative to the black velvet dress. Therefore the correct choice is (A) blue glossy eye at center.

Q. Which is farther from the black velvet dress at center? Options: (A) blue glossy eye at center (B) blue glass eye at center

A. A

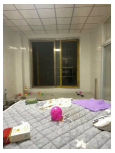
Level 3 (mental world modeling)

Tasks at this level construct a mental scene model that extrapolates beyond the observed input—novel viewpoints, top-down maps, ego/exocentric switches, and rigid-body rotations.

BEV synthesis (L3)

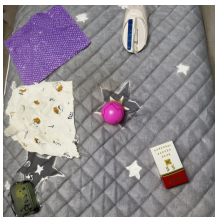
MindCube BEV (L3) • *Example 1* MindCube/behind-me


FRONT VIEW


LEFT VIEW


BACK VIEW


RIGHT VIEW


BEV (INITIAL)


BEV (TURNED AROUND)

THOUGHT 0 — PLAN

In the queried view, the camera is facing the cardboard-covered sliding door. To determine what is behind that viewpoint, I first need a bird's-eye view centered on the pink ball so the arrangement of the nearby reference objects on the bed becomes stable and the facing direction in the queried scene can be tracked without perspective distortion.

THOUGHT 1 — AFTER FIRST INTERMEDIATE

The first bird's-eye view shows the pink ball with the purple cloth, patterned cloth, white iron, green device, and red-white box laid out clearly around it. Matching those bed items to the queried scene fixes the current orientation: from image 3, the camera is looking toward the side of the bed where the iron appears beyond the ball and the red-white box sits to the right. Since the question asks what is behind me, I keep the same spot and rotate the top-down layout by 180 degrees to represent turning around.

THOUGHT 2 — AFTER SECOND INTERMEDIATE

The transformed bird's-eye view is the turned-around orientation, and it matches the arrangement seen in the view with the vent hood and flexible duct above the bed. That means the direction in front of that matching original view is exactly what lies behind the observer in image 3. In that matched room view, the prominent object ahead is the smoking machine, so the answer is B.

Q. Based on these four images (image 1, 2, 3, and 4) showing the pink ball from different viewpoints, with each camera aligned with room walls and partially capturing the surroundings: If I am standing at the same spot and facing the same direction as shown in image 3, what is behind me? A. Cardboard-covered sliding door B. Smoking machine C. Wall D. Window

A. B

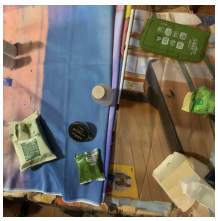
MindCube BEV (L3) • *Example 2* MindCube/directional-query


FRONT VIEW


LEFT VIEW


BACK VIEW


RIGHT VIEW


BEV (INITIAL)


BEV (ROTATED CW)

THOUGHT 0 — PLAN

I first inspect the queried view, the one with the red bed frame behind the powder bottle. The question asks what is to the left of the bottle from that specific viewpoint, so I need to convert this egocentric left/right relation into the room layout. To do that, I will generate a bird's-eye view of the tabletop and nearby surroundings from the queried viewpoint, because a top-down layout makes it easier to track how the bottle and surrounding scene rotate across the different camera directions.

THOUGHT 1 — AFTER FIRST INTERMEDIATE

The first generated bird's-eye view shows the powder bottle near the center-right of the blue mat, with the black round container and green packet on one side, and the tissue pack and green rectangular pouch on the other. This top-down layout clarifies where objects sit around the bottle independent of perspective distortion. Since the question is asking for what is to the left in the queried view, I now rotate this overhead layout by 90 degrees so that the queried camera's left direction becomes a forward-looking direction in an aligned comparison view. That rotated top-down view will let me match the orientation to one of the original normal-perspective images.

THOUGHT 2 — AFTER SECOND INTERMEDIATE

In the rotated bird's-eye view, the bottle's aligned forward direction matches the original viewpoint that looks out toward the open room with the table and chairs in the distance. That is the same orientation as the image showing the dining area beyond the tabletop. In that matched original view, the space directly ahead of the bottle is the dining area, so translating back to the queried red-bed-frame viewpoint means the bottle's left side corresponds to the dining area. Therefore the correct choice is A. Dining area.

Q. Based on these four images (image 1, 2, 3, and 4) showing the powder bottle from different viewpoints, with each camera aligned with room walls and partially capturing the surroundings: From the viewpoint presented in image 3, what is to the left of the powder bottle? A. Dining area B. Black cabinet C. Red bed frame with decorative ironwork D. White wall and windows

A. A

Panorama (L3)

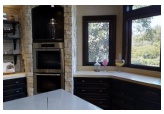
MMSI stitching (L3) • *Example 1* MMSI v11


LEFT EXTRA 4


LEFT VIEW


RIGHT VIEW


LEFT EXTRA 3


LEFT EXTRA 2


COMPOSITE

THOUGHT 0 — PLAN

Across the kitchen views, the stainless steel double-door refrigerator is clearly visible in the kitchen wall cabinetry, especially in the refrigerator-facing views. The brown upholstered armchair appears through the opening into the adjacent sitting area, visible in the room-opening views. Since these two objects are seen in different directions from the same general camera position, I will generate a merged panoramic view that combines the refrigerator side and the sitting-room opening so both objects can be placed in one continuous layout and their left/right plus front/back relationship can be resolved.

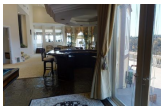
THOUGHT 1 — AFTER COMPOSITE

In the merged panoramic view, the stainless steel double-door refrigerator sits near the center of the panorama, while the brown upholstered armchair appears to the right side through the opening into the adjacent room. The armchair is also farther from the camera than the refrigerator: the refrigerator is in the nearer kitchen plane, while the chair sits deeper into the next room beyond the doorway. Combining these cues, the armchair is positioned to the right and behind the refrigerator. Therefore, the brown upholstered armchair is to the back-right of the stainless steel double-door refrigerator.

Q. Both the stainless steel double-door refrigerator and the brown upholstered armchair are visible in the room. Where is the brown upholstered armchair relative to the stainless steel double-door refrigerator? Options: A: front left, B: to the back-right, C: directly left, D: directly behind

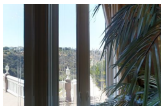
A. B

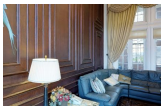
MMSI stitching (L3) • *Example 2* MMSI v11


LEFT EXTRA 3


RIGHT VIEW


LEFT EXTRA 1


LEFT EXTRA 2


LEFT VIEW


COMPOSITE

THOUGHT 0 — PLAN

Photo 2 shows the pool table and the surrounding wall with doorways; the question says this photo was taken facing east, so rightward in that photo corresponds to east. The blue sectional sofa appears in Photo 5 along the wood-paneled wall and windows, which seems to be part of the same room but outside the field of view of Photo 2. To determine the sofa's direction from the camera position of Photo 2, I will generate a merged panoramic view that stitches the pool-table view with the sofa view so both landmarks appear in one continuous layout.

THOUGHT 1 — AFTER COMPOSITE

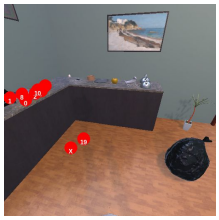
In the merged panoramic view, the blue sectional sofa is positioned to the left of the pool table. Since the panorama connects the sofa corner and the pool-table view from the same room, this shows that when standing where Photo 2 was taken and facing east, the sofa lies to the left of that facing direction. With east defined as right in Photo 2, left corresponds to north. Therefore the blue sectional sofa was north.

Q. When I took Photo 2 facing east, in which direction was the blue sectional sofa that appears in Photo 5? Assume east in Photo 2 is to the right when answering. Options: A: north, B: east, C: southwest, D: west

A. A

Alternate-view rendering (L3)

SAM3D viewpoint (L3) • *Example 1* SAT-perspective


INPUT


INTERMEDIATE IMAGE

THOUGHT 0 — PLAN

From the room view, the standing point "X" is on the floor near the inner corner of the L-shaped counter. The toaster is identified as the object near marker 1, which sits on the left run of the counter. To determine where the toaster ends up relative to the new facing direction after a 90° left rotation, I need to generate an egocentric (first-person) view from point X after performing that left turn, since left/right judgments depend on the viewer's orientation.

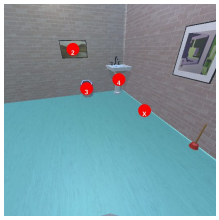
THOUGHT 1 — INTERMEDIATE VIEW

In the generated egocentric view from X after turning left, the toaster is directly ahead on the countertop, occupying the central forward area of the scene rather than appearing off to the right side. This indicates that, relative to the new facing direction, the toaster is approximately straight in front of you. Therefore the correct option is A.

Q. Standing at the spot labeled "X" in the image, imagine you rotate 90° to your left. Relative to your new facing direction, where would the Toaster (near marker 1) be located? Options: A: approximately straight in front of you, B: on your right

A. A

SAM3D viewpoint (L3) • *Example 2* SAT-perspective



INPUT



INTERMEDIATE IMAGE

THOUGHT 0 — PLAN

In the room view, the standing point “X” is near the right wall/corner. The painting labeled “2” is on the far wall to the left side of the scene, so from X it’s at a diagonal across the corner rather than directly in front. To judge whether a 90° left rotation makes painting 2 closer or farther, I need to generate an egocentric (first-person) view from X after turning left, because rotation changes which wall you face and whether the painting becomes more directly aligned (shorter straight-line distance) or more off-angle.

THOUGHT 1 — INTERMEDIATE VIEW

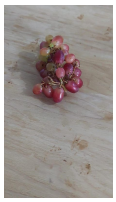
In the generated egocentric view from X after turning 90° left, the wall with the landscape painting comes into the forward-facing view on the right side, indicating the rotation reorients the viewer toward that wall instead of looking across the corner. With the painting now more directly in the facing direction (less around-the-corner/diagonal), its effective straight-line distance from the viewer is reduced compared to the original orientation. Therefore, the painting near marker 2 becomes closer (Option B).

Q. Standing at the spot labeled "X" in the picture, if you rotate 90° to your left, does the painting near marker 2 become closer or farther away? Options: A: Further, B: Closer

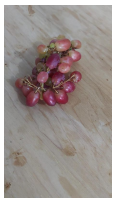
A. B

Object rotation (L3)

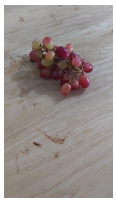
Object rotation (L3) • *Example 1* Viewpoint-100K



INITIAL VIEW



AFTER STEP 1



AFTER STEP 2

THOUGHT 0 — STEP 1

After rotating 33° to the right, the object now looks like this:

THOUGHT 1 — STEP 2

After rotating 80° to the left, the object now looks like this:

Q. Given the initial photo of an object, generate the views after performing the following camera rotations: Step 1: rotate 33° to the right; Step 2: rotate 80° to the left.

A. Step 1: rotate 33° to the right; Step 2: rotate 80° to the left

Object rotation (L3) • *Example 2* Viewpoint-100K



INITIAL VIEW | AFTER STEP 1 | AFTER STEP 2

THOUGHT 0 — STEP 1

After rotating 45° to the right, the object now looks like this:

THOUGHT 1 — STEP 2

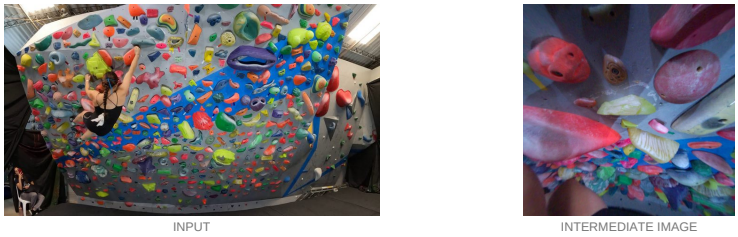
After rotating 99° to the right, the object now looks like this:

Q. Given the initial photo of an object, generate the views after performing the following camera rotations: Step 1: rotate 45° to the right; Step 2: rotate 99° to the right.

A. Step 1: rotate 45° to the right; Step 2: rotate 99° to the right

Ego-exo transfer (L3)

Ego-exo transfer (L3) • *Example 1* Ego-Exo4D



INPUT | INTERMEDIATE IMAGE

THOUGHT 0 — PLAN

In the outside scene, the climber in the black top with the long braid is the person of interest. The two holds named in the question are near the left side of the wall: a large red hold and a smaller brown hold nearby. But the camera is viewing her from behind and below, so the wall's left-right arrangement in the photo does not directly tell me what is left or right from her own viewpoint. To judge that correctly, I need to generate a first-person view from her position and facing direction.

THOUGHT 1 — INTERMEDIATE VIEW

In the generated first-person view, the large red hold appears on the left side of her view, and the small brown hold is positioned to its right. This perspective resolves the ambiguity from the outside camera angle. Therefore, from the climber's perspective, the small brown hold is to the right of the large red hold. Answer: B.

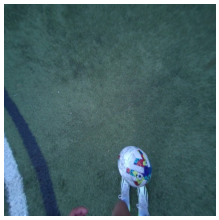
Q. From the perspective of the climber in a black top with a long braid, is the small brown hold to the left or right of the large red hold on the left? A: left, B: right

A. B

Ego-exo transfer (L3) • *Example 2* Ego-Exo4D



INPUT



INTERMEDIATE IMAGE

THOUGHT 0 — PLAN

In the outside scene, the player in the red shirt is the reference person. The white shoe and the soccer ball are both near his feet, but from this side-on camera angle it is hard to judge their vertical relation from his own viewpoint, because the camera is not aligned with where he is looking. To answer the question correctly, I would generate a first-person view from the red-shirted player's position and facing direction so the ball-versus-shoe relation can be checked as he sees it.

THOUGHT 1 — INTERMEDIATE VIEW

The generated first-person view looks downward from the red-shirted player's perspective. In that view, the white shoe appears lower in the frame while the soccer ball is positioned above it. That means from his perspective, the ball is above the white shoe. So the correct answer is A: above.

Q. From the perspective of the soccer player in the red shirt, is the soccer ball above or below the white shoe? A: above, B: below

A. A

F Pretext-270K Examples

We show one representative Text–Image–Text trace per pretext task, taken from the released val splits. Each card pairs the corrupted input with its clean target; Thought 0 (under the input) describes the degraded view, Thought 1 (under the target) describes the restored scene.

Jigsaw (Pretext)

Jigsaw • *Example 1* PixMo-Cap



CORRUPTED INPUT



CLEAN TARGET

THOUGHT 0 — DEGRADED VIEW

The scene is broken into a shuffled grid of rectangular tiles with thin white separator lines, most noticeably a strong horizontal white band across the middle and vertical divisions that chop the utensils into segments, so the picture reads like mismatched pieces of a tabletop arrangement. Even through the disarray, several shiny metal spoons on a warm golden wooden surface are recognizable: some tiles show rounded spoon bowls with bright reflections, while others show long narrow handles with a small circular mark near the end. The wooden background has horizontal grain, but the grain and spoon parts fail to align across neighboring tiles; for example, at the central horizontal seam the top row shows only handle shafts while directly below it the spoon bowls suddenly appear, and along one vertical boundary near the middle a bowl edge meets a handle segment that clearly belongs elsewhere instead of continuing smoothly. Another obvious mismatch appears toward the right side, where adjacent tiles place reflective spoon bowls beside handle tops at inconsistent heights, breaking the …

THOUGHT 1 — RESTORED VIEW

After the tiles snap into the correct positions, six small identical spoons sit neatly in a single horizontal row on a medium light golden wooden surface, their handles pointing downward and their bowls toward the top of the frame. The three spoons on the left have their bowls facing up, while the three on the right are turned over with their bowl backs facing up, making the right half even more mirrorlike; in those reflective surfaces, a faint reflection of the photographer can be seen. The spoons have a rich coppery-gold metallic sheen, and each handle ends with a small circular logo or engraved emblem accented by a reddish round detail, though the fine etching remains hard to make out. They are evenly spaced, and the warm lighting creates bright gleams and soft highlights across the polished metal. Beneath them, the wood grain runs horizontally and becomes more pronounced toward the right, giving a simple, elegant backdrop that emphasizes the uniformity of the arrangement.

Colorization (Pretext)

Colorization • *Example 1* PixMo-Cap



CORRUPTED INPUT



CLEAN TARGET

THOUGHT 0 — DEGRADED VIEW

This appears to be a magazine or match-program cover tightly cropped on a person from the neck to the torso, with the face out of frame and the shirt taking up most of the composition. The whole view is desaturated, so everything reads in shades of gray, but the layout is still clear: large vertical block letters run down the left side spelling out Darlington F.C., a broad dark rectangular sponsor panel sits across the chest with the words Hutchison Telecom in large serif type, and a club crest is printed on the upper right side of the shirt. Smaller text below announces official club sponsors, Barclays League Division Three, a match against Doncaster Rovers, the date Monday 12th April 1993, kickoff at 3.00 p.m., and a note about today's match sponsor, Erection Site Services Ltd. The shirt itself looks light-toned with a subtle geometric pattern, dark trim at the collar and placket, and dark angular shapes near the lower corners, though the exact colors are impossible to tell in this grayscale rendering. I'll restore the natural colors next.

THOUGHT 1 — RESTORED VIEW

With the colors back in place, the cover resolves into a crisp white football shirt with soft cool-grey shading in the folds and a faint pearly sheen, worn by a person whose visible neck is a warm peachy pink. The polo collar and placket are accented with black trim and bright sky-blue edging, and on the upper chest the Darlington crest stands out in red, white, and black: a shield with a pilgrim hat, flanked by white leafy ornament and a ribbon reading The Quakers. Across the middle, a deep blue-violet sponsor panel carries the white words Hutchison Telecom, identifying the official club sponsor for Barclays League Division Three. Running vertically down the left side, large pale blue block letters spell out Darlington F.C. against black rectangular blocks. Beneath the sponsor panel, the printed details announce the match against Doncaster Rovers on Monday, 12th April 1993, with kickoff at 3 p.m., and the cover identifies itself as the official programme while also naming today's match sponsor as Erection Site Services LTD.

Super-resolution (Pretext)

Super-resolution • *Example 1* PixMo-Cap



CORRUPTED INPUT



CLEAN TARGET

THOUGHT 0 — DEGRADED VIEW

A slightly blurry close-up shows a round, clear plastic container filled with chunky salsa on a beige countertop. Even with the soft focus, the mixture is clearly a reddish, liquidy dip packed with diced tomato-like pieces, bits of onion, green herb flecks, and a few pale seeds suspended throughout. The bowl takes up most of the frame, with its translucent rim visible around the top and sides, and a few stray herb fragments sticking to the inner wall. The countertop behind it appears light tan with a brighter area toward the upper right, though the fine texture and the smallest ingredients are hard to distinguish because the image lacks sharp detail. I'll restore the fine detail next.

THOUGHT 1 — RESTORED VIEW

Once the fine detail is restored, the salsa looks vivid and textured inside the clear plastic bowl, with a deep red base crowded by juicy tomato chunks, green pieces resembling green tomatoes, and finely chopped green herbs scattered across the surface. Crisp bits of white onion and some purple onion skin pieces stand out among the vegetables, while tiny whitish seeds are visible throughout the mixture. The transparent container shows the beige countertop faintly through its sides, and a few herbs cling to the bowl's inner wall near the rim. Light from the upper right brightens that corner of the counter and adds clearer highlights along the plastic edge and the glossy surface of the salsa.

Masked inpainting (Pretext)

Masked inpainting • *Example 1* PixMo-Cap



CORRUPTED INPUT



CLEAN TARGET

THOUGHT 0 — DEGRADED VIEW

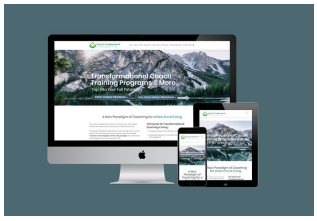
A moody stage-like scene is visible under deep blue and purple lighting, with three figures arranged across the frame. On the left stands a woman in a black outfit with a sleek bob haircut, turned mostly sideways so her back and shoulder are emphasized. In the center-right, another woman in black extends an arm across to a man at the far right who wears a dark shirt and a black fedora and appears to be leaning forward slightly. Much of the central woman's face and upper head are hidden by a large irregular white patch spread across the upper middle of the image, and there is also a rectangular blurred mask covering the man's face. The woman's fishnet-like sleeve and the man's hat brim remain visible around the occlusions, while dim silhouettes of additional people fade into the dark background. A red circular WatchMojo logo with white lettering sits in the lower left corner. Let me now fill in the masked regions.

THOUGHT 1 — RESTORED VIEW

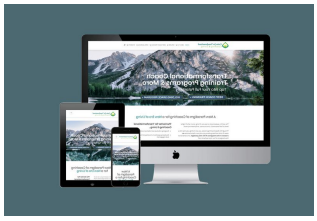
Once the masks are filled in, the full thumbnail resolves into a polished promotional still from "Chicago," with Catherine Zeta-Jones in the foreground placing her hands on the shoulders of a man in a black shirt and fedora who looks downward. To her left is another woman with the same bob haircut in a black outfit, the two women positioned back to back, while the woman on the right faces toward the camera. The dark blue and purple background keeps a theatrical, dramatic mood and partially veils other figures behind them. In the lower left, the WatchMojo logo stands out clearly as a red circle with white trim and white "Mojo" lettering, with the stylized "O.J.O." arranged like a face with eyes and a smile.

Rotation and flip recovery (Pretext)

Orthogonal transform • *Example 1* PixMo-Cap



CORRUPTED INPUT



CLEAN TARGET

THOUGHT 0 — DEGRADED VIEW

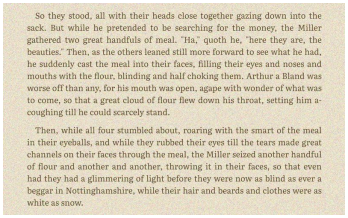
A promotional mockup of a website is shown on three devices against a muted blue-gray background: a large silver Apple desktop monitor dominates the center-left, while a black smartphone stands in front near the lower middle and a black tablet sits to the right, partially overlapping the monitor. All three screens display the same mountain-themed coaching website in a normal readable orientation, with the headline about transformational coach training over a scenic lake and rocky peaks, and blocks of text below. The Apple logo is centered on the monitor's lower bezel, and the tablet and phone both present narrower responsive versions of the same page. Let me now flip the image horizontally.

THOUGHT 1 — RESTORED VIEW

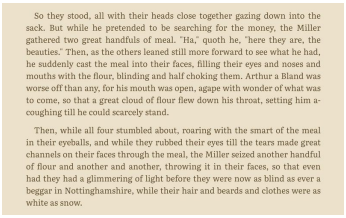
After flipping the image horizontally, the same device mockup appears mirrored left-to-right: the large Apple monitor now sits more toward the center-right, the tablet has moved to the lower left, and the smartphone stands in front near the lower center-left. The website content on every screen is reversed, so the headline and navigation read backward, and the mountain scene is mirrored as well, with the bright sunlit area now appearing on the opposite side. The silver monitor still anchors the composition with the black Apple logo at the bottom center, while the tablet and phone continue to show the responsive layout in front of it.

Denoising (Pretext)

Denoising • *Example 1* PixMo-Cap



CORRUPTED INPUT



CLEAN TARGET

THOUGHT 0 — DEGRADED VIEW

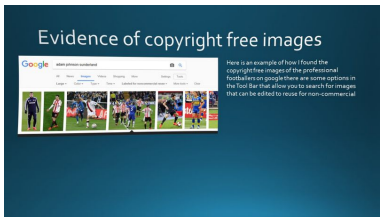
A heavily noise-laden view shows a rectangular page filled with a long block of serif text, the dark letters still mostly readable despite dense multicolored grain scattered across the entire surface. The page sits on a light tan or beige background, with several paragraphs arranged in neat lines and a noticeable indented break near the lower half, but the speckling softens edges and makes some words and punctuation less certain at a glance. Even through the interference, it is clear this is a photographed or scanned passage of writing rather than an illustrated scene, with brownish text spread evenly across the page in a formal book-like layout. With that noted, I can clear the noise from the scene.

THOUGHT 1 — RESTORED VIEW

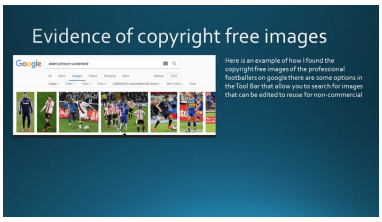
With the noise cleared away, the page becomes crisp and smooth, showing meticulously typed dark brown serif text on a beige, sandy-colored background. The writing is sharply legible across two dense paragraphs, beginning with “So they stood, all with their heads close together, gazing down into the sack,” and continuing with a vivid account of the miller pretending to search for money before flinging meal into the others’ faces. The clean restoration makes the structure easy to follow, including the indented paragraph break and the evenly spaced lines, while the passage goes on to describe Arthur a Bland suffering the worst of it, flour flying down his throat, and then all four men stumbling about half-blinded as more handfuls are thrown until their hair, beards, and clothes are “as white as snow.” The overall impression is of a carefully presented excerpt from a story, its sharp lettering and calm beige page contrasting with the dramatic, deceptive scene described in the text.

Deformation recovery (Pretext)

Deformation recovery • *Example 1* PixMo-Cap



CORRUPTED INPUT



CLEAN TARGET

THOUGHT 0 — DEGRADED VIEW

A presentation slide fills the view with a dark teal to lighter blue gradient background, headed by large white text reading “Evidence of copyright free images.” On the left sits a wide screenshot of a Google image search for “adam johnson sunderland,” and on the right is a block of smaller white explanatory text. Most of the slide is readable, but a localized warp bends the geometry around the lower middle of the screenshot: the bottom edge bows and pinches inward, the row of soccer-player thumbnails kinks around that point, and nearby vertical and horizontal lines no longer stay straight, as if the image has been pushed downward and twisted slightly in that region. The rest of the composition remains roughly intact, with the title steady across the top and the paragraph still legible on the right. I'll unwarp the distorted geometry next.

THOUGHT 1 — RESTORED VIEW

After unwarping the geometry, the slide resolves into a clean, rectangular presentation layout with a smooth gradient background transitioning from dark teal at the top to a lighter shade below. Large white title text across the upper portion reads, “Evidence of copyright free images.” On the left, a neat screenshot of a Google Images search shows the Google logo in the upper left, the search bar containing “adam johnson sunderland,” and beneath it a row of rectangular thumbnails featuring a man in several soccer action poses, wearing blue-toned kits. On the right, a left-aligned block of white text explains that this is an example of finding copyright-free images of professional footballers on Google, noting that toolbar options can be used to search for images that may be edited or reused for non-commercial purposes.

G Training and Evaluation Details

This section documents the training and evaluation setup. The training configuration below produces the ThinkMorph-SSL checkpoint reported in Section 4.3. Both stages fine-tune the same BAGEL-7B-MoT backbone [17] under FSDP2 sharding with the in-house `bagel_fsdp2_trainer`; what differs across stages is the data, the training horizon, the world size, and a few token-budget parameters. Table 10 summarises every hyperparameter. The loss and the verbatim system prompt follow it, compute and runtime numbers are in Table 12, and Section G.2 describes the evaluation.

G.1 Training

Per-stage hyperparameters. Table 10 lists every hyperparameter that differs between Stage 1 (pretext reasoning) and Stage 2 (instruction tuning) alongside the few that are shared. Stage 1 trains Pretext-270K from the BAGEL-7B-MoT checkpoint released by [17] on 32 GPUs across 4 nodes; Stage 2 resumes from the Stage 1 EMA checkpoint at step 5500 on 48 GPUs across 6 nodes. The Stage 1 learning rate is the $\sqrt{2}\times$ scaling of a 2e-5 base when moving from a 16-GPU pilot to the production 32-GPU run; Stage 2 reverts to the unscaled 2e-5. The per-device batch size is fixed at 1 with gradient accumulation 1, so the effective batch size is the world size (32 in Stage 1, 48 in Stage 2). We rely on dynamic sequence packing up to `max_num_tokens` (16,384 in Stage 1, 20,480 in Stage 2) instead of micro-batching, which keeps per-step throughput high under the heterogeneous trace lengths of Instruct-180K.

Ablation runs. The BAGEL-7B ablations in Section 4.4 use the trainer and loss above. Table 11 lists the settings that differ from Table 10.

Full training data. The Stage-2-only run trains Instruct-180K from BAGEL-7B-MoT with the Stage-2 settings of Table 10 on 32 GPUs. The staged run is the main Stage-2 run on 48 GPUs. We compare the Stage-2-only EMA checkpoint at step 3,000 with the staged EMA checkpoint at step 2,000, so both runs have taken about 100K Stage-2 sequences (steps $\times$ world size).

Spatial training subset. All runs use 8 GPUs with per-device batch size 1, so the effective batch size is 8, and no EMA. Stage 1 trains on 50K pretext traces built from the images of the spatial training subset for 10,000 steps. Stage 2 trains on the spatial training subset of Instruct-180K from the Stage-1 checkpoint at step 10,000, and we evaluate every 2,000 steps; the figures call each such interval a Stage-2 epoch. The one-stage run trains on both sets together and samples them in proportion to their size. The instructed variant differs from the staged run only in the Stage-1 inputs. InternVL-U-4B uses the same settings as BAGEL-7B.

Loss and optimisation. The training objective sums a cross-entropy term on text tokens and a mean-squared-error term on the VAE latents that the image-generation head produces; both terms are evaluated at every interleaved trace position the processor labels as supervised. We never freeze the image-generation pathway: at Pretext-270K time the VAE-MSE term teaches the model to reproduce the restored image, and at Instruct-180K time it teaches the model to generate intermediate visual states that the subsequent thoughts can attend to. The VAE encoder itself (`vae_model`) is frozen across both stages so the latent space stays fixed. Both loss terms have weight 1.0, and both stages use the same processor and conditional-dropout settings.

System prompt and special tokens. Both stages share an identical action-control system prompt that declares the three special tokens (`<think>`, `<gen_image>`, `<answer>`) and the inline image-size syntax. We set `wrap_assistant_reasoning_with_think=true` so every reasoning span is wrapped in `<think>...</think>` before any image is emitted, which keeps the trace machine-parseable even when the model emits multiple image-thought rounds. The box below reproduces the prompt verbatim.

Action-control system prompt (verbatim, identical across Stage 1 and Stage 2)

```
Use <think>...</think> for intermediate reasoning or planning.

To generate an image, write <gen_image> followed by an image-size declaration:
  <gen_image>[Image size: (height, width)]
```

Table 10: Per-stage training hyperparameters. Stage 1 trains Pretext-270K from BAGEL-7B-MoT; Stage 2 resumes from the Stage 1 EMA checkpoint at step 5500 and trains Instruct-180K.

Parameter	Stage 1 (Pretext-270K)	Stage 2 (Instruct-180K)
Initialization		
Initial weights	BAGEL-7B-MoT [17]	Stage 1 EMA, step 5500
Distributed setup		
Nodes	4	6
GPUs per node	8	8
World size	32	48
Trainer	`bagel_fsdp2_trainer`	`bagel_fsdp2_trainer`
Sequence-parallel deg.	1	1
Optimisation horizon		
`max_steps`	10,000	12,000
`warmup_steps`	1,000	1,000
Peak learning rate	2.83e-5 ($\sqrt{2}\times$ of 2e-5)	2.0e-5
LR schedule	cosine to zero	cosine to zero
`per_device_train_batch_size`	1	1
`gradient_accumulation_steps`	1	1
Effective batch size	32	48
Sequence packing		
`packing_mode`	dynamic	dynamic
`max_num_tokens`	16,384	20,480
`stride_sharding`	true	true
Adaptive image-size to budget	—	true (margin 1.05, floor 512)
Frozen / trainable modules		
`freeze_modules`	`[vae_model]`	`[vae_model]`
LLM + image generation head	trainable	trainable
SigLIP vision encoder	trainable	trainable
EMA		
`ema_enabled`	true	true
`ema_master_dtype`	fp32	fp32
`ema_decay`	0.9998	0.9998
`ema_start_step`	1,000	1,000
`ema_update_every`	1	1
Checkpointing		
`save_strategy`	steps	steps
`save_steps`	500	500
`checkpoint_keep_latest_k`	10	10
`checkpoint_async_mode`	false	true
`save_only_model`	true	true
System integration		
Attention impl.	FlashAttention-2	FlashAttention-2
Liger kernels	RoPE, fused linear-CE, RMSNorm, SwiGLU	same
Activation ckpt (`tm`-style)	true	true
`compile_model`	false	false
`ddp_timeout`	3,600s	3,600s

```
Generated images become part of your context and can inform subsequent
reasoning. You may generate more than one image if needed.

If the task requires a final textual answer, wrap it in <answer>...</answer>.
If no textual answer is needed, the response may end after reasoning or
image generation.

Example:
  <think>reasoning step</think>
  <gen_image>[Image size: (512, 512)]
  <think>observe and reason about the generated image</think>
  <answer>final answer</answer>
```

Table 11: Settings of the ablation runs that differ from Table 10.

	Full training data	Spatial training subset
Backbones	BAGEL-7B	BAGEL-7B, InternVL-U-4B
GPUs (effective batch)	32 (S2 only), 48 (staged)	8
Stage-1 data	Pretext-270K	50K traces from the subset's images
Stage-1 checkpoint	main Stage 1, EMA step 5,500	step 10,000
Stage-2 data	Instruct-180K	Spatial training subset
Compared checkpoints	step 3,000 (S2 only), 2,000 (staged)	every 2,000 steps up to 8,000
Peak learning rate	2e-5	2e-5 (both stages)
Warmup steps	1,000	1,000
`max_num_tokens` (S1 / S2)	16,384 / 20,480	16,384 / 20,480
EMA	yes	no

Compute, runtime, and reproducibility. Table 12 summarises the operational footprint. We run on 80 GB GPUs. Stage 1 averages roughly 12 s per step and Stage 2 roughly 45 s per step.

Table 12: Compute and runtime.

Item	Stage 1	Stage 2
GPUs (8×/node)	32 (4 nodes)	48 (6 nodes)
GPU memory	80 GB	80 GB
Avg. wall-clock per step	∼12 s	∼45 s
Approx. MFU	0.18–0.21	0.05–0.08
Total wall-clock (10K / 12K steps)	∼1.4 days	∼6 days
Reported ThinkMorph-SSL checkpoint	EMA, step 5500	EMA, step 5500
EMA window ($1/1-\text{decay}$)	5,000 steps	5,000 steps

Determinism and reproducibility. We do not pin random seeds for data shuffling because dynamic sequence packing already breaks bit-for-bit reproducibility across world sizes: changing the number of GPUs reshuffles which raw samples land in which packed sequence. We therefore release the per-stage configuration, the launch script, and the dataset list (Section C for Pretext-270K, Section D for Instruct-180K) so that runs are reproducible *at the configuration level*. All training code is built on top of the public BAGEL release.

G.2 Evaluation

Evaluation framework. We document the evaluation framework, the per-benchmark scoring protocols, and the LLM-judge configuration used to produce all reported numbers. The same fork, the same set of benchmarks, and the same scoring protocols are used for our model and for every baseline.

Fork and patches. We evaluate inside a fork of VLMEvalKit that adds the patches required for our interleaved (text+image) generation interface, plus the dataset modules for the spatial benchmarks that are not yet upstream (MMSI, MINDCUBE, OMNISPATIAL, SPATIALVIZ-BENCH).

Inference settings. Decoding settings used by our model on every benchmark are listed in Table 13. We use deterministic text decoding (`do_sample=false`; `text_temperature=0.3` only affects the soft-max for tie-breaking when sampling is enabled) so that each checkpoint produces a single trajectory per question. Baselines are evaluated under their own decoding defaults as published in VLMEvalKit; we do not override them, so each baseline is reproduced under the same configuration that was used to report its public numbers.

Per-benchmark scoring protocols. We evaluate the twelve benchmarks reported in the main text, with CV-Bench counted once although its 2D and 3D splits are reported separately. For each benchmark we list the split we use and the scoring rule. Unless otherwise stated, scoring is rule-based MCQ: the predicted letter is extracted from the model's final `<answer>` span (or, for non-interleaved baselines, from the raw response) and compared against the gold letter; questions whose response cannot be parsed into a letter fall back to the LLM judge described below.

Table 13: Decoding settings used by our model for every evaluated benchmark. Baselines use their own published defaults.

Field	Value	Notes
`ngpus`	8	one process per GPU, NCCL backend
`batch_size`	8	per rank
`do_sample`	`false`	greedy text decoding
`text_temperature`	0.3	inactive under greedy
`max_think_token_n`	4096	per text round
`max_rounds`	3	at most 3 think→image rounds
`num_timesteps`	50	rectified-flow image steps
`cfg_text_scale`	3.0	classifier-free guidance, text
`cfg_img_scale`	1.5	classifier-free guidance, image
`cfg_interval`	[0.4, 1.0]	guided steps
`timestep_shift`	3.0	rf shift
`cfg_renorm_type`	`global`	renormalization
`use_torch_compile`	`true`	compiled forward pass

V*Bench. [94] The V* score is the unweighted mean of the `direct_attributes` and `relative_position` splits. Rule-based MCQ.

HRBench. [86] `HRBench4K` and `HRBench8K` are reported separately because the two resolutions exercise different aspects of the visual encoder. Rule-based MCQ.

MME-RealWorld-Lite. [116] We score the `Reasoning` and `Perception` splits separately and report their average as MME-RW-L in the main tables.

RealWorldQA. [98] Rule-based MCQ on the official split.

BLINK. [25] We report only the spatial subset of BLINK. Concretely we keep the spatial-reasoning task families (counting, jigsaw, multi-view, relative depth, spatial relation) and drop the perception subset. Rule-based MCQ.

CV-Bench. [78] `CV-Bench-2D` and `CV-Bench-3D` are reported separately. The 2D number averages `COCO` and `ADE20K`; the 3D number averages `Depth` and `Distance`.

SAT. [67] We evaluate the circular variant `SAT_circular`: each question is presented under all k rotations of the option order and counted as correct only if all k predictions are correct.

MMSI. [104] We use the non-circular split (this is what `MMSI` maps to by default in our fork). The score is the unweighted mean over the 11 task categories defined by the official benchmark (`Pos-Cam-Cam`, `Pos-Obj-Obj`, `Pos-Reg-Reg`, `Pos-Cam-Obj`, `Pos-Obj-Reg`, `Pos-Cam-Reg`, `Attr-Meas`, `Attr-Appr`, `Motion-Cam`, `Motion-Obj`, `MSR`).

MindCube. [107] We use the tiny raw-QA split (`MindCubeBench_tiny_raw_qa`). The full `MindCubeBench` is prohibitively large for per-checkpoint sweeps; the tiny split is the official subset used for ablations.

OmniSpatial. [39] We use the default split.

SpatialViz-Bench. [85] Rule-based MCQ on the official split.

Interleaved generation (our model only). Our model can choose to emit an intermediate image during reasoning. At each round it generates a text segment with `do_sample=false`; if the segment contains the special string `<gen_image>`, the current context is fed to the image generator with the CFG settings of Table 13, and the resulting image is appended to the context (via the VAE encoder) before the next text round. Otherwise the loop terminates and the final `<answer>` span is returned. Up to `max_rounds=3` text→image rounds are allowed before the model is forced to commit to an answer. This trigger is specific to our interleaved interface; baselines run plain text-only decoding.

Judge configuration and prompts.

Judge model. For benchmarks that do not reduce to exact letter matching we use a single LLM judge: `GPT-5`. Decoding is deterministic (`temperature=0`, `max_tokens=default`), and we retry up to three times before falling back to grade `C` (INVALID).

Generic extract-and-grade prompt. The judge uses one prompt template for every benchmark. It is asked to do two things in one call: *(i)* extract a normalized final answer from the candidate trace, and *(ii)* grade that extracted answer against the gold answer as one of `A`=correct, `B`=incorrect, `C`=invalid. The prompt is reproduced verbatim in the box below. The output format is a single line `<GRADE>\t<EXTRACTED_ANSWER>`; we parse the first non-empty line of the response to recover both fields.

Judge prompt (verbatim) used for MMSI fallback and OmniSpatial

```
You are an expert grading assistant.
Your job has TWO tasks:
1) From the candidate's full response, EXTRACT the final answer in a
   concise, normalized form.
2) Compare this extracted answer with the STANDARD ANSWER and grade it as:
   - A: CORRECT
   - B: INCORRECT
   - C: INVALID

Here are the detailed evaluation criteria:
1. ALWAYS refer to the given STANDARD ANSWER. You do NOT need to re-solve the
   question; the standard answer has already been provided and is always
   correct. Your only job is to judge whether the candidate's final answer is
   consistent with the standard answer according to the form of the question.
   THE STANDARD ANSWER IS ALWAYS CORRECT AND THE QUESTION IS PERFECTLY VALID.
   NEVER QUESTION THEM.
2. ONLY compare the FINAL ANSWER -- COMPLETELY IGNORE any potential errors or
   issues in the REASONING PROCESS. Even if the reasoning is wrong, as long
   as the final answer matches the standard answer, grade it as CORRECT.
3. Answers may be expressed in different ways (e.g., mathematical expressions,
   textual descriptions). As long as the meaning is the same as the standard
   answer, treat them as equivalent. If the standard answer does not specify
   a unit but the candidate's answer includes a correct unit for the given
   value, consider it CORRECT.
4. Some answers may consist of multiple items, such as multiple-choice
   questions with multiple correct options, multi-select questions, or
   multi-blank fill-in-the-blank questions. Regardless of the question type,
   the final answer is considered CORRECT only if it matches the standard
   answer exactly at the level of all required items. For multi-select or
   multi-blank questions, ALL parts must be answered correctly and match the
   standard answer exactly to be deemed CORRECT.
5. If the candidate's answer is wrapped in LaTeX-style markers like
   \boxed{...}, IGNORE the \boxed and only use the inner content as the
   candidate's final answer when comparing with the standard answer.
6. If the candidate's answer is INVALID -- for example, incomplete (cut off
   mid-response), containing a large amount of abnormal repetitive content,
   clearly irrelevant to the question, or explicitly refusing to answer due
   to ethical concerns, lack of information, or other external factors --
   then you MUST grade it as C: INVALID.
7. This instruction applies to all problem types, including single-choice
   MCQ, multi-select MCQ, numeric problems, short-answer questions, and
   general VQA-style questions. In all cases, only the FINAL ANSWER and its
   consistency with the standard answer matter.
8. The question or options may contain image placeholders such as '<image>',
   '<image id=1>', or similar tokens. You CANNOT see these images. Treat
   these placeholders as unknown content and DO NOT hallucinate or infer any
   specific visual details from them. If the standard answer or candidate's
   answer refers to an option associated with an image (e.g., 'choose A'),
   judge correctness only based on the stated answer, not by imagining the
   image.
```

```
IMPORTANT -- OUTPUT FORMAT:
* You MUST return EXACTLY ONE line in the following format:
  <GRADE>\t<EXTRACTED_ANSWER>
  where <GRADE> is one of A, B, or C.
* <EXTRACTED_ANSWER> should be the final answer you extracted from the
  candidate's response, in a normalized, concise form (e.g., a number, a
  letter option, or a short phrase).
* If you cannot extract any meaningful answer, or the response is INVALID,
  output:
  C\tN/A
* Do NOT add any extra text, explanation, or additional lines.

Now, judge the following question.
<Original Question Begin>
{question}
{options_block}
<Original Question End>
<Standard Answer Begin>
{gold_answer}
<Standard Answer End>
<Candidate's Answer Begin>
{llm_response}
<Candidate's Answer End>
Your output:
```

Per-benchmark customization. For MCQ benchmarks the judge sees the option block formatted as `Options:\n A. ...\n B. ...`; the gold answer is rewritten as `<letter>. <option text>` so that the judge can recover from cases where the candidate writes only the option text without the letter. For free-form (VQA-mode) benchmarks the option block is empty and the gold answer is passed through verbatim. The grade→score mapping is `A`→1, `B`→0, `C`→0, with the `C` count separately reported as the *invalid rate* so that overall accuracy and accuracy-among-valid can both be recovered.

H Additional Analyses

This section presents qualitative cases, discusses overlap between our training data and the evaluation benchmarks, and covers alternative routes without visual-state training. Evaluations follow the protocol in Section G.2.

H.1 Qualitative cases

Here we show six reasoning traces from the evaluation: three where ThinkMorph-SSL answers correctly (Figures 10 to 12) and three where it does not (Figures 13 to 15). Every card lays the input scene, the question, the parsed answers from Qwen2.5-VL-7B and BAGEL-7B-MoT (think-4k) on the same prompt, and ThinkMorph-SSL's full `<think>–[Generated Image: ...]–<think>` trace ending in a parsed answer letter.

The intermediate image changes with the level of the question. On L1, the model renders a focused view of the queried region; on L2, it emits a spatial aid such as a depth map or the objects to count, masked out; on L3, it constructs a scene state that the input does not show, such as a top-down map or a shifted viewpoint on MindCube and a merged panorama on MMSI.

The cases share a high-level pattern: ThinkMorph-SSL's intermediate image is a *purpose-built visual scaffold* that converts the question into a read-off step. When the scaffold matches an operator the Instruct-180K curriculum has trained (Section D) and is correctly instantiated, ThinkMorph-SSL outperforms both 7B-class baselines. When the scaffold is the wrong tool, the trace can still be locally coherent but lands on the wrong answer.

Good cases: what the intermediate image is doing. The three success traces use two distinct intermediate-image roles.

Depth views answer relative-distance questions. Figures 10 and 11 show CV-Bench-3D prompts that compare the real-world distances of three boxed objects. ThinkMorph-SSL renders a viridis depth map and reads the answer off the colour bands.

Crops disambiguate counts in cluttered scenes. Figure 12 asks how many people are seated at a café table next to a standing distractor. From the global view both baselines say "two". ThinkMorph-SSL crops to the table and counts only the seated bodies, arriving at three.

In all three cases the heavy lifting sits in the rendered image; the surrounding text reasoning is short.

Failure modes. The three failures split into two qualitatively different kinds: one out-of-distribution case, where the question is far from anything Instruct-180K trains and ThinkMorph-SSL has no scaffold that fits, and two in-distribution mistakes, where the right kind of intermediate image is in scope but the rendered version is noisy or incomplete.

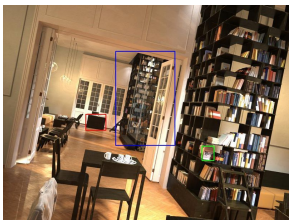
Out-of-distribution: no useful intermediate to render. Figure 13 asks which of four cube nets, when folded, cannot reproduce a shown cube — a question about face adjacencies under unfolding. ThinkMorph-SSL has not seen anything like it during training. It still emits a generated image, but the rendered view simply colors every view blue, so the second thought reasons over depth-coded "adjacency" that has nothing to do with the topology of the question and arrives at the wrong answer. Both baselines reach the right answer by language-only reasoning.

Noisy intermediate. The rendered intermediates in the next two cases are themselves wrong, and the read-back steps inherit their mistakes.

Figure 14 asks for the number of cars in a festival scene. ThinkMorph-SSL announces a mask-overlay intermediate to isolate only the cars, but the rendered mask highlights a person carrying an umbrella instead of any of the vehicles. The second thought reads back from that wrong mask and the parsed answer is wrong, even though the question is well within Instruct-180K's scope.

Figure 15 states a cardinal constraint in text ("chandelier is north of the stove") and asks what lies east of the dining room. ThinkMorph-SSL chooses the right kind of intermediate (a panorama merge of the two ego frames), but the rendered "panorama" is essentially Input 1 duplicated — it does not actually splice information from Input 2 into a single continuous layout. The second thought

CV-3D Intermediate image ✓



INPUT

Q. Estimate the real-world distances between objects in this image. Which object is closer to the desk (highlighted by a red box), the bookcase (highlighted by a blue box) or the books (highlighted by a green box)?

(A) bookcase **(B)** books

Gold: A

BASELINES

Qwen2.5-VL-7B → **B** ✗

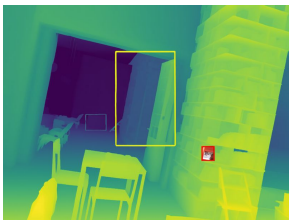
BAGEL-7B (think-4k) → **B** ✗

ThinkMorph-SSL → **A** ✓ *(after generating intermediate)*

ThinkMorph-SSL REASONING TRACE

THINK 1

The question asks which object is closer to the red-boxed desk: the blue-boxed bookcase or the green-boxed books. From the original room view, the desk sits deeper into the room near the back, while the bookcase is also farther back but appears adjacent to the desk area, and the books are on the large shelving unit in the right foreground. To judge real-world proximity more reliably, I will generate a depth-support view next so I can compare how far the desk, the blue-boxed bookcase, and the green-boxed books are from the camera and infer which is nearer to the desk.



GENERATED

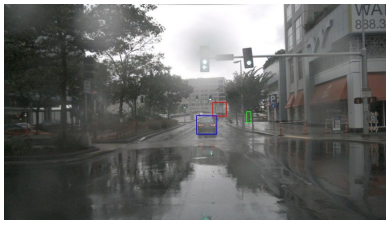
THINK 2

In the depth-support view, the desk region near the back-left is shown in darker blue-purple tones, indicating it is farther away. The large shelving unit on the right is bright yellow-green, so the books on it are much closer to the camera. The blue-boxed bookcase lies between the desk and the right-side shelving, with a depth more similar to the desk than to the foreground books. Since the desk is farther back and the bookcase is positioned near that same back area, the bookcase is closer to the desk than the books are. Therefore the correct choice is A, bookcase.

Figure 10: CV-Bench-3D: depth map shows the bookcase shares the back-wall band with the desk while the books sit in the foreground.

reasons over a degenerate panorama and picks the wrong east object; Qwen2.5-VL-7B reaches the gold answer with no intermediate image at all.

CV-3D Intermediate image ✓



INPUT

Q. Estimate the real-world distances between objects in this image. Which object is closer to the truck (highlighted by a red box), the car (highlighted by a blue box) or the pedestrian (highlighted by a green box)?

(A) car **(B)** pedestrian

Gold: B

BASELINES

Qwen2.5-VL-7B → **A** ✗

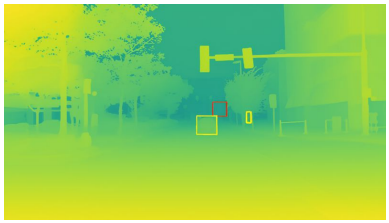
BAGEL-7B (think-4k) → **A** ✗

ThinkMorph-SSL → **B** ✓ *(after generating intermediate)*

ThinkMorph-SSL REASONING TRACE

THINK 1

In the rainy street scene, the truck is the red-boxed vehicle near the center distance, the car is the blue-boxed vehicle slightly closer to the camera, and the pedestrian is the green-boxed figure on the right sidewalk. To decide which object is closer to the truck, I need a depth-support view that converts the scene into relative distance bands, so I will generate a depth map of the same scene and compare the truck's depth with the depths of the car and pedestrian.



GENERATED

THINK 2

The generated depth-support view shows the road foreground in warmer yellow-green tones and the far background in cooler teal tones, so objects with similar colors are at similar real-world distances. The truck's red-boxed location sits in a cooler, farther band near the center background. The pedestrian's green-boxed location is also in that same distant region, while the car's blue-boxed location is noticeably warmer and therefore closer to the camera. Since the truck is farther back than the pedestrian but much closer in depth to the pedestrian than to the nearer car, the pedestrian is closer to the truck. Therefore the answer is B pedestrian.

Figure 11: CV-Bench-3D: same depth scaffold on a rainy street; the truck and pedestrian fall in the same far band while the car is closer.

HR-4k Intermediate image ✓



INPUT

Q. How many people are seated at the table on the right side of the image where a woman in a striped shirt is standing?

(A) Two people **(B)** Four people **(C)** Five people **(D)** Three people

Gold: D

BASELINES

Qwen2.5-VL-7B → A ✗

BAGEL-7B (think-4k) → A ✗

ThinkMorph-SSL → D ✓ *(after generating intermediate)*

ThinkMorph-SSL REASONING TRACE

THINK 1

Start by locating the woman in the striped shirt mentioned in the question. She is standing near the lower middle-right area of the scene, beside a café table and among several metal chairs. Since the question asks about the table on the right side of the image where she is standing, focus on the outdoor seating area immediately to her right. This area contains multiple tables, chairs, and diners, but some people are partially blocked by the woman, the table, and the nearby chairs, so the count is not fully reliable from the full scene alone. To resolve this, I will generate a closer view of the café seating area around the striped-shirt woman so I can count the seated people at that …



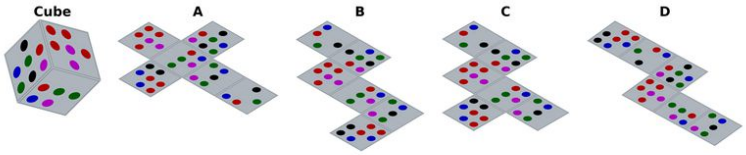
GENERATED

THINK 2

In the zoomed-in view, the relevant table is much clearer. Around the table, I can distinguish three seated people: one person in a dark hoodie seated on the right side, one person in a light blue sleeveless top seated in front, and another partially visible person seated on the left side near the edge of the table. The striped-shirt woman is standing beside this table rather than seated, so she is not counted. Counting only the seated people at that table gives a total of three. Therefore, the correct answer is D.

Figure 12: HRBench-4k: a tight crop separates seated diners from a standing distractor.

SpatViz Incorrect ✗



INPUT

Q. The left image shows a cube with different patterns on its six faces from a particular viewing angle. The options are nets (unfolded patterns) of the cube, which are folded upward to form the cube. Which net, when folded, cannot form the cube shown in the left image? Please answer from options A, B, C, or D.

(A) A **(B)** B **(C)** C **(D)** D

Gold: D

BASELINES

Qwen2.5-VL-7B → **D** ✓

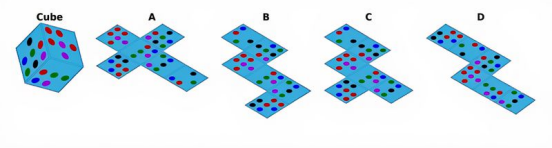
BAGEL-7B (think-4k) → **D** ✓

ThinkMorph-SSL → **C** *(after generating intermediate)*

ThinkMorph-SSL REASONING TRACE

THINK 1

I first compare the cube in the left image with the four candidate nets. The cube has three visible faces: one with red dots, one with black dots, and one with green dots, plus the opposite faces showing pink, blue, and another patterned face. To determine which net cannot fold into that cube, I need to check whether the face adjacencies in the net match the cube's visible adjacencies. I will generate a depth-support view next so the cube's front/back and left/right structure is easier to read, which helps verify which faces must touch each other in a specific order.



GENERATED

THINK 2

The generated depth-support view makes the cube's geometry clearer: the red-face side is the front, the black-face side is to the right, and the green-face side is to the left, with the pink and blue faces on the back and the remaining face on the opposite side. This means that in a valid net, the red, black, and green faces must lie in a consistent arrangement around the central cross, with the black face adjacent to the red face and the green face adjacent to the red face on the opposite side. Looking at the options, net C fails this condition because its black and green faces are placed on the same side of the central face, which would make them opposite each other rather than adjacent. …

Figure 13: SpatialViz-Bench (out-of-distribution): the rendered intermediate is a near-uniform blue image with no usable structure, so the second thought has nothing to read back from.

CV-2D Incorrect ✗



INPUT

Q. How many cars are in the image?

(A) 3 **(B)** 4 **(C)** 1.0 **(D)** 0.0 **(E)** 2.0

Gold: E

BASELINES

Qwen2.5-VL-7B → **C** ✗

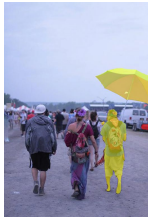
BAGEL-7B (think-4k) → **Z** ✗

ThinkMorph-SSL → **B** *(after generating intermediate)*

ThinkMorph-SSL REASONING TRACE

THINK 1

The scene shows a festival walkway with several people in the foreground and vehicles farther back on the right side. Because the cars are small and partially blended into the background, I will generate a mask overlay that isolates only the car instances on a plain background so each distinct vehicle can be counted without distraction from the people, umbrella, and tents.



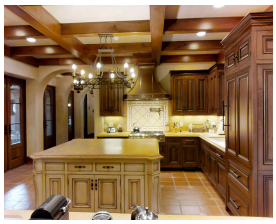
GENERATED

THINK 2

The mask overlay isolates the target vehicles and makes the count clearer. In the right background, there are two separate car instances visible: one darker car behind the people near the center-right and one light-colored car farther right. Since these are distinct isolated vehicle shapes, the total number of cars is 2.

Figure 14: CV-Bench-2D mistake: the model declares a car-segmentation mask but the rendered overlay highlights a person carrying an umbrella; the read-back step then reasons about the wrong objects.

MMSI Incorrect ✗



INPUT 1



INPUT 2

Q. The direction of the chandelier relative to the stove is north. What object is in the east of the dining room?

(A) Painting **(B)** Cabinet **(C)** Stove **(D)** Chandelier

Gold: A

BASELINES

Qwen2.5-VL-7B → **A** ✓

BAGEL-7B (think-4k) → **B** ✗

ThinkMorph-SSL → **B** *(after generating intermediate)*

ThinkMorph-SSL REASONING TRACE

THINK 1

From the kitchen view, the stove is centered against the back wall and the chandelier hangs above the island. The second view shows the adjacent dining room through the archway on the right side of the kitchen. To answer what lies east of the dining room, I need a merged panoramic view that combines both viewpoints so the kitchen, dining room opening, and the objects around them can be placed in one continuous layout.



GENERATED

THINK 2

In the merged panoramic view, the dining room is clearly visible through the archway on the right side of the kitchen. The stove remains on the back wall, and the chandelier is above the island, matching the condition that the chandelier is north of the stove. Using that orientation, east corresponds to the right side of the scene. Looking to the east of the dining room, the prominent object there is the tall wooden cabinet. Therefore the correct option is B.

Figure 15: MMSI mistake: the rendered panorama is essentially Input 1 duplicated rather than a real merge of the two ego views, so the layout the second thought reads is incomplete.

H.2 Data overlap and contamination

BAGEL's pretraining data are not public, so we cannot measure image-level overlap between our source images and its pretraining corpus, and we do not claim that every source image in Pretext-270K is unseen. Possible overlap in raw images does not, however, make Stage 1 a lookup task. Each Stage-1 trace presents a newly corrupted or transformed input, and the model must infer what changed, generate the restored image, and continue the trace. The corruptions, trace format, and descriptions are constructed by our pipeline, so even an image seen during pretraining did not appear in this form. On the evaluation side, Instruct-180K does not reuse benchmark questions or official templates, and all benchmarks are evaluated on held-out test splits. These measures rule out training on evaluation questions and prior exposure to our exact traces, but not every effect of overlap in pretraining images.

H.3 Alternative routes without visual-state training

Text-only fine-tuning could in principle give a unified model visual reasoning without training it to generate and use intermediate images. Prompting alone does not suffice, as shown in Table 2.

Text-only reasoning fine-tuning. ThinkMorph [30] fine-tunes BAGEL-7B on the same training problems with either text-only or interleaved traces, so the data are fixed and only the reasoning mode changes. Text-only traces raise out-of-domain accuracy from 55.49 to 56.02 on $V^\star$, from 67.33 to 68.67 on BLINK-Jigsaw, and from 70.33 to 76.33 on MMVP; interleaved traces reach 63.87, 73.33, and 82.66. With the data held fixed, the larger gain comes with constructing a visual state, not with reasoning fine-tuning as such.

I Declaration of LLM Usage

We used a general-purpose coding assistant for routine, low-level code tasks and an LLM-based writing aid for editing-pass polish on prose drafts authored by the human authors. Both were used under continuous author supervision, with all generated text and code verified before inclusion. No LLM contributed to the core research method, the dataset construction logic, or any quantitative result reported in this paper. Separately, LLMs and image-editing models serve as tools in data construction and evaluation: GPT-5 writes the text of Pretext-270K (Section 3), the Instruct-180K pipelines use the models named in Section D, and GPT-5 is the evaluation judge (Section G.2).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state three concrete claims — (i) a two-stage pretext-then-instruct curriculum with BAGEL-7B-MoT as the main backbone and InternVL-U-4B as a second backbone, (ii) consistent gains over text-only chain-of-thought on the L1–L3 cognitive levels, and (iii) the largest gains on tasks that require constructing a scene state the input does not directly show. Each claim is matched by an experiment in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Limitations paragraph of Section 6 discusses the untested scaling with pretext data, compute, and model size, the fixed set of pretext tasks and evaluated domains, inference cost, and possible pretraining overlap.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper presents an empirical study of a training recipe and curated datasets; it does not contain theoretical results.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The data curation pipeline is documented end-to-end in Section D (per-task pipelines, prompts, hyperparameters), and the training/eval setup is documented in Section G (Stage 1, Stage 2, loss, prompts, compute, EMA) and Section G.2 (VLMEvalKit configuration, per-benchmark protocols, judge prompts).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We release both checkpoints (ThinkMorph-SSL-Base and ThinkMorph-SSL-Instruct) and the Pretext-270K and Instruct-180K corpora on Hugging Face. The model cards give the system prompt and loading instructions. Section D and Section G document the data construction and the training setup.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Stage 1 and Stage 2 hyperparameters, schedule, loss configuration, prompt templates, and EMA settings are reported in Section G; per-benchmark inference settings (decoding, max-think tokens, judge prompts) are reported in Section G.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: We do not report formal error bars. Each main-table cell is a single deterministic-decoding run on the step-5500 EMA checkpoint of the same training run; full multi-seed sweeps were infeasible under our compute budget.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Hardware (GPU type and count), parallelism (FSDP2), step time, and total wall-clock budget for Stage 1 and Stage 2 are reported in Section G.1. Inference-time settings (decoding budget) are listed in Section G.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics. The paper introduces no new human-subject data collection, uses only publicly released datasets under their stated research licenses, does not target safety-critical or surveillance applications, and discusses potential negative impacts in Section 6.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The Broader impact paragraph of Section 6 discusses possible assistive uses, the risk that plausible but wrong intermediate images are taken as evidence, possible surveillance use of viewpoint synthesis, and inherited dataset and model biases.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Apart from a small hand-collected set of 25 scene groups used to train the bird's-eye-view renderer (Appendix D), all image data is sourced from existing publicly released datasets that have already passed their original release reviews; the rest of our contribution is re-annotated Text–Image–Text traces. We plan a release strategy — a research-use license that prohibits surveillance, biometric ID, or impersonation use, a model card documenting failure modes, gated weight access tied to a code-of-conduct agreement, and a responsible-disclosure channel. We do not claim that these measures are exhaustive.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets used to construct Pretext-270K and Instruct-180K (PixMo-Cap, SA-1B, Matterport3D-derived panorama jigsaw, RefSpatial, Ego-Exo4D, Viewpoint-100K, MMSI, MindCube) are cited in Section 3 with their original references; the BAGEL-7B-MoT base model is also cited. We use each asset within its stated research-use license terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The two new derived corpora (Pretext-270K and Instruct-180K) are documented in detail: corpus composition in Section 3, per-task curation pipelines and prompts in Section D, and example traces in Sections F and E.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. Apart from a small hand-collected set of 25 scene groups, all training data is derived from existing publicly released datasets, and Text–Image–Text captions are produced by an LLM, not by human annotators.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects, so IRB review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were used only for prose-editing polish and routine low-level coding assistance, with continuous human supervision and verification. They were not part of the research methodology or used to produce any quantitative result. (The use of LLMs as tools in data construction and as the evaluation judge is documented in Section I; that is part of the method, not the LLM-as-author scenario this question targets.)

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.