ByteDance | Seed

Learning and Reasoning with Geometry: How Generation Helps Spatial Understanding

Rui Yang[1], **Yanwei Li**[2,✉], **Sili Chen**[3], **Jingjia Huang**[3], **Yi Lin**[3], **Hengshuang Zhao**[1,✉]

[1]The University of Hong Kong, [2]Shanghai Jiao Tong University, [3]ByteDance Seed

Abstract

Image and video generators can produce spatially consistent visual content, but how to use this capability to improve spatial understanding remains underexplored. We study this problem from two perspectives: geometric supervision during training and explicit geometric evidence during inference. To this end, we introduce a unified geometric understanding and generation model (GeoUG) that integrates depth and multi-view generation with spatial understanding. First, we design a multi-scale depth codec that enables metric-depth generation using a pretrained visual autoencoder, and show that joint training yields task-dependent gains in spatial understanding. Second, we introduce 3DCueBench to control geometric difficulty while balancing selected 2D cues, and find that generated depth provides larger reasoning gains on questions requiring fine-grained geometric discrimination. Building on this finding, we develop adaptive interleaved reasoning, in which GeoUG first drafts an answer and then decides whether to generate geometric evidence to confirm or revise it. We train this decision through reinforcement learning, balancing accuracy gains over retaining the draft against generation cost. GeoUG achieves circular accuracies of 97.9%, 90.8%, and 94.2% on CV-Bench-3D, 3DCueBench, and DA-2K with fewer generation calls than always generating.

Date: October 23, 2026
Code: https://github.com/hku-sail/GeoUG

1 Introduction

Foundation models increasingly learn from multimodal observations of the world, extending beyond the knowledge conveyed through language [2, 3, 64]. While language efficiently expresses concepts and relations, it often leaves the fine-grained geometry of physical scenes and their continuous changes underspecified [60]. Learning to predict visual observations offers a complementary source of supervision, requiring models to capture spatial structure and how it changes across views and over time. Recent advances in visual generation and editing [26, 50, 61] suggest growing capabilities to maintain coherent scene structure through such changes. Recent studies of multimodal pretraining further highlight the potential for generation and understanding to benefit from shared learning, while revealing that knowledge transfer is asymmetric and task-dependent [28, 64]. This raises a concrete question for geometry, which underlies both coherent visual generation and accurate spatial judgments: *how can geometric generation help models learn spatial relations and reason about them?*

To address this question, we examine geometric generation during both learning and reasoning. During learning, visual question answering (VQA) typically supervises the specific relations queried in a scene, whereas

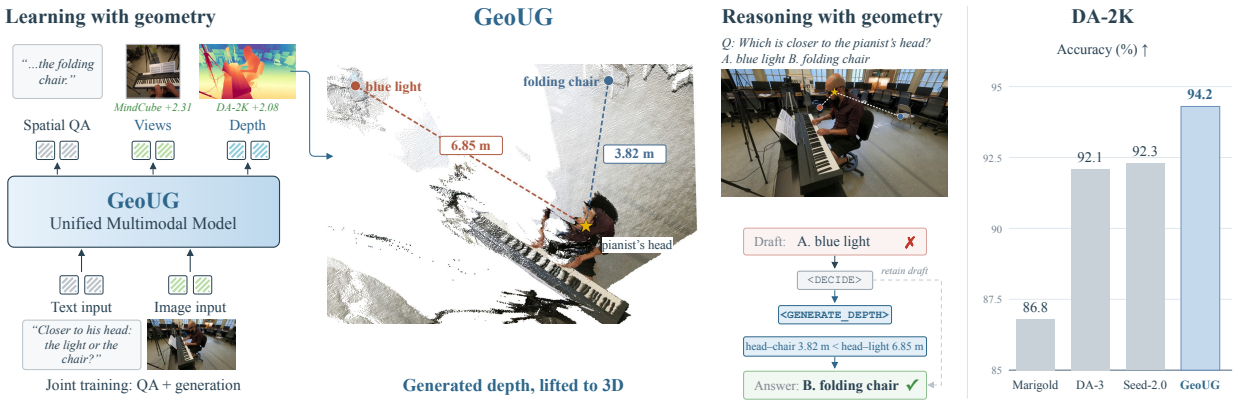


Figure 1 Learning and reasoning with geometry. *Learning with geometry* (left): GeoUG co-trains spatial understanding with geometric generation of new views and metric depth, improving MindCube and DA-2K (Section 5.1). *Reasoning with geometry* (middle and right): GeoUG first drafts an answer, then decides whether to generate depth. Here it generates depth, queries 3D distances from the generated depth lifted to 3D, and revises the incorrect draft. Far right: comparison of GeoUG with Marigold [36], DA3 [43], and Seed-2.0 [13] on DA-2K [77].

generating depth or images from other viewpoints provides supervision over richer scene structure and its visual consequences. This motivates examining how different geometric generation objectives contribute to spatial understanding. During reasoning, generated geometry can provide explicit intermediate representations from which depth and distance measurements can be obtained to support spatial judgments. We therefore ask when using this additional evidence improves an answer and whether models can learn to acquire it selectively. These two perspectives connect learning spatial relations from geometric supervision with reasoning over generated geometry.

To study these two roles, we develop GeoUG (Figure 1), a unified model that integrates spatial understanding with metric-depth generation and view-generation tasks. To enable metric-depth generation, we introduce a Multi-scale Metric Depth Codec (MMDC). MMDC encodes metric depth into a bounded three-channel representation compatible with a pretrained visual autoencoder, with each channel operating at a different depth scale. During training, we jointly optimize a next-token prediction objective for spatial understanding and a flow-matching objective for geometric generation. Comparing different geometric generation objectives reveals task-dependent benefits: metric-depth generation primarily improves depth-related spatial abilities, whereas view generation benefits spatial understanding across viewpoints. During inference, GeoUG uses generated metric depth as explicit geometric evidence. To characterize when this evidence is useful, we introduce 3DCueBench, which balances cases in which selected 2D cues agree or conflict with the true 3D relations across different geometric difficulty levels. We find that generated depth provides larger gains when the compared objects have smaller depth or distance gaps. Motivated by this finding, we develop adaptive interleaved reasoning, in which GeoUG first drafts an answer and then decides whether to generate geometric evidence to confirm or revise it. We train this draft-conditioned policy using reinforcement learning with verifiable rewards (RLVR) [20], introducing a decision-specific advantage that compares generation with retaining the draft while accounting for generation cost. By acquiring geometric evidence selectively, GeoUG retains most of the gains from always generating while requiring fewer generation calls. It achieves circular accuracies of 97.9%, 90.8%, and 94.2% on CV-Bench-3D [62], 3DCueBench, and DA-2K [77], respectively. Together, these findings highlight the task-dependent benefits of geometric supervision during training and the value of selectively acquiring geometric evidence during inference.

Our main contributions are as follows:

- We study geometric generation as both training supervision and reasoning evidence, analyzing the task-dependent benefits of different generation objectives and how inference gains vary with geometric difficulty.
- We introduce GeoUG, which integrates metric-depth and camera-conditioned multi-view generation into a unified understanding and generation model. Using the draft answer as a reference, reinforcement learning balances accuracy gains against generation costs to support on-demand generation and use of geometric

evidence.

- We introduce 3DCueBench, which controls geometric difficulty while balancing the validity of selected 2D cues, enabling a more targeted evaluation of spatial reasoning. Across three spatial understanding benchmarks, GeoUG achieves the highest circular accuracy among the evaluated vision-language models, while using fewer generation calls and maintaining accuracy close to or above the always-generate configuration.

2 Related Work

Unified Understanding and Generation. Unified multimodal models combine language prediction with visual generation [76, 90]. BAGEL uses a Mixture-of-Transformer-Experts (MoT) architecture with specialized understanding and generation experts [21]. MetaMorph reports mutual benefits with asymmetric contributions from understanding and generation data [63]. Recent pretraining studies investigate cross-modal complementarity, knowledge transfer, and conditions for synergy [28, 64]. We examine these interactions specifically through geometric generation and spatial understanding.

Generation for Understanding and Reasoning. Generative supervision improves visual understanding through reconstruction [67] and semantic proxy tasks [82]. At inference time, MVoT generates visual reasoning traces [41], while visual world-modeling studies identify task-dependent benefits of generated intermediates [73]. COOPER learns relative-depth and segmentation generation alongside adaptive interleaved spatial reasoning [86]. For cross-view reasoning, recent work examines synthesized viewpoints and whether models effectively use generated visual evidence [78, 85]. Our work develops multi-scale metric-depth encoding and studies depth and view generation as training supervision and generated depth as evidence during inference, emphasizing how geometric difficulty and generation errors affect downstream benefits. This analysis distinguishes the benefits of learning geometric representations from those of explicitly generating and interpreting them at test time.

Spatial Understanding and Reasoning. Existing approaches strengthen spatial capabilities through geometric representations and spatial supervision. VLM-3R incorporates a geometry encoder to extract implicit 3D tokens from monocular video [22]. From a data perspective, SpatialVLM, Visual Spatial Tuning, and OpenSpatial construct spatial instruction data to improve perception and reasoning [18, 47, 79]. Their instruction targets provide spatially sparse supervision through selected attributes, objects, and relations, rather than dense per-pixel geometric targets. We complement such supervision with metric-depth and multi-view generation objectives, providing dense visual supervision for learning geometry and enabling generated representations to support spatial reasoning.

3 Method

We investigate how geometric generation helps spatial understanding through learning and reasoning with geometry. We build GeoUG on BAGEL [21], a unified understanding and generation model. For learning with geometry, we study co-training configurations that jointly optimize spatial understanding and a geometric generation task, using cross-entropy loss on text tokens and flow-matching loss for visual generation. For reasoning with geometry, the model first drafts an answer and then decides whether to generate depth and query geometric measurements to confirm or revise it. We train this decision to balance answer correctness against generation cost.

3.1 Geometric Generation Tasks

We unify geometric generation tasks through a common visual output representation and velocity-prediction objective based on flow matching [46, 48]. Each target is represented as a three-channel image: an encoded metric-depth map or an RGB target view. Following BAGEL, a frozen VAE encodes this target into latents, and the model predicts the flow velocity from noise-corrupted latents conditioned on the input image, task instruction, and timestep. At inference, the predicted velocity field transforms noise into target latents, which the VAE decodes into the visual output. The tasks thus share one generation mechanism and differ only in conditioning and target.

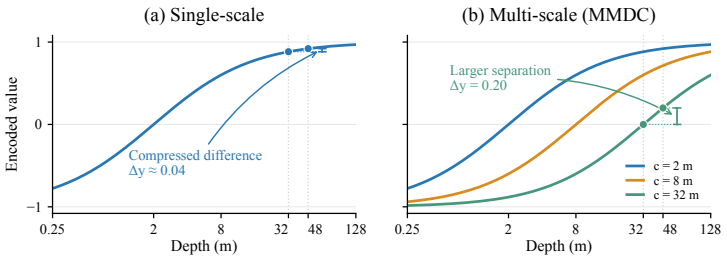


Figure 2 Multi-scale depth encoding. (a) A single pivot compresses depth differences far from its scale. (b) A second channel with a larger pivot separates depths saturated in the first.

We study two forms of geometric supervision within this framework. **Metric-depth generation** captures the depths of visible surfaces, providing explicit geometry for spatial measurement. **View generation** captures changes in spatial layout, visibility, and occlusion induced by a change of viewpoint. In camera-conditioned multi-view generation (*multi-view* for short), the target viewpoint is given by a camera-motion instruction, which we derive from relative camera poses to account for both rotation and translation (Appendix D.2). In exo-to-ego generation, the target viewpoint is that of a person in a third-person image; generating that person's first-person view requires perspective taking, imagining the scene from another observer's viewpoint (Appendix D.3).

Multi-scale Metric Depth Codec. To represent metric depth within this common image interface, we introduce the Multi-scale Metric Depth Codec (MMDC). It maps depth into a bounded three-channel representation compatible with the pretrained VAE and converts the generated output back into metric depth, preserving a consistent physical scale across images.

Generative depth estimators such as Marigold [36] normalize each ground-truth depth map using its scene-specific 2nd and 98th percentiles. Given a metric ground-truth depth value d at a pixel, they construct the normalized target $\tilde{d} = 2\left((d - d_2)/(d_{98} - d_2) - 1/2\right)$, where d_2 and d_{98} are computed independently for each depth map. This produces an affine-invariant target compatible with the bounded input range of an RGB VAE. Because the metric values of d_2 and d_{98} are unavailable at inference, the predicted normalized depth cannot be converted back to metric depth without recovering the unknown scene-specific scale and shift. We instead encode the same metric depth $d > 0$ as a normalized value $y \in (-1, 1)$ using a fixed, invertible transformation with a fixed pivot $c > 0$, where d and c are both in metres:

$$y = \frac{d - c}{d + c} \in (-1, 1), \qquad d = c\frac{1 + y}{1 - y}. \tag{1}$$

Every encoded value now has a consistent metric meaning across images, allowing the generated representation to be decoded into a metric depth map at inference. However, a single pivot c compresses relative depth differences away from its characteristic scale, amplifying reconstruction errors during decoding, as shown in Figure 2(a). MMDC addresses this by exploiting the three input channels of the RGB VAE. Rather than repeat a single-scale encoding three times, we assign each channel a different depth scale:

$$Y_i = \frac{d - c_i}{d + c_i}, \qquad (c_1, c_2, c_3) = (2, 8, 32)\,\text{m}. \tag{2}$$

The channels provide complementary precision across depth ranges: where one channel approaches saturation, another can better distinguish depth differences (Figure 2). The pivots are fixed across datasets, and the resulting representation fits the pretrained VAE without architecture changes.

For decoding, we invert each generated channel and fuse the per-channel estimates:

$$\hat{d}_i = c_i\frac{1 + \hat{Y}_i}{1 - \hat{Y}_i}, \qquad w_i = (1 - \hat{Y}_i^2)^2, \qquad \hat{d} = \frac{\sum_i w_i \hat{d}_i}{\sum_i w_i}. \tag{3}$$

The weight is largest near a channel's pivot and decreases towards saturation, allowing different channels to contribute at different depth scales. More details are provided in Appendix B.

MMDC introduces no learned parameters. During training, encoded depth maps serve as targets through the frozen VAE encoder. At inference, MMDC converts the VAE-decoded outputs into metric depth maps for geometric measurement and interleaved reasoning.

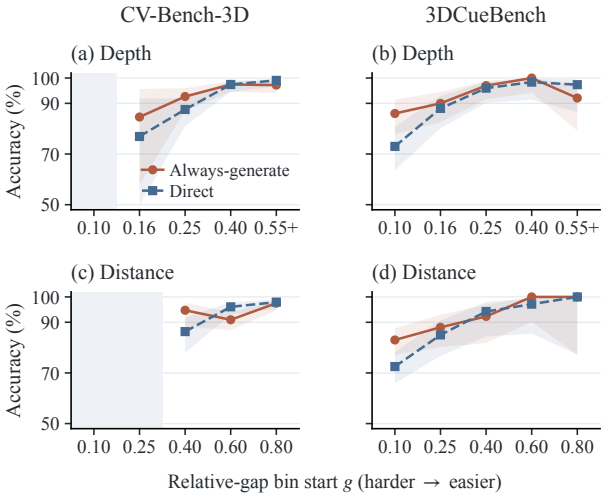


Figure 3 When does generation help? Accuracy versus relative gap g; smaller is harder. Non-circular accuracy from separately trained models. Generation gives larger gains on difficult 3DCueBench questions. Gains on easier questions are limited.

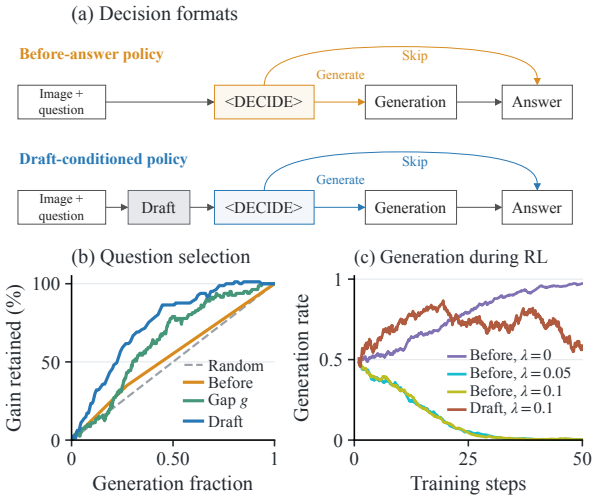


Figure 4 When to generate. (a) Policies decide whether to generate before answering or after a draft. (b) Gain retained at different generation budgets on 3DCueBench (protocol in Appendix C). Gap g prioritizes questions with smaller ground-truth relative gaps. (c) Generation rates during RL with generation cost λ.

3.2 Interleaved Geometric Reasoning

We next turn from geometric supervision during training to using generated depth as evidence for spatial reasoning at inference time. Before designing the reasoning procedure, we address two questions through two pilot studies:

Pilot I: When Does Generation Help? We compare direct answering (*Direct*) with reasoning that always generates depth before answering (*Always-generate*) on CV-Bench-3D [62] and 3DCueBench (Section 4.1). We measure difficulty by the relative gap

$$g = \frac{|d_a - d_b|}{\max(d_a, d_b)}, \tag{4}$$

where d_a and d_b are the camera depths or distances to the anchor being compared. We group questions in both datasets into bins according to their relative gap. Figure 3 shows larger gains from generation-assisted reasoning over direct answering on difficult questions with smaller gaps. On easier questions, direct answering already achieves high accuracy, while generation-assisted reasoning provides limited gains and sometimes reduces accuracy. This degradation may result from errors in generated depth or geometric measurements. These findings motivate adaptive interleaved reasoning, in which the model decides when to generate geometric evidence.

Pilot II: Can the Model Tell When It Needs Geometry? The ground-truth gap is unavailable at test time, so the model must decide whether to generate from its own reasoning context. A straightforward approach is a *before-answer policy*, which chooses between direct answering and generation-assisted reasoning before producing an answer (Figure 4a, top). To evaluate whether the policy identifies questions that benefit from generation, we use its *generation probability*: the probability of choosing the depth-generation branch at the `<DECIDE>` step in Figure 4a. For each generation budget, questions with the highest generation probabilities receive generation-assisted reasoning, while the remainder are answered directly, yielding the orange *Before* curve in Figure 4b. In contrast, the green *Gap g* curve ranks questions by their ground-truth relative gaps in ascending order, prioritizing the smallest-gap questions. The vertical axis reports the fraction of the accuracy gain from always generating retained by each selection strategy on 3DCueBench (the detailed protocol can be found in Appendix C). The Gap g reference requires ground-truth geometry and is unavailable at test time. Random selection retains a fraction of the gain equal to the generation budget in expectation.

At a 50% generation budget, the before-answer policy retains only 55% of the gain from always generating, close to the 50% expected from random selection (Figure 4b). Reinforcement learning also fails to produce selective generation in our runs: the policy tends to generate for either almost no questions or almost all questions (Figure 4c). These results suggest that, in this setup, the before-answer policy does not learn to identify questions that benefit from generation. We draw inspiration from reflective reasoning in chain-of-thought, where a model forms a tentative answer and subsequently reconsiders it before reaching a final answer. This motivates us to use the tentative answer as a draft and place the generation decision at this point of reconsideration.

Specifically, we introduce a *draft-conditioned policy*, which first produces an initial draft from the input image and question without depth generation, then decides whether to retain it as the final answer or generate geometric evidence to verify or revise it (Figure 4a, bottom). Under the same ranking protocol and a 50% generation budget, this policy retains 86% of the gain from always generating, compared with 55% for the before-answer policy (Figure 4b). Each policy's retained gain is normalized by its own direct and always-generate accuracies. We instantiate this policy with depth generation. For example, to decide which of two objects is closer, the model generates a depth map, decodes it into metric depth using MMDC, and queries the objects' depths before confirming or revising its draft.

Learning When to Generate. We train the draft-conditioned policy in two stages. First, we cold-start the policy through supervised fine-tuning on chain-of-thought (CoT) sequences that follow the draft-conditioned reasoning format: an initial answer draft, a decision on whether to generate depth, and a final answer that either retains the draft or uses generated geometric evidence to verify or revise it. Draft tokens are excluded from the training loss (Appendix C).

Starting from this initialization, we use GSPO-based reinforcement learning [87] to optimize the policy's generation decisions for the accuracy–cost trade-off. The reward is $r_i = c_i - \lambda a_i$, where $c_i \in \{0, 1\}$ indicates final-answer correctness, $a_i \in \{0, 1\}$ indicates generation, and λ is the generation cost.

For G rollouts of a question, GSPO assigns each a shared trajectory-level advantage:

$$\bar{r} = \frac{1}{G}\sum_{j=1}^{G} r_j, \qquad A_i^{\text{group}} = \frac{r_i - \bar{r}}{s_r + \epsilon}, \tag{5}$$

where s_r is the standard deviation of rewards within the group and $\epsilon > 0$ ensures numerical stability. However, when every rollout generates, the generation cost cancels:

$$r_i - \bar{r} = (c_i - \lambda) - (\bar{c} - \lambda) = c_i - \bar{c}, \qquad \bar{c} = \frac{1}{G}\sum_{j=1}^{G} c_j. \tag{6}$$

Since subtracting a shared cost also leaves the standard deviation unchanged, the advantage no longer depends on λ. In such groups, the decision receives no cost-dependent signal for learning whether generation is worthwhile. In practice we replace s_r with a constant scale, since normalizing by s_r also removes the magnitude of λ (Appendix C); the cancellation above holds for either choice.

We therefore assign the generation decision an advantage that explicitly compares generation with retaining the draft. Because answering directly returns the draft unchanged, its correctness $c_{d,i}$ is known from the answer label, even for rollouts that generate. For a sampled decision, we use

$$A_i^{\text{gate}} = \begin{cases} (1 - p_i)\,(c_i - \lambda - c_{d,i}) & a_i = 1, \\ -p_i\,(\widehat{Q}_{\text{gen},i} - \lambda - c_{d,i}) & a_i = 0, \end{cases} \tag{7}$$

where p_i is the generation probability conditioned on the draft, and $\widehat{Q}_{\text{gen},i}$ estimates correctness after generation using forced-generation rollouts for the same question. This preserves a signal about the benefit and cost of generation even when all sampled rollouts choose the same branch.

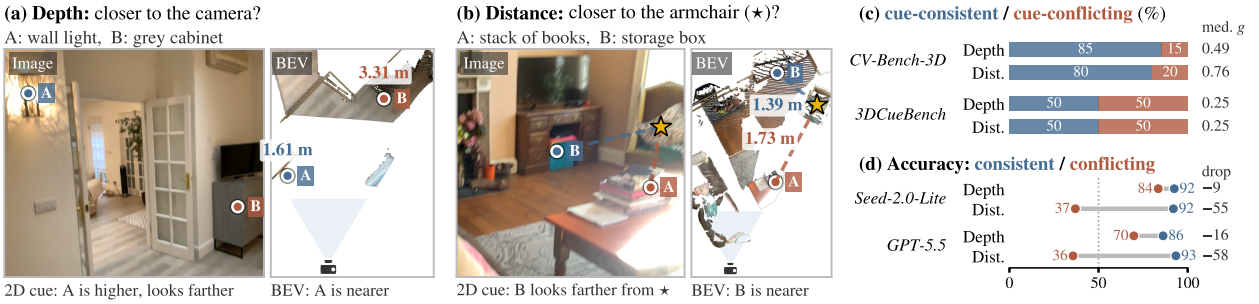


Figure 5 2D shortcuts versus 3D geometry. (a) "Lower is closer" picks the cabinet, but the bird's-eye view (BEV) shows the higher wall light is nearer. (b) The books are closer to the armchair in the image plane, but the storage box is closer in 3D. (c) Share of cue-consistent (2D rule correct) and cue-conflicting (2D rule wrong) questions, with median relative gap g. (d) Ordinary accuracy of two VLMs on cue-consistent vs. cue-conflicting 3DCueBench questions (Appendix Table 10).

4 Evaluation

4.1 3DCueBench: Controlled Evaluation of Geometric Reasoning

VLMs may rely on 2D image-plane relationships to answer 3D spatial questions without understanding the underlying 3D structure. Figure 5(a,b) shows two such shortcuts, "lower in the image is closer" and "closer in the image plane is closer in 3D", each contradicted by the scene's 3D geometry. In our analysis of CV-Bench-3D, choosing the object with the lower bounding-box bottom correctly answers 85.3% of Depth questions (Figure 5(c)), and no question has a relative gap below 0.16. Current VLMs follow such cues: on 3DCueBench Distance questions where the rule gives the wrong answer, Seed-2.0-Lite [13] and GPT-5.5 [51] reach only 37.0% and 35.7% (Figure 5(d)). This makes it difficult to determine whether high accuracy reflects geometric reasoning or reliance on image-plane cues.

We therefore introduce **3DCueBench**, a benchmark of 800 carefully selected questions in two tasks: Depth compares the camera depths of two targets, and Distance compares their 3D distances to an anchor. In half of the questions (*cue-consistent*), the 2D rules used to construct the benchmark favor the correct answer; in the other half (*cue-conflicting*), they favor the wrong answer. Each rule therefore achieves 50% accuracy, so models must go beyond these rules to perform well. To control difficulty, we sample questions across intervals of the relative gap (Equation 4), keeping cue-consistent and cue-conflicting questions balanced within each interval (Appendix E.2).

4.2 Evaluation Protocol

Benchmarks and Metrics. We evaluate single-image depth and distance reasoning on CV-Bench-3D [62], DA-2K [77], and our 3DCueBench. Unless suffixed -*Visual*, CV-Bench-3D and DA-2K denote their textual versions, which give object points or boxes as coordinates in the question text. We further assess multi-view spatial reasoning on MindCube [70] and the camera-motion subset of MMSI-Bench [80], with accuracy or circular accuracy as noted in each figure or table. For dense depth estimation, we evaluate on NYU-v2 [57], KITTI [24], ETH3D [56], Sintel [12], and DIODE [66], reporting AbsRel and δ_1 (threshold 1.25).

5 Experiment

5.1 Generative Co-training

Generative co-training yields task-dependent gains in spatial understanding. We compare understanding-only training with three co-training configurations, each combining spatial understanding with one geometric generation task introduced in Section 3.1: metric-depth generation, camera-conditioned multi-view generation (multi-view), or exo-to-ego generation (exo-to-ego). Figure 6 plots benchmark accuracy against the number of understanding CE tokens consumed during training. All three tasks improve final accu-

Table 1 Depth and distance understanding. Circular accuracy (%); both option orders must be correct. Bold marks the best result per column. GeoUG uses adaptive interleaved reasoning and is initialized from the draft-conditioned SFT model (Section 3.2). DA3 answers from its predicted metric depth with the same measurement procedure as GeoUG, without a VLM. Avg.: mean of the three Overall columns.

Model / configuration	CV-Bench-3D			3DCueBench			DA-2K	Avg.
	Depth	Distance	Overall	Depth	Distance	Overall	Overall	
BAGEL-7B-MoT [21]	81.5	59.3	70.4	73.8	49.5	61.6	55.1	62.4
Qwen2.5-VL-7B [5]	80.2	61.5	70.8	61.8	35.5	48.6	45.3	54.9
Qwen3-VL-8B [4]	93.8	80.3	87.1	80.0	53.5	66.8	44.6	66.2
InternVL3.5-8B [72]	83.2	74.7	78.9	62.8	52.5	57.6	31.6	56.0
SenseNova-SI-1.4-8B [16]	93.5	85.2	89.3	81.3	63.0	72.1	63.9	75.1
SenseNova-SI-1.5-8B [16]	90.8	84.0	87.4	79.0	64.0	71.5	61.1	73.3
VST-7B-SFT [79]	95.5	90.8	93.2	83.8	72.0	77.9	63.5	78.2
DA3 [43] (depth readout)	96.0	94.5	95.3	**95.3**	**86.5**	**90.9**	92.1	92.8
GeoUG (Ours)	**98.5**	**97.3**	**97.9**	95.0	**86.5**	90.8	**94.2**	**94.3**

racy on CV-Bench-3D-Visual by 1.50–1.67 points. Depth generation gives the largest gain on DA-2K-Visual (+2.08 points), while multi-view and exo-to-ego generation improve MindCube by 2.21 and 2.31 points. On MMSI camera motion, multi-view generation improves accuracy, whereas depth generation lowers it. These results suggest that the benefits depend on how closely the generation task matches the spatial ability being evaluated. Appendix E.1 reports the generation quality of the co-trained models, and Appendix F.1 shows that generation pretraining followed by understanding-only SFT does not reliably yield such gains.

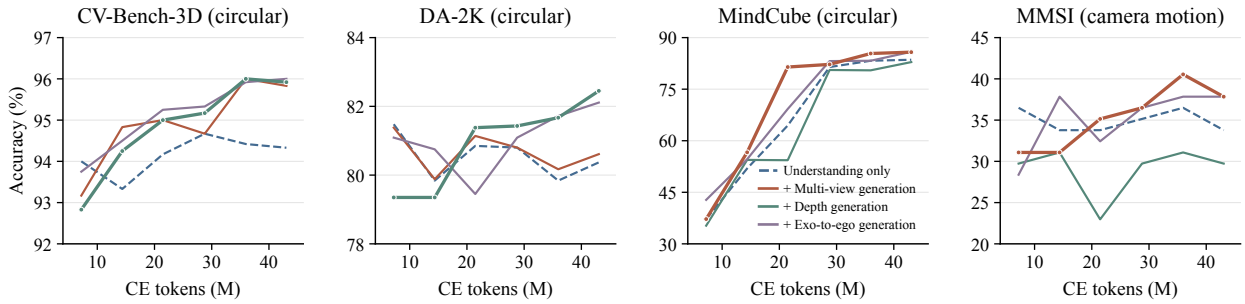


Figure 6 Generative co-training. Circular accuracy on CV-Bench-3D-Visual, DA-2K-Visual, and MindCube; accuracy on MMSI camera motion. x-axis: understanding CE tokens.

Co-training also teaches geometrically controlled generation. Figure 7 shows the three geometric generation abilities. For depth generation, the generated metric depth preserves the scene layout, such as the receding arches and free-standing pillars of the hallway and the overhanging roof of the terrace, and back-projects to coherent 3D reconstructions. For multi-view generation, the original BAGEL [21] often returns a view close to the input, whereas our model moves the camera as instructed and brings new content into view: it swings the view around the cannon when moving right and turning left, and pulls back to reveal the shoreline when moving backward, while BAGEL nearly copies the input. For exo-to-ego generation, BAGEL depicts the described person, whereas our model renders the court from that person's viewpoint, looking up at the hoop during a layup or down at the ball and the goal before a penalty kick. Quantitatively, our models improve PSNR by up to 1.25 dB on multi-view generation and from 7.75 to 12.41 dB on exo-to-ego generation (Appendix E.1); depth accuracy is evaluated in Section 5.4, and more examples are in Appendix A.

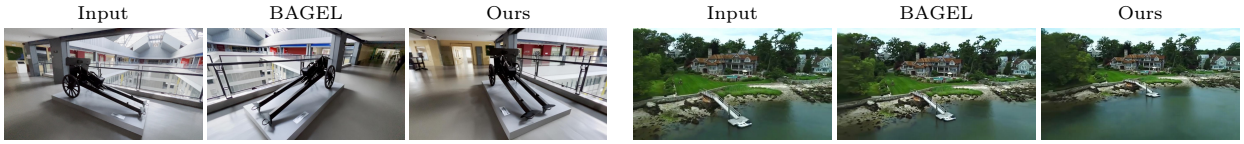
5.2 Interleaved Geometric Reasoning

Table 1 compares GeoUG ($\lambda = 0.1$; Section 3.2) with open-source VLMs and DA3 on depth and distance questions, using circular accuracy.

GeoUG outperforms the evaluated VLMs across all three benchmarks. Table 1 compares GeoUG with open-source VLMs and the depth estimator DA3. Using the same measurement procedure, GeoUG



(a) Depth generation on ETH3D [56]: metric depth and its back-projected 3D reconstruction.



Move the camera right 0.88 m and rotate the camera left 20 degrees

Move the camera backward 1.29 m

(b) Multi-view generation on DL3DV [45] (left) and RealEstate10K [91] (right).



Show me the ego view of the person performing a Mikan layup basketball drill (wearing a white tank top, striped shorts with a blue star, and black shoes)

Show me the ego view of the person in black clothing preparing to take a soccer penalty kick.

(c) Exo-to-ego generation on Ego-Exo4D [27].

Figure 7 Geometric generation after co-training. BAGEL [21] is the original model; Ours are the multi-view and exo-to-ego co-training models (Figure 6) at their final checkpoints. Instructions are shown below each example.

exceeds DA3 by 2.7 points on CV-Bench-3D and 2.1 points on DA-2K and achieves similar accuracy on 3DCueBench (90.8% versus 90.9%), while answering 32.6% of 3DCueBench questions without generating depth. Supplying existing VLMs with DA3 measurements improves accuracy on CV-Bench-3D and 3DCueBench, but their performance remains below GeoUG (Appendix F.3). Referring format affects models differently, with no consistent advantage for either format (Appendix F.2).

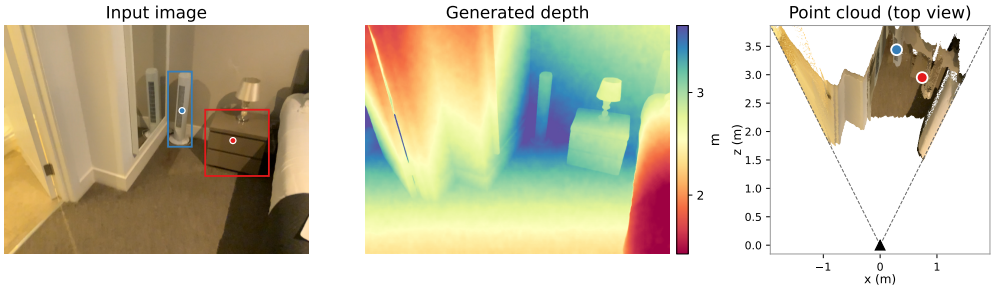
Depth generation provides larger gains on harder geometric questions. Within the same model, always generating depth improves circular accuracy over direct answering by 6.0, 15.9, and 23.2 points on the three benchmarks (Table 2). On 3DCueBench, the gain reaches 23.0 points in the two hardest gap intervals and decreases to 12.5 and 5.0 points in the easier intervals (Table 12). Generation also reduces option-position bias: the fraction of questions answered with the same letter under both option orders drops from 9.7% to 0.1% on average.

Generated depth corrects wrong drafts. Figure 8 shows examples from the three benchmarks in which the draft answer is wrong. In each case, the policy chooses to generate depth, reads metric depths or 3D distances from it, and revises the draft to the correct answer under both option orders. More examples, including direct answers and a failure case, are in Appendix A.

5.3 Adaptive Interleaved Reasoning

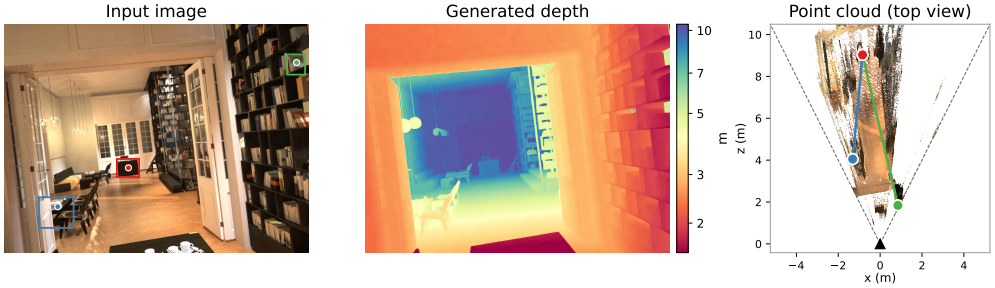
Adaptive interleaved reasoning maintains accuracy with fewer depth generations. With $\lambda = 0.1$, the trained GeoUG generates depth for 50.3–72.4% of questions while staying within 0.7 percentage points of always generating across all three benchmarks (Table 2). The initial SFT policy generates less often but achieves lower accuracy, particularly on DA-2K. Reducing λ to 0.05 increases generation rates to 58.3–78.7%, with accuracy changing by at most 0.5 points. On 3DCueBench, this policy stays within 1.0 point of always generating across difficulty intervals (Table 12).

The draft-conditioned policy selects questions that benefit from generation. At the same generation



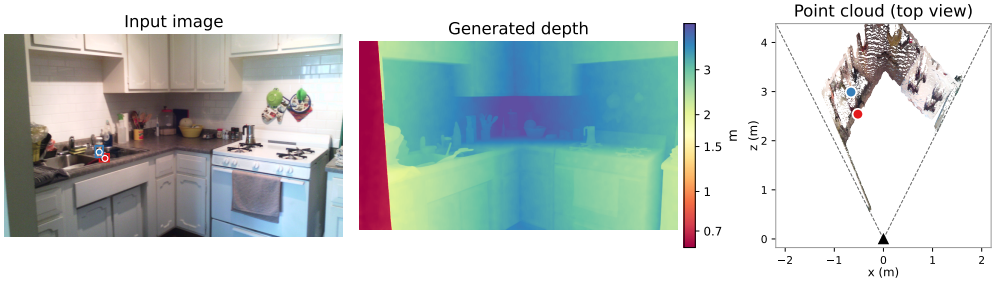
Question. Which object is closer to the camera taking this photo, the **nightstand** or the **tower fan**? A. nightstand B. tower fan
GeoUG. To answer this, I need each object's metric distance from the camera. [boxes and points for both objects] My initial answer without depth: B. tower fan. `<DECIDE><GENERATE_DEPTH>` $p_{\text{gen}} = 0.93/0.96$
[generated depth map, 1.48–7.28 m] `query_point_depth`: **nightstand** 2.95 m, **tower fan** 3.44 m.
From the tool, ... Since $2.95 < 3.44$, the nightstand is closer and the tower fan is further. **Answer: A. nightstand** ✓

(a) 3DCueBench, Depth, relative gap in $[0.10, 0.16)$.



Question. Estimate the real-world distances between objects in this image. Which object is closer to the **desk**, the **table** or the **books**? A table B books
GeoUG. To answer this, I need the 3D distance from the anchor to each candidate. [a point inside each given box] My initial answer without depth: B books. `<DECIDE><GENERATE_DEPTH>` $p_{\text{gen}} = 0.94/0.95$
[generated depth map, 0.89–12.08 m] `query_3d_distance`: **desk**–**table** 5.01 m, **desk**–**books** 7.50 m.
This asks which object is closer to the desk ... Since $5.01 < 7.50$, the table is closest to the desk. **Answer: A table** ✓

(b) CV-Bench-3D [62], Distance, objects given as boxes.



Question. Which object is closer to the camera taking this photo, the **first point** or the **second point**? A. first point B. second point
GeoUG. To answer this, I need each object's metric distance from the camera. [a box around each point] My initial answer without depth: B. second point. `<DECIDE><GENERATE_DEPTH>` $p_{\text{gen}} = 0.98/0.93$
[generated depth map, 0.57–4.83 m] `query_point_depth`: **first point** 2.54 m, **second point** 2.99 m.
From the tool, ... Since $2.54 < 2.99$, the first point is closer and the second point is further. **Answer: A. first point** ✓

(c) DA-2K [77], two nearby points on the sink.

Figure 8 Adaptive interleaved reasoning with GeoUG. GeoUG drafts an answer, decides at `<DECIDE>` whether to generate metric depth, and measures on its own generated depth to confirm or revise the draft. In all three examples, a wrong draft is corrected under both option orders. Images show the input, the generated depth, and a top-down point cloud; p_{gen} is the probability of generation for both option orders. More examples are in Appendix A.

Table 2 Adaptive interleaved reasoning. Circular accuracy (%) and generation rate (Gen., %). Pos.: % of questions answered with the same letter under both option orders (mean over benchmarks). The random gate generates at the $\lambda = 0.1$ rate for random questions.

Configuration	CV-Bench-3D		3DCueBench		DA-2K		Pos.
	Acc.	Gen.	Acc.	Gen.	Acc.	Gen.	
Direct	91.4	0	75.4	0	71.7	0	9.7
Always-generate	97.4	100	91.3	100	94.9	100	0.1
Initial policy (SFT)	97.2	16.0	85.4	25.2	85.1	28.3	3.5
Random gate	94.4	50.3	86.1	67.4	88.5	72.4	3.2
RL, $\lambda = 0.05$	97.5	58.3	90.6	74.6	94.7	78.7	1.0
Ours (λ=0.1)	97.9	50.3	90.8	67.4	94.2	72.4	0.7

Table 3 Ablation of the decision advantage ($\lambda = 0.1$, 20 updates from the same initial policy and training questions). Circular accuracy (%) and generation rate (Gen., %); Avg.: mean accuracy over the three benchmarks. Ours: two-branch estimate with forced generation.

Decision advantage	CV-Bench-3D		3DCueBench		DA-2K		Avg.
	Acc.	Gen.	Acc.	Gen.	Acc.	Gen.	
Always-generate	97.4	100	91.3	100	94.9	100	94.5
Initial policy (SFT)	97.2	16.0	85.4	25.2	85.1	28.3	89.2
Trajectory-level	96.6	44.1	90.8	66.2	**94.1**	71.1	93.8
Sampled action	**97.6**	41.6	89.6	62.8	92.8	65.6	93.4
Two-branch, unforced	97.3	29.2	89.5	49.3	92.5	53.0	93.1
Ours	**97.6**	50.7	**91.0**	69.3	93.9	73.4	**94.2**

rates, GeoUG outperforms random selection by 3.5, 4.7, and 5.7 points on CV-Bench-3D, 3DCueBench, and DA-2K, respectively (Table 2). This advantage holds across all difficulty intervals on 3DCueBench. After RL, fewer incorrect drafts are returned without checking, indicating that the policy learns to use depth where direct answering fails.

Comparing generation with retaining the draft and using forced rollouts improves accuracy. We ablate how the draft-conditioned policy learns when to generate (Section 3.2). Our decision advantage in Equation 7 compares generation with retaining the draft and uses forced-generation rollouts to estimate correctness after generation. Starting from the same initial policy, we evaluate after 20 RL steps at $\lambda = 0.1$ (Table 3). Our method achieves 94.2% mean accuracy across the three benchmarks. Replacing A_i^{gate} with the trajectory-level advantage in Equation 5 lowers accuracy by 0.34 points; applying the draft-correctness baseline only to the sampled action lowers it by 0.81 points. Estimating $\widehat{Q}_{\text{gen}}$ only from rollouts that chose to generate, without forced generation, lowers accuracy by 1.09 points. In this last variant, all 16 rollouts answer directly on 4% of the training questions, leaving no estimate of $\widehat{Q}_{\text{gen}}$ and no learning signal for those decisions. Our method generates more often at the same λ, bringing accuracy closer to always generating.

5.4 Depth Estimation

GeoUG-Depth outperforms the evaluated generative depth estimators. We evaluate depth generation on five benchmarks using affine alignment (Section 4.2, Table 4). GeoUG-Depth achieves lower AbsRel and higher δ_1 than Marigold-v1.1 and Lotus on every dataset with reported results. It also outperforms Depth Anything V2 (DA2) on both metrics across all five benchmarks, while MoGe-V2 retains lower AbsRel. Qualitative examples show the generated depth maps and corresponding 3D reconstructions (Figure 7a and Appendix A).

Table 4 Affine-invariant depth estimation. AbsRel ×100 (↓) and δ_1 in % (↑) after affine alignment.

Type	Method	NYU-v2		KITTI		ETH3D		Sintel		DIODE	
		AbsRel	δ_1	AbsRel	δ_1	AbsRel	δ_1	AbsRel	δ_1	AbsRel	δ_1
Dis	ZoeDepth [7]	4.76	97.30	5.59	95.10	7.27	94.20	21.80	69.20	7.80	90.90
	DA V2 [77]	4.16	97.90	6.77	94.30	4.63	97.20	17.10	76.60	5.41	94.60
	Metric3D V2 [31]	3.94	97.60	3.50	98.40	3.24	99.00	26.60	71.70	2.75	98.70
	UniDepth V2 [52]	2.96	98.60	3.85	98.10	2.95	98.50	13.30	83.20	4.05	96.50
	Depth Pro [9]	3.67	98.20	5.12	96.80	4.97	96.40	15.80	80.10	4.66	95.60
	MoGe-V2 [71]	2.89	98.60	3.75	98.10	2.80	99.10	13.30	82.50	3.14	97.40
Gen	Marigold-v1.1 [36]	5.50	96.40	10.50	90.20	6.90	95.70	29.80	78.20	-	-
	Lotus [29]	5.40	96.80	8.50	92.20	5.90	97.00	-	-	9.80	92.40
	GeoUG-Depth (ours)	3.32	98.46	4.84	97.27	3.75	97.85	14.72	79.65	4.50	96.22

Multi-scale depth encoding improves accuracy, with further gains from more training data. Table 5 compares MMDC with a single-scale encoding that repeats one channel ($c = 5\,\text{m}$) across the three VAE channels. MMDC reduces mean AbsRel from 7.47 to 7.24, improving four datasets and matching the baseline on NYU-v2. Expanding the training data further reduces mean AbsRel to 6.23 and increases mean δ_1 from 92.6% to 93.9%.

Table 5 Depth encoding and data scaling. Protocol and units follow Table 4; AVG is the unweighted mean across five datasets. The last row is GeoUG-Depth in Table 4.

Method	AVG		NYU-v2		KITTI		ETH3D		Sintel		DIODE	
	AbsRel	δ_1	AbsRel	δ_1	AbsRel	δ_1	AbsRel	δ_1	AbsRel	δ_1	AbsRel	δ_1
single-disparity	7.47	92.20	3.35	98.40	6.61	95.22	4.76	96.66	16.68	76.77	5.96	93.96
multi-disparity (MMDC)	7.24	92.57	3.35	98.38	6.51	95.34	4.62	96.99	16.31	77.23	5.40	94.88
MMDC + data scaling	6.23	93.89	3.32	98.46	4.84	97.27	3.75	97.85	14.72	79.65	4.50	96.22

6 Conclusion

We studied how geometric generation supports spatial understanding through both learning and reasoning, using GeoUG as a unified framework. During training, joint learning with depth and multi-view generation yields task-dependent gains. During inference, our evaluation on 3DCueBench shows larger benefits from generated depth on questions requiring fine-grained geometric discrimination. Building on these findings, we developed adaptive interleaved reasoning, where a draft-conditioned policy learns through reinforcement learning to balance accuracy gains against generation cost. Across three benchmarks, this policy maintains accuracy close to or above always generating with fewer generation calls. These results highlight two complementary roles for geometric generation: supervision for learning spatial relations and evidence acquired selectively during reasoning.

References

[1] Wendy J. Adams, James H. Elder, Erich W. Graf, Julian Leyland, Arthur J. Lugtigheid, and Alexander Muryy. The Southampton-York natural scenes (SYNS) dataset: Statistics of surface attitude. *Scientific Reports*, 6, 2016.

[2] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *CVPR*, 2023.

[3] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

[4] Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[5] Shuai Bai, Keqin Chen, Xuejing Liu, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[6] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021.

[7] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

[8] Michael J Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. Bedlam: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *CVPR*, 2023.

[9] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. In *ICLR*, 2025.

[10] Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. Omni3D: A large benchmark and model for 3D object detection in the wild. In *CVPR*, 2023.

[11] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In *IROS*, 2025.

[12] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012.

[13] ByteDance Seed. Seed2.0 model card: Towards intelligence frontier for real-world complexity. *arXiv preprint arXiv:2607.00248*, 2026.

[14] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020.

[15] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

[16] Zhongang Cai, Ruisi Wang, Chenyang Gu, et al. Scaling spatial intelligence with multimodal foundation models. In *CVPR*, 2026.

[17] 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. *arXiv preprint arXiv:1709.06158*, 2017.

[18] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. *arXiv preprint arXiv:2401.12168*, 2024.

[19] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[20] DeepSeek-AI et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

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

[22] Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Peihao Wang, Huaizhi Qu, Shijie Zhou, Dilin Wang, Zhicheng Yan, Hongyu Xu, Justin Theiss, Tianlong Chen, Jiachen Li, Zhengzhong Tu, Zhangyang Wang, and Rakesh Ranjan. VLM-3R: Vision-language models augmented with instruction-aligned 3D reconstruction. In *CVPR*, 2026.

[23] Michael Fonder and Marc Van Droogenbroeck. Mid-air: A multi-modal dataset for extremely low altitude drone flights. In *CVPR Workshops*, 2019.

[24] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[25] Jose L Gómez, Manuel Silva, Antonio Seoane, Agnès Borrás, Mario Noriega, Germán Ros, Jose A Iglesias-Guitian, and Antonio M López. All for one, and one for all: Urbansyn dataset, the third musketeer of synthetic driving scenes. *Neurocomputing*, 637:130038, 2025.

[26] Google DeepMind. Gemini 2.5 Flash and native capabilities—audio & image: Model card. Model card, 2025.

[27] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *CVPR*, 2024.

[28] Junlin Han, Shengbang Tong, David Fan, Minghao Chen, Philip Torr, Filippos Kokkinos, and Mike Lewis. Towards physics of multimodal pretraining: Knowledge flow, modality synergy, early unification, and recipes. *arXiv preprint arXiv:2608.05000*, 2026.

[29] Jing He, Haodong Li, Wei Yin, Yixun Liang, Leheng Li, Kaiqiang Zhou, Hongbo Zhang, Bingbing Liu, and Ying-Cong Chen. Lotus: Diffusion-based visual foundation model for high-quality dense prediction. In *ICLR*, 2025.

[30] 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. In *ICLR*, 2022.

[31] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *TPAMI*, 46(12):10579–10596, 2024.

[32] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *CVPR*, 2018.

[33] Weikai Huang, Jieyu Zhang, Sijun Li, Taoyang Jia, Jiafei Duan, Yunqian Cheng, Jaemin Cho, Matthew Wallingford, Rustin Soraki, Chris Dongjoo Kim, Shuo Liu, Donovan Clay, Taira Anderson, Winson Han, Ali Farhadi, Bharath Hariharan, Zhongzheng Ren, and Ranjay Krishna. WildDet3D: Scaling promptable 3D detection in the wild. *arXiv preprint arXiv:2604.08626*, 2026.

[34] HyunJun Jung, Patrick Ruhkamp, Guangyao Zhai, Nikolas Brasch, Yitong Li, Yannick Verdie, Jifei Song, Yiren Zhou, Anil Armagan, Slobodan Ilic, Ales Leonardis, Nassir Navab, and Benjamin Busam. On the importance of accurate geometry data for dense 3D vision tasks. In *CVPR*, 2023.

[35] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Dynamicstereo: Consistent dynamic depth from stereo videos. In *CVPR*, 2023.

[36] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *CVPR*, 2024.

[37] Tobias Koch, Lukas Liebel, Friedrich Fraundorfer, and Marco Körner. Evaluation of CNN-based single-image depth estimation methods. In *ECCV Workshops*, 2018.

[38] Justin Lazarow, David Griffiths, Gefen Kohavi, Francisco Crespo, and Afshin Dehghan. Cubify anything: Scaling indoor 3D object detection. In *CVPR*, 2025.

[39] Hoang-An Le, Thomas Mensink, Partha Das, Sezer Karaoglu, and Theo Gevers. Eden: Multimodal synthetic dataset of enclosed garden scenes. In *WACV*, 2021.

[40] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[41] Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. Imagine while reasoning in space: Multimodal visualization-of-thought. *arXiv preprint arXiv:2501.07542*, 2025.

[42] Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. Matrixcity: A large-scale city dataset for city-scale neural rendering and beyond. In *ICCV*, 2023.

[43] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

[44] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[45] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *CVPR*, 2024.

[46] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023.

[47] Jianhui Liu, Haoze Sun, Wenbo Li, Yanbing Zhang, Rui Yang, Zhiliang Zhu, Yijun Yang, Shenghe Zheng, Nan Jiang, Jiaxiu Jiang, Haoyang Huang, Tien-Tsin Wong, Nan Duan, and Xiaojuan Qi. Openspatial: A principled data engine for empowering spatial intelligence. *arXiv preprint arXiv:2604.07296*, 2026.

[48] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

[49] Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *CVPR*, 2023.

[50] MiniMax. MiniMax H3: An open model breaking the boundaries between tasks and modalities. Official research blog, 2026.

[51] OpenAI. GPT-5.5 system card. Technical report, OpenAI, 2026.

[52] Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. Unidepthv2: Universal monocular metric depth estimation made simpler. *arXiv preprint arXiv:2502.20110*, 2025.

[53] Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[54] Pierluigi Zama Ramirez, Fabio Tosi, Matteo Poggi, Samuele Salti, Stefano Mattoccia, and Luigi Di Stefano. Open challenges in deep stereo: the Booster dataset. In *CVPR*, 2022.

[55] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *ICCV*, 2021.

[56] Thomas Schöps, Johannes L Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017.

[57] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[58] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *CVPR*, 2015.

[59] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020.

[60] Leonard Talmy. How language structures space. In *Spatial Orientation: Theory, Research, and Application*, pages 225–282. Plenum Press, 1983.

[61] Team Seedance, De Chen, Liyang Chen, Xin Chen, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

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

[63] Shengbang Tong, David Fan, Jiachen Zhu, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. Metamorph: Multimodal understanding and generation via instruction tuning. *arXiv preprint arXiv:2412.14164*, 2024.

[64] 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.

[65] Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. Smd-nets: Stereo mixture density networks. In *CVPR*, 2021.

[66] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z Dai, Andrea F Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R Walter, and Gregory Shakhnarovich. Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019.

[67] Haochen Wang, Anlin Zheng, Yucheng Zhao, Tiancai Wang, Zheng Ge, Xiangyu Zhang, and Zhaoxiang Zhang. Reconstructive visual instruction tuning. *arXiv preprint arXiv:2410.09575*, 2024.

[68] Kaixuan Wang and Shaojie Shen. Flow-motion and depth network for monocular stereo and beyond. *IEEE Robotics and Automation Letters*, 5(2):3307–3314, 2020.

[69] Qiang Wang, Shizhen Zheng, Qingsong Yan, Fei Deng, Kaiyong Zhao, and Xiaowen Chu. Irs: A large naturalistic indoor robotics stereo dataset to train deep models for disparity and surface normal estimation. In *ICME*, 2021.

[70] Qineng Wang, Baiqiao Yin, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, Saining Xie, Jiajun Wu, Li Fei-Fei, and Manling Li. Mindcube: Spatial mental modeling from limited views. *arXiv preprint arXiv:2506.21458*, 2025.

[71] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *arXiv preprint arXiv:2507.02546*, 2025.

[72] Weiyun Wang, Zhangwei Gao, Lixin Gu, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[73] Jialong Wu, Xiaoying Zhang, Hongyi Yuan, Xiangcheng Zhang, Tianhao Huang, Changjing He, Chaoyi Deng, Renrui Zhang, Youbin Wu, and Mingsheng Long. Visual generation unlocks human-like reasoning through multimodal world models. *arXiv preprint arXiv:2601.19834*, 2026.

[74] Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. Rgbd objects in the wild: Scaling real-world 3d object learning from rgb-d videos. In *CVPR*, 2024.

[75] Yicheng Xiao, Wenhu Zhang, Lin Song, Yukang Chen, Wenbo Li, Nan Jiang, Tianhe Ren, Haokun Lin, Wei Huang, Haoyang Huang, et al. Spatialedit: Benchmarking fine-grained image spatial editing. *arXiv preprint arXiv:2604.04911*, 2026.

[76] 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.

[77] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *NeurIPS*, 2024.

[78] Qian Yang, Ankur Sikarwar, Huy Le, Le Zhang, Zhuan Shi, Perouz Taslakian, and Aishwarya Agrawal. How and what to imagine? visual thinking in unified multimodal models for cross-view spatial reasoning. *arXiv preprint arXiv:2605.27310*, 2026.

[79] Rui Yang, Ziyu Zhu, Yanwei Li, Jingjia Huang, Shen Yan, Siyuan Zhou, Zhe Liu, Xiangtai Li, Shuangye Li, Wenqian Wang, et al. Visual spatial tuning. *arXiv preprint arXiv:2511.05491*, 2025.

[80] 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.

[81] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023.

[82] Songsong Yu, Yuxin Chen, Ying Shan, and Yanwei Li. Semantic generative tuning for unified multimodal models. *arXiv preprint arXiv:2605.18714*, 2026.

[83] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018.

[84] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[85] Yanbing Zhang, Bo Wang, Jianhui Liu, Nan Jiang, Jiaxiu Jiang, Haoze Sun, Yijun Yang, Shenghe Zheng, Lin Song, Haoyang Huang, Nan Duan, and Wenbo Li. Thinking with novel views: A systematic analysis of generative-augmented spatial intelligence. *arXiv preprint arXiv:2605.10588*, 2026.

[86] Zefeng Zhang, Xiangzhao Hao, Hengzhu Tang, Zhenyu Zhang, Jiawei Sheng, Xiaodong Li, Zhenyang Li, Li Gao, Daiting Shi, Dawei Yin, and Tingwen Liu. Cooper: A unified model for cooperative perception and reasoning in spatial intelligence. *arXiv preprint arXiv:2512.04563*, 2025.

[87] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

[88] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *ECCV*, 2020.

[89] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *CVPR*, 2017.

[90] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024.

[91] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

Appendix

To support a thorough understanding of our work, a detailed Appendix has been compiled. The table of contents below serves as a guide to the specific sections of interest.

Appendix Contents

A Qualitative Results **18**

B Details and Analysis of MMDC **24**
B.1 Depth sensitivity 24
B.2 Fusion weights 24
B.3 Implementation details 24

C Details of Adaptive Interleaved Reasoning **25**

D Data **29**
D.1 Metric Depth Data 29
D.2 Multi-view Data 29
D.3 Exo-to-Ego Generation Data 33

E Benchmark **33**
E.1 Multi-view and Exo-to-Ego Generation 33
E.2 3DCueBench 34

F Additional Experimental Results **39**
F.1 Generation Pretraining Followed by Understanding SFT 39
F.2 Visual and Textual Referring Formats 39
F.3 External Depth Tools 40

A Qualitative Results

Figure 9 shows predicted depth maps and their corresponding colored 3D reconstructions. The examples preserve recognizable scene layout, including the tiered seating in the lecture hall, tables in front of the courtyard facade, and the recessed doorway in the office. Depth boundaries follow many visible object contours, while the reconstructions also expose gaps and distortions near some boundaries. These examples illustrate recovered spatial structure without serving as a quantitative test of metric calibration.

Figures 10 and 11 compare the original BAGEL [21] with our models on multi-view and exo-to-ego generation; quantitative results are in Tables 7 and 8. The examples are chosen from random subsets of the evaluation sets (Appendix E.1): 90 multi-view pairs (30 each from RealEstate10K [91], DL3DV [45], and ScanNet++ [81]) and 64 of the 469 exo-to-ego pairs from Ego-Exo4D [27]. For multi-view generation, BAGEL often returns a view close to the input, even for instructions that move the camera by more than a metre or rotate it by more than 30°. Our model, trained on the pose-aligned instructions of Appendix D.2, moves the camera in the instructed direction and brings the expected content into view. For exo-to-ego generation, BAGEL mostly depicts the described person, or the object the person is working on, instead of rendering the scene from that person's viewpoint. Our model instead produces a first-person view, with hands, tools, and the surrounding layout roughly matching the ground truth. The generated views are not exact reconstructions: fine details, such as the arrangement of objects on the table, and the precise viewing direction often differ from the ground truth.

Figure 8 in Section 5.2 and Figures 12–14 show adaptive interleaved reasoning by GeoUG on 3DCueBench, CV-Bench-3D [62], and DA-2K [77]. Each trace is the policy's output for the first option order, abridged: tool-call arguments are omitted and tool results are reduced to the readings the policy uses. p_{gen} is the probability of `<GENERATE_DEPTH>` at `<DECIDE>`, reported for both option orders. The depth maps are those



Figure 9 Qualitative depth generation. Rows show input images, predicted depth maps, and colored 3D reconstructions.

the policy generated. The point clouds back-project them with the same pinhole camera that the tool uses when no intrinsics are given (focal length equal to the longer image side, centered principal point), and are shown from above with the camera at the bottom; points more than 0.3 m above the highest queried object are removed so that the ceiling does not hide the scene. Colored markers are the object centroids returned by the tool; for distance questions, the lines connect the anchor to each candidate.

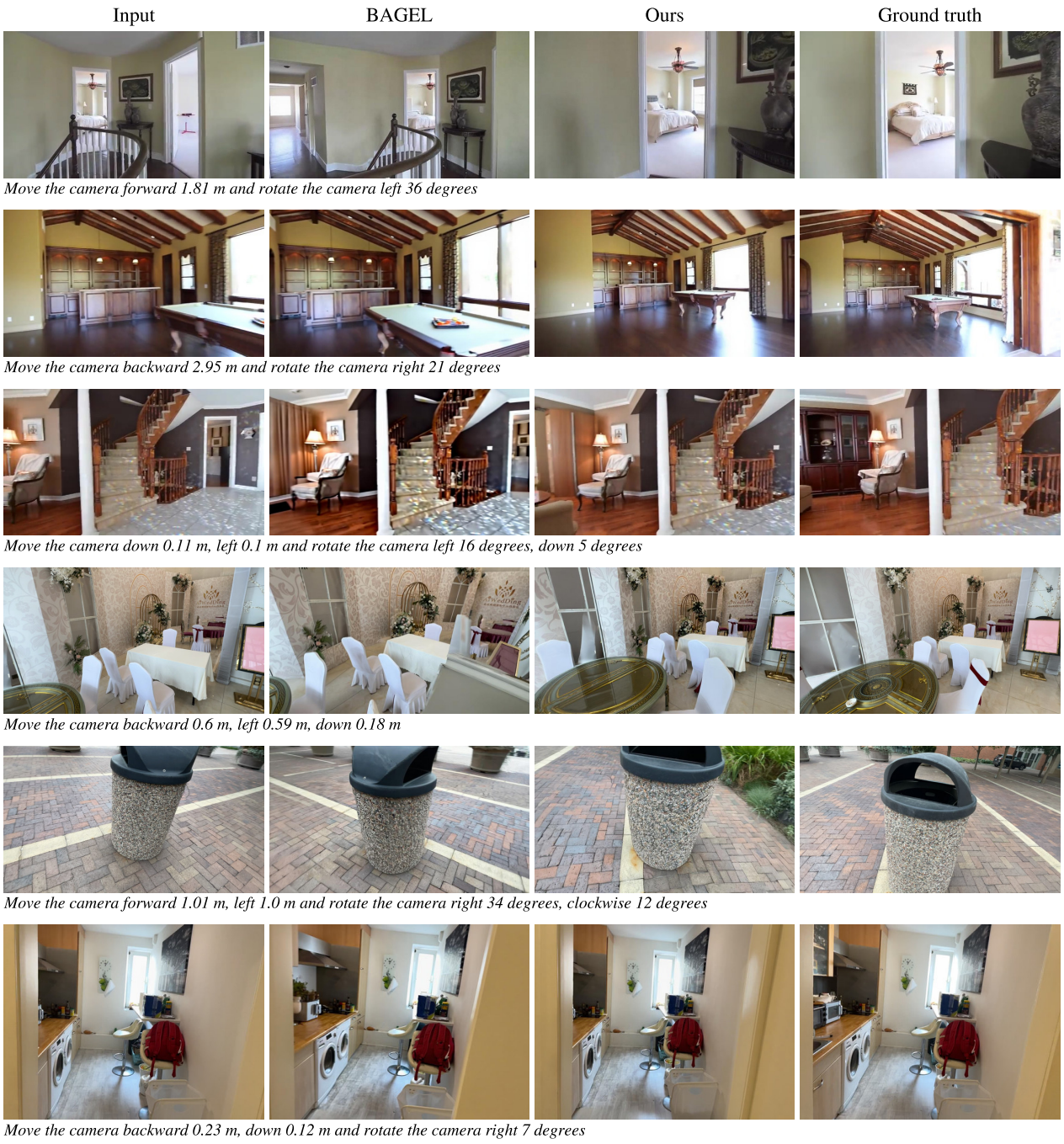


Figure 10 Qualitative multi-view generation with ground truth. Rows 1–3 are from RealEstate10K [91], rows 4–5 from DL3DV [45], and row 6 from ScanNet++ [81]. BAGEL [21] is the original model. The camera-motion instruction is given below each row. Ours is the multi-view co-training model (Figure 6) at its final checkpoint.

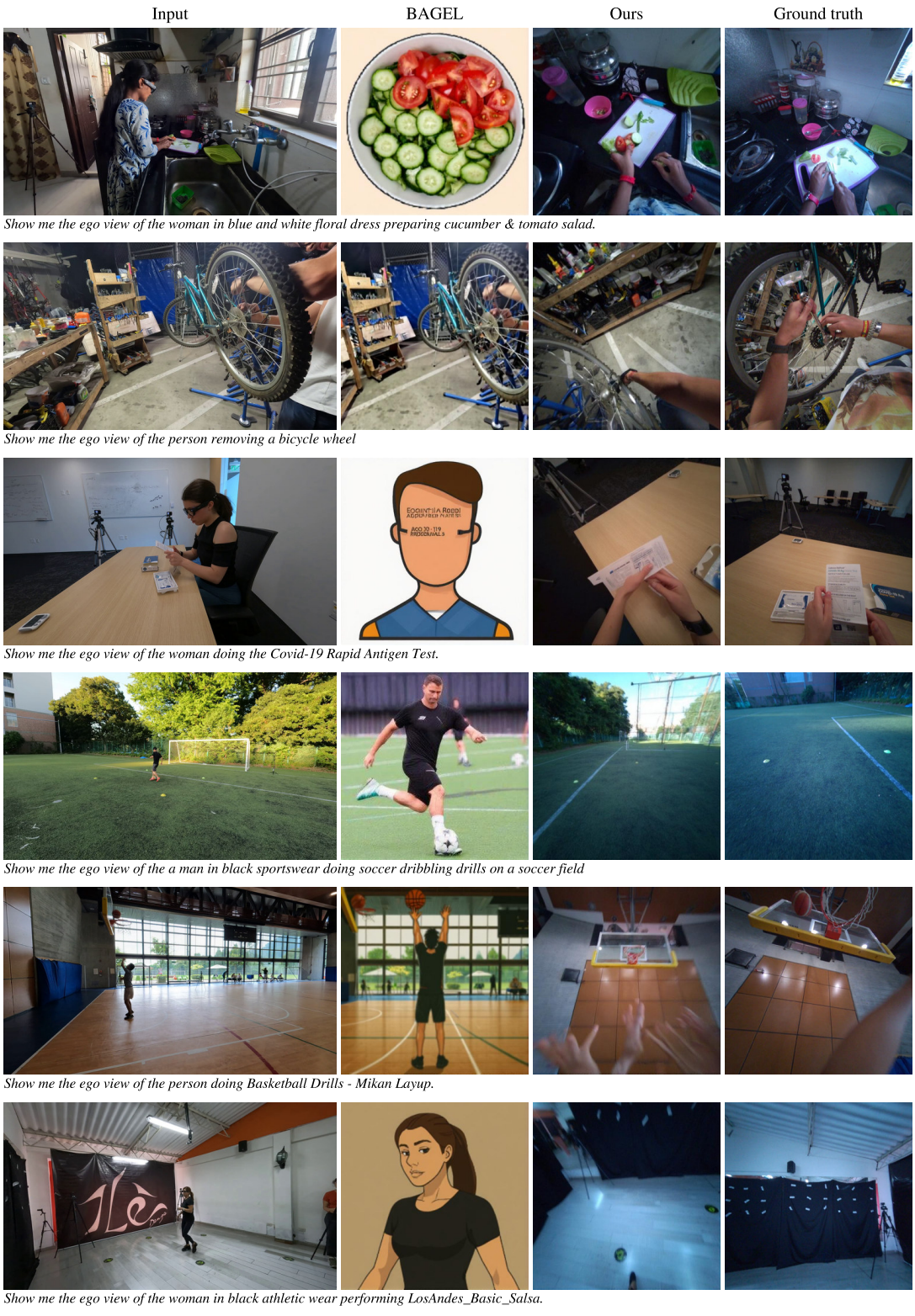
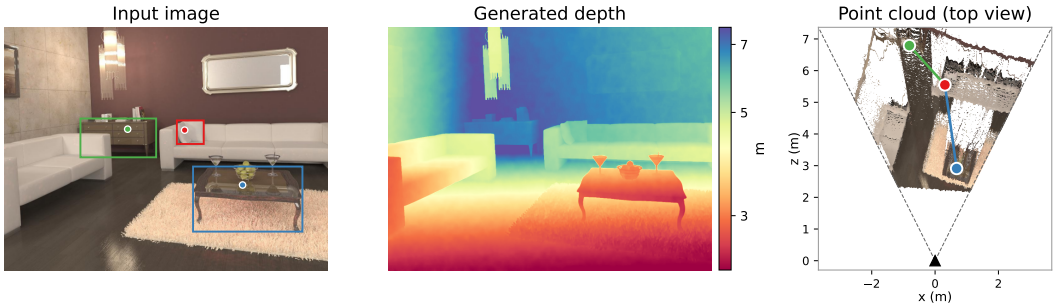


Figure 11 Qualitative exo-to-ego generation on Ego-Exo4D [27] with ground truth. Given an exo image and an instruction naming the person, the model generates that person's ego view. BAGEL [21] is the original model. Ours is the exo-to-ego co-training model (Figure 6) at its final checkpoint. Instructions are shown verbatim.



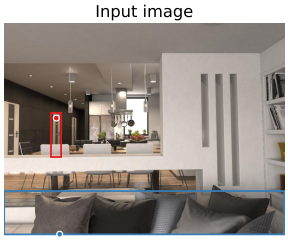
Question. Estimate the real-world distances between objects in this image. Which object is closer to the **white cushion**, the **glass coffee table** or the **wooden sideboard**? A. glass coffee table B. wooden sideboard

GeoUG. To answer this, I need the 3D distance from the anchor to each candidate. [boxes and points for the three objects] My initial answer without depth: A. glass coffee table. `<DECIDE><GENERATE_DEPTH>` $p_{\text{gen}} = 0.99/0.99$

[generated depth map, 2.18–9.34 m] `query_3d_distance:` **white cushion–glass coffee table** 2.74 m, **white cushion–wooden sideboard** 1.68 m.

This asks which object is closer to the white cushion — that is the distance between objects, not the distance from the camera . . . Since $1.68 < 2.74$, the wooden sideboard is closest to the white cushion. **Answer: B. wooden sideboard** ✓

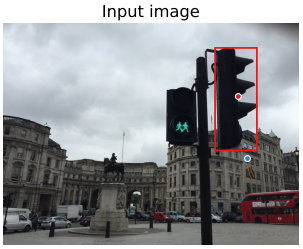
Figure 12 **3DCueBench, Distance**, relative gap in $[0.10, 0.16)$. The draft picks the candidate nearer to the camera. After generating depth, the policy compares 3D distances between objects rather than camera depths; both option orders are answered correctly.



Question. Which object is closer to the camera taking this photo, the **door** or the **pillow**? A door B pillow

GeoUG. [a point inside each given box] My initial answer without depth: B pillow. `<DECIDE>` $p_{\text{gen}} = 0.02/0.05$

Answer: B pillow ✓

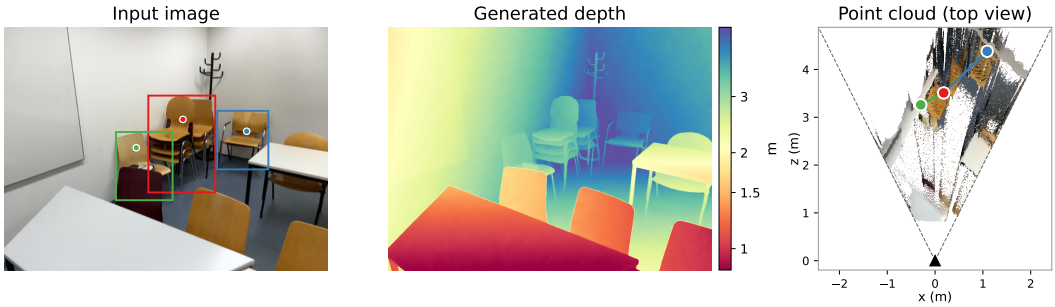


Question. Which object is closer to the camera taking this photo, the **first point** or the **second point**? A. first point B. second point

GeoUG. [a box around each point] My initial answer without depth: A. first point. `<DECIDE>` $p_{\text{gen}} = 0.02/0.13$

Answer: A. first point ✓

Figure 13 **Answering without generation.** Left: CV-Bench-3D [62]. Right: DA-2K [77]. The draft is correct and the policy answers directly after `<DECIDE>`; both option orders are answered correctly without generating depth.



Question. Estimate the real-world distances between objects in this image. Which object is closer to the **stacked chairs**, the **wooden armchair** or the **wooden chair**? A. wooden armchair B. wooden chair

GeoUG. To answer this, I need the 3D distance from the anchor to each candidate. [boxes and points for the three objects] My initial answer without depth: A. wooden armchair. `<DECIDE><GENERATE_DEPTH>` $p_{\text{gen}} = 0.87/0.88$

[generated depth map, 0.81–6.20 m] `query_3d_distance`: **stacked chairs**–**wooden armchair** 1.25 m, **stacked chairs**–**wooden chair** 0.63 m.

. . . Since $0.63 < 1.25$, the wooden chair is closest to the stacked chairs. **Answer: B. wooden chair** × (ground truth: A)

Figure 14 Failure case: 3DCueBench, Distance, relative gap in $[0.25, 0.40)$. The draft is correct, but the policy generates depth and follows the tool, whose measured distances place the wooden chair closer to the anchor. Both option orders are answered incorrectly.

B Details and Analysis of MMDC

B.1 Depth sensitivity

For a channel with pivot c_i, let $u_i = \log(d/c_i)$ measure proportional changes in depth. The encoding's sensitivity and local error amplification are

$$s_i(d) = \frac{\partial Y_i}{\partial u_i} = \frac{2c_i d}{(d+c_i)^2} = \frac{1-Y_i^2}{2}, \qquad \frac{\Delta d}{d} \approx \frac{\Delta Y_i}{s_i(d)}. \tag{8}$$

A first-order expansion of the inverse gives the second expression: the same encoding error produces a larger relative depth error when sensitivity is low. Sensitivity peaks at $d = c_i$ and decreases towards both smaller and larger depths (Figure 15a).

The encoding can also be written as $Y_i = \tanh(u_i/2)$. Changing the pivot therefore translates the sensitivity profile along the log-depth axis. Our geometrically spaced pivots $(2, 8, 32)$ m place these profiles at equal log-depth intervals, providing overlapping precision ranges. The pivots remain fixed across datasets.

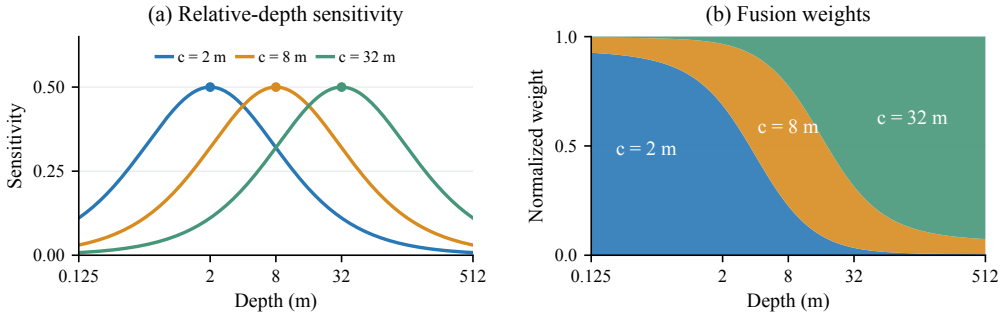


Figure 15 MMDC sensitivity and fusion. (a) Channel sensitivity. (b) Normalized fusion weights. Both are evaluated on noiseless encoded depths.

B.2 Fusion weights

We use the sensitivity analysis to derive a local approximation to the fusion weights. At one pixel, assume small, independent, zero-mean encoding errors with equal variance σ^2: $\hat{Y}_i = Y_i + \varepsilon_i$. Away from the clipping boundaries, equation 8 gives

$$v_i \equiv \mathrm{Var}\left(\frac{\hat{d}_i - d}{d}\right) \approx \frac{\sigma^2}{s_i(d)^2} = \frac{4\sigma^2}{(1-Y_i^2)^2}. \tag{9}$$

For a weighted estimate $\hat{d} = \sum_i a_i \hat{d}_i$, with $\sum_i a_i = 1$, the relative-error variance is approximately $\sum_i a_i^2 v_i$. Minimizing it yields inverse-variance weights:

$$a_i = \frac{v_i^{-1}}{\sum_j v_j^{-1}}, \qquad w_i \propto s_i(d)^2 \propto (1-Y_i^2)^2. \tag{10}$$

Channels with higher sensitivity receive greater weight (Figure 15b). In practice, we substitute the clipped predictions $\hat{Y}_i$ for the unknown noiseless values Y_i, giving equation 3. This motivates the decoder as a first-order approximation under the stated noise assumptions, rather than an exact optimum for arbitrary generation errors.

B.3 Implementation details

Depths and pivots are expressed in metres. The reference implementation marks positive depths as valid and intersects them with an optional validity mask; invalid entries are filled with zero, with the mask retained separately. Before inversion, predicted channels are clipped to $[-1 + 10^{-4}, 1 - 10^{-4}]$. Fusion weights are floored at 10^{-12}, and the decoded depth has a configurable upper bound of 10^4 m by default. These operations keep the inverse and weighted average numerically well-defined near saturation.

C Details of Adaptive Interleaved Reasoning

Pilot evaluation protocols. Both pilot studies report ordinary (non-circular) accuracy. Figure 3 compares separately trained Direct and Always-generate checkpoints. Within each task, both datasets use five fixed relative-gap bins, drawn at equal spacing. Depth bins start at 0.10, 0.16, 0.25, 0.40, and 0.55; distance bins at 0.10, 0.25, 0.40, 0.60, and 0.80. Intervals are left-closed and right-open, except that the last includes 1.00. Bins with fewer than ten questions are shaded and their estimates omitted; bands are pointwise 95% Wilson intervals.

In Figure 4b, we rank the 800 3DCueBench questions by generation probability (descending), or by the ground-truth relative gap (ascending), and use generated answers for the top fraction and direct answers for the remainder. We report the fraction of the always-generate gain retained,

$$\frac{\mathrm{Acc} - \mathrm{Acc}_{\mathrm{direct}}}{\mathrm{Acc}_{\mathrm{always}} - \mathrm{Acc}_{\mathrm{direct}}}.$$

Direct and generated answers come from the same checkpoint within each curve; the before-answer and draft-conditioned policies use different checkpoints and supervised training. Ties are averaged under uniform random ordering. Random selection follows the diagonal in expectation. Figure 4c plots generation rates during RL with reward $c - \lambda a$. Each update uses 16 questions; the plotted rates use an exponential moving average over questions with smoothing factor 0.97. The proposed policy uses the 1500-step supervised initialization, and its forced-generation rollouts are excluded from the plotted rate. The before-answer and draft-conditioned settings differ in model, training questions, and credit assignment; this comparison does not isolate decision position alone.

Pilot analysis of the before-answer policy. The before-answer policy is trained on a format in which the decision token follows a short, largely templated reasoning preamble and both branches are supervised with equal frequency; the before-answer results in Fig. 4b and c use this policy. Its generation probability took two values across the 800 3DCueBench questions (0.469 and 0.500), and the resulting gate was 0.6 points above a random gate at the same generation rate. Hidden states at the decision token were nearly identical across questions (median pairwise cosine 0.998), and a linear probe predicting whether depth helps reached an AUC of 0.536, compared with 0.513 for shuffled labels. With symmetric supervision, the maximum-likelihood gate is a constant. Under reinforcement learning (Fig. 4c), $\lambda = 0$ drove the final generation rate to 0.96, and $\lambda \in \{0.05, 0.07, 0.1\}$ to at most 0.005; evaluated accuracy followed the random-gate line (residual standard deviation 0.025, below the per-point standard error of 0.05). In these runs, changing the cost mainly changes the overall generation rate, without producing an effective question-selection policy. Ranking by the ground-truth gap in Fig. 4b uses the draft-conditioned policy's outcomes; each curve is normalized by its own policy's direct and always-generate accuracies (before-answer 82.5% and 90.0%, draft-conditioned 81.0% and 91.1%).

Advantage scale. Standard GSPO [87] divides by the within-group standard deviation s_r (Eq. equation 5). With a generation cost, this normalization removes the magnitude of λ in a second way. In a group where every rollout answers correctly but only some generate, $r_i = 1 - \lambda a_i$ and

$$A_i^{\mathrm{group}} = \frac{-\lambda(a_i - \bar{a})}{\lambda s_a + \epsilon} \approx -\frac{a_i - \bar{a}}{s_a},$$

where s_a is the standard deviation of a_i within the group. Any $\lambda > 0$ then yields the same full-magnitude advantage against generating, whereas in groups whose correctness varies the cost only shifts the advantages by λ / s_r. Because all-correct groups are frequent, this term dominates the decision's update. We therefore divide all advantages by a constant $s = 0.5$, the standard deviation of a Bernoulli variable with mean 0.5. Every advantage is then linear in λ, and groups whose rewards differ only by the cost receive proportionally small updates.

Failure of trajectory-level advantages. With one group-normalized advantage per trajectory, $A_i = (r_i - \bar{r})/s$, consider a group in which every rollout generates. Then $r_i = c_i - \lambda$ for all i, so $A_i = (c_i - \bar{c})/s$ does not

depend on λ: the cost no longer pushes the decision, whose update reflects only which generating rollouts happened to answer correctly. Once the policy generates on nearly every rollout, nothing pushes it to stop. In addition, a rollout that generates although its draft was already correct receives a positive advantage whenever $\bar{r} < 1 - \lambda$, and wrong draft tokens share the positive advantage when depth corrects them. In our first runs with this objective (500-step initialization), the generation rate rose from 0.64 to 0.99 within about ten updates for $\lambda = 0.05$, and drafts drifted (18 of 63 drafts without an answer letter after eight updates, versus 7 of 64 at the start).

Segment-wise advantages. Each trajectory is split into four segments: grounding tokens (localization), draft tokens (from "My initial answer without depth:" through `<DECIDE>`), the decision token, and execution tokens (depth request, geometric queries, final answer). With G rollouts of one question, let $\mathcal{D}$ be the rollouts that generated and $\mathcal{F} \subset \mathcal{D}$ those forced to generate.

- Grounding tokens: $c_i - \bar{c}_{\mathcal{D}}$ for $i \in \mathcal{D}$ and 0 otherwise.
- Draft tokens: 0, so that reinforcement learning does not sharpen the draft confidence the decision reads; in the supervised stage, the draft answer also receives no loss.
- Decision token: equation 7 for $i \notin \mathcal{F}$ and 0 for $i \in \mathcal{F}$. $\widehat{Q}_{\text{gen},i}$ is the mean correctness of forced rollouts with the same draft correctness as rollout i, falling back to all forced rollouts if that cell is empty. For exact conditional branch values, $Q_{\text{dir},i} = c_{d,i}$ and $V_i = p_i(Q_{\text{gen},i} - \lambda) + (1 - p_i)Q_{\text{dir},i}$ yield the two action advantages in equation 7. Their expected unclipped score-function update is $p_i(1 - p_i)(Q_{\text{gen},i} - \lambda - c_{d,i})$ with respect to the decision logit. In practice, the pooled forced rollouts approximate the conditional generation value. They are selected uniformly at random to avoid relying only on rollouts whose policy chose generation; grouping by draft correctness does not guarantee an unbiased estimate for every distinct reasoning prefix. p_i is read from the sampled token's log-probability; the two action tokens carry at least 99.6% of the probability mass at the decision token.
- Execution tokens: $c_i - \bar{c}_{\mathcal{D}}$ for $i \in \mathcal{D}$; for direct rollouts the final answer equals the draft and the advantage is 0.

All advantages are divided by a constant scale of 0.5. The policy loss uses clipped sequence-level importance ratios [87], one per segment, with each segment weighted by its token count.

Draft-conditioned supervised data. Supervised sequences are built from comparison questions generated by our data engine on COCO [44] and ADE20K [89] images with DA3 [43] depth. Drafts come from an intermediate checkpoint of the same before-answer training run, which also initializes the draft-conditioned supervised stage. For each question, we feed the image, the question, and the reasoning prefix of the training sequence followed by "My initial answer without depth:", and take the option letter with the highest next-token probability. Drafts thus reflect the beliefs of the model that supervised training starts from, and greedy selection matches greedy decoding at inference. Drafts are wrong on about 22% of the questions. Each question yields up to three sequences sharing the prefix up to `<DECIDE>` (Fig. 16): (i) a correct draft answered directly; (ii) a correct draft followed by depth generation, geometric queries, and the answer; and (iii) a wrong draft followed by depth generation and the correct answer. Draft tokens receive no loss. The decision token is supervised only in (i), as stopping, and in (iii), as generating; in (ii) it is masked, so that sequences with identical context do not pull it toward 0.5, while the rest of (ii) teaches how to use depth. A wrong draft is never answered directly, and generation follows both correct and wrong drafts, so that generating does not simply mean changing the answer. Keeping one quarter of the direct sequences balances the supervised decisions. After removing near ties (relative gap below 5%), the set contains 262K questions and 312K sequences.

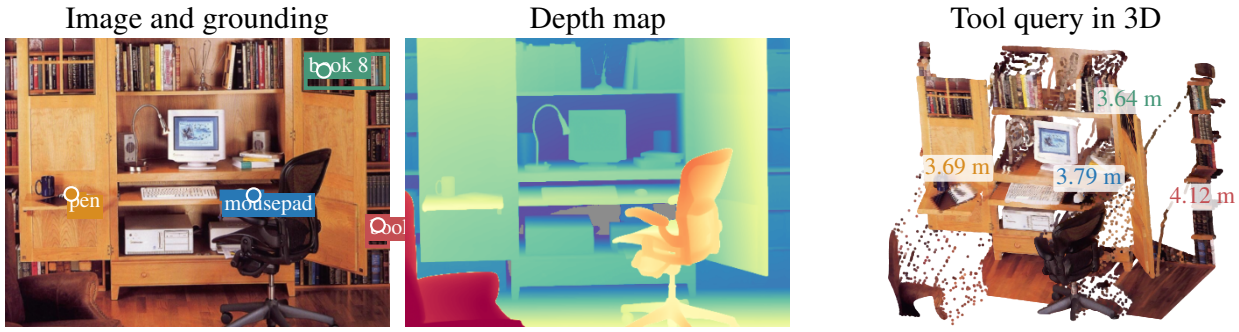
Reinforcement-learning training questions. Reinforcement learning uses questions generated by our data engine on ScanNet++ [81] (indoor) and nuScenes [15] (outdoor) images. Labels come from sensor depth measured on object surfaces rather than box centers. We remove questions that name annotation artifacts or large structural surfaces (walls, floors, ceilings, doors, windows), whose distance is ill-defined. For object-to-object questions, we keep only those where 2D cues, such as image-plane distances between objects, point to

a wrong answer; object-to-camera questions are kept regardless. We keep at most two questions per image, preferring those with the most misleading 2D cues within each question type, and sample a pool balanced between indoor and outdoor scenes and across question types, split by image into training and validation. The 800 training questions contain one question per image and are balanced across the two sources, four question types, nearer versus farther targets, and answer letters.

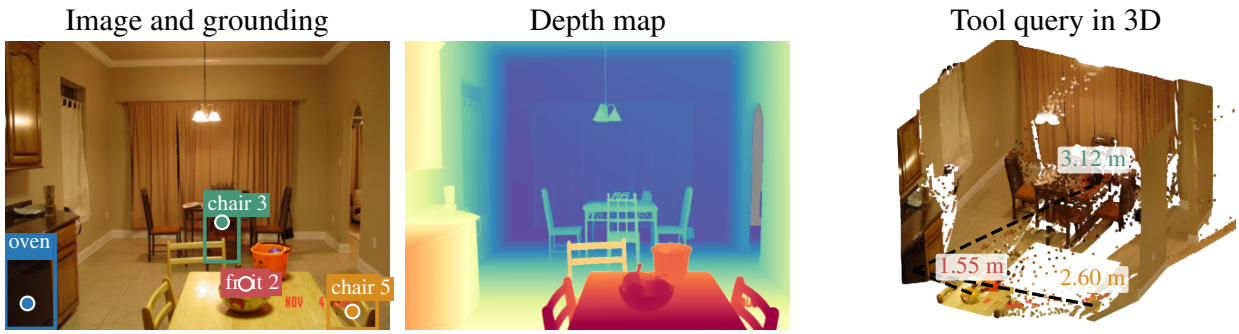
Training setup. The draft-conditioned supervised stage trains for 3000 steps. The policy is initialized from its 1500-step checkpoint, which has the most accurate drafts among 500–3000 steps. We train LoRA [30] adapters (rank 32) on the language model with learning rate 5×10^{-5}, 16 rollouts per question of which 4 are forced to generate, and 16 questions per update, for one epoch over the 800 questions (50 updates).



(a) *Which object is closer to the camera, the laptop 2 or the person?*
To answer this, I need each object's metric distance from the camera. [boxes and points] My initial answer without depth: A. laptop 2 <DECIDE></think> A. laptop 2



(b) *Which one is farthest from the camera: the mousepad, the pen, the book 8, or the book 12?*
To answer this, I need each object's metric distance from the camera. [boxes and points] My initial answer without depth: C. book 8 <DECIDE><GENERATE_DEPTH> [depth map] [query the depths of the four objects: 3.79 m, 3.69 m, 3.64 m, 4.12 m] The book 12 is farthest from the camera. </think> D. book 12



(c) *Which of these is closest to the oven: the fruit 2, the chair 3, or the chair 5?*
To answer this, I need the 3D distance from the anchor to each candidate. [boxes and points] My initial answer without depth: C. chair 5 <DECIDE><GENERATE_DEPTH> [depth map] [query distances from the oven: chair 5, 2.60 m; chair 3, 3.12 m; fruit 2, 1.55 m] The fruit 2 is closest to the oven. </think> A. fruit 2

Figure 16 Training sequences of the draft-then-decide format. (a) A confident, correct draft answered directly (form (i)). (b, c) Wrong drafts checked with depth (form (iii)) for a camera-depth and an object-to-object question. Panels show the image with the queried boxes and points, the depth map the model learns to generate, and the query in 3D: the depth map is back-projected with the camera intrinsics, and markers show the positions returned by the geometric query, labeled with their depths (b) or distances to the anchor (c). The draft answer after "My initial answer without depth:" receives no loss; geometric queries and their results are summarized in brackets.

D Data

Table 6 summarizes the sources of instruction data used for different tasks. Below, we outline the data processing details for specific tasks.

Task	Data sources
Relative depth estimation	Hypersim [55], VirtualKitti [14], UrbanSyn [25], UnrealStereo4K [65], Structured3D [88], Spring [49], MidAir [23], MatrixCity [42], IRS [69], GTA-SfM [68], EDEN [39], DynamicReplica [35], Bedlam [8], MVS-Synth [32]
Metric depth estimation	Scannet++ [81], Matterport3D [17], Arkitscenes [6], Taskonomy [83], Wildrgbd [74], Cityscapes [19], Waymo [59], AgiBotWorld [11]
Multi-view generation	RealEstate10K [91], DL3DV-10K [45]
Exo-to-ego generation	Ego-Exo4D [27]
Visual instruction following	LLaVA-OneVision [40], VST [79]

Table 6 Data source statistics.

D.1 Metric Depth Data

Real Data Refinement. Real training data for the metric depth estimation are commonly derived from either LiDAR scans or Structure-from-Motion (SfM) reconstructions: LiDAR suffers from sensor–camera synchronization errors that corrupt depth along object boundaries, while SfM fails on reflective surfaces, thin structures, and sharp discontinuities, leaving holes or oversmoothed regions. A depth estimator trained on synthetic data avoids both issues—color and depth are exactly registered by construction and fine local geometry is preserved—but it predicts relative depth only up to an unknown affine transform. We therefore use such a relative depth estimator [43] as a teacher and treat its output as a dense pseudo label that repairs the real measurement, after first removing this affine ambiguity.

Concretely, we fit a single scale a and shift b that take the relative depth $\tilde{D}$ into the metric frame of the real depth D. Let $m_i \in \{0, 1\}$ be a validity mask indicating which pixels carry a usable real measurement. The two scalars are determined by a masked least-squares fit,

$$(\hat{a}, \hat{b}) \;=\; \arg\min_{a,\,b} \sum_{i=1}^{N} m_i \left(a\,\tilde{D}_i + b - D_i\right)^2, \qquad \hat{D} \;\triangleq\; \hat{a}\,\tilde{D} + \hat{b}, \tag{11}$$

which has a unique closed-form solution as long as the valid pixels span more than one depth value. The aligned depth $\hat{D}$ matches the metric scale of D on the supervised pixels while inheriting the dense, sharp geometry of the relative depth estimator, and it is $\hat{D}$ that enters our subsequent training objectives.

D.2 Multi-view Data

Instruction–View Alignment. Camera-motion instructions should describe the transformation between the source and target views. An instruction that specifies only rotation can omit translation present in the paired images. For example, the SpatialEdit pair in Figure 18 is labeled as *Pan Left*, although the camera also undergoes *Truck Right*. The original BAGEL shows the same inconsistency at generation time: Figure 19 shows an output that rotates right and shifts sideways when a left rotation is requested. We hypothesize that such misaligned pairs in existing training data contribute to this failure, since they associate the stated motion with visual changes caused by a different transformation. We therefore construct instructions from camera poses so that they account for both rotation and translation. Figure 20 shows the resulting training pairs.

Scale Unification. Since the camera poses in RealEstate10K [91] and DL3DV [45] are extracted using SfM-like methods, they inherently lack a consistent physical metric scale. This scale ambiguity prevents us

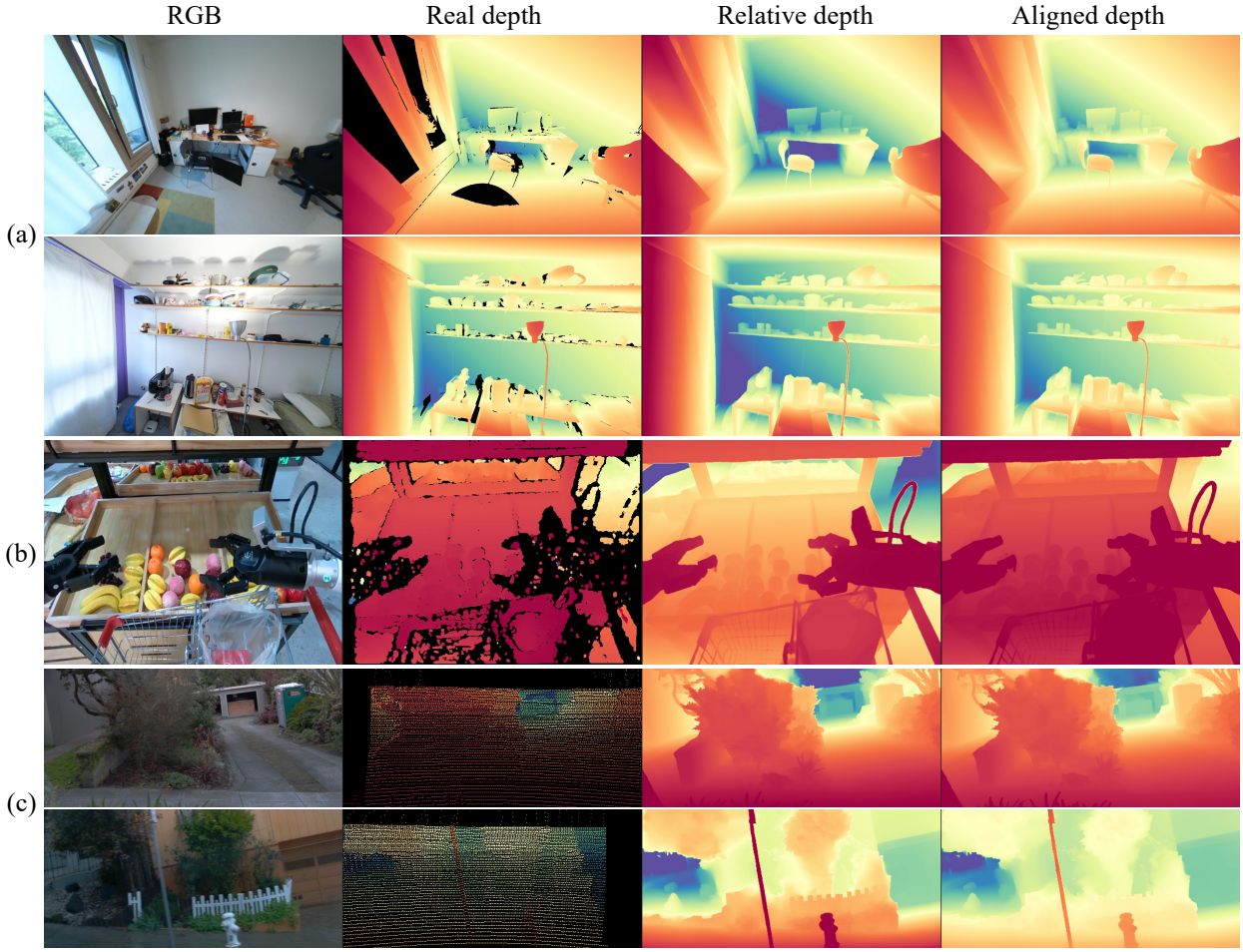


Figure 17 Relative-to-metric depth alignment. (a) ScanNet++ [81]; (b) AgiBotWorld [11]; (c) Waymo [59].



Move the camera. Camera rotation: Yaw 45.0°, Pitch 0.0°
×: The camera also moves rightward

Move the camera. Camera rotation: Yaw 0.0°, Pitch -75.0°
×: The camera also moves upward

Figure 18 SpatialEdit instruction examples [75]. Translation is omitted from the camera instructions.

from directly mapping camera translations to text instructions. For example, “move forward by 1 meter" could equal 1 unit in one scene but 10 units in another scene. To address this inconsistency, we re-estimate all camera poses using DA3 [43] to unify the scale information. Thus, for a video, we get the world-to-camera poses: $\{P_i = (R_i, t_i)\}_{i=1}^{N}$.

Instruction Construction. Our pose-based data engine expresses relative motion in the source camera’s coordinate frame and uses its rotation and translation to construct the instruction. Each training example consists of a source image, the resulting motion instruction, and the paired target image. Given a video paired with the camera poses, we label every short segment of a clip with a few words from a simple vocabulary *Dolly / Truck / Pedestal / Pan / Tilt / Roll* (each signed), plus *Stay* for near-static cases. The motion between



Figure 19 Instruction misalignment. Asked to rotate left by 30°, BAGEL instead rotates the camera about 21° right and moves it about 1 m left. BEV: input (black), instructed (blue), and generated (red) cameras.



Figure 20 Our multi-view samples. Instructions describe both rotation and translation after scale unification.

frames a and b, written in camera-a's local frame, is

$$\mathbf{d}_{ab} = R_a(c_b - c_a), \quad \boldsymbol{\omega}_{ab} = \mathrm{Euler}_{yxz}(R_b R_a^\top), \quad c_k = -R_k^\top t_k; \tag{12}$$

each component is thresholded to decide whether a given atomic motion fires (e.g. a sufficiently positive d_z triggers *Dolly In*).

A single frame gap is brittle: short gaps are dominated by pose noise, long ones blur neighbouring motions. We therefore smooth the trajectory with a quaternion-aware EMA (flipping the sign to stay on the shorter arc) and repeat the atomic extraction over a grid $(\Delta, \alpha) \in \mathcal{G}$ of gaps and smoothing strengths, with thresholds rescaled by $\sqrt{\Delta}$ so that every scale contributes comparable votes. For each unit interval we pool these votes, drop the loser of any opposite pair (e.g. *Pan Left*/*Right*), and keep the top label alone if it leads by $3\times$, otherwise every label within half of the leader, capped at four.

Neighbouring intervals with identical labels are merged into segments $[a, b)$; to obtain multi-scale supervision we further enumerate every sub-interval inside each segment. For each one we recompute the exact magnitudes $\mathbf{m}$ from the *unsmoothed* poses and project them onto the voted labels (e.g. the magnitude of *Dolly In* is simply d_z). Three light filters then clean the result: a *consistency* check removes labels whose recomputed direction contradicts the vote, an *absolute* check drops per-frame magnitudes below the sensor-noise level, and a *relative* check, applied separately to translation and rotation, discards secondary motions weaker than ρ times the dominant one. Segments whose strongest motion still falls below a global cut-off are dropped entirely, leaving the final supervision $\{([a, b), \mathcal{Y}, \mathbf{m})\}$ summarized in Algorithm 1.

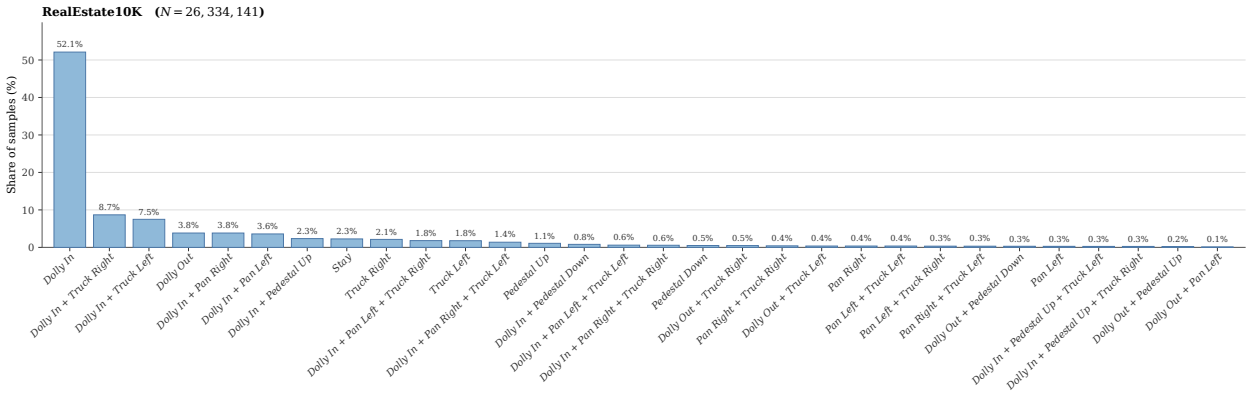
We sub-sample with stride $s_0 = 3$ and set $\mathcal{G} = \{1, 3, 5\} \times \{0.03, 0.05, 0.1\}$, $(\tau_t, \tau_r) = (0.01, 1.5°)$, $(\tau_t^{\mathrm{abs}}, \tau_r^{\mathrm{abs}}) = (0.006, 0.35°)$, $\rho = 0.3$, and $(\tau_t^{\min}, \tau_r^{\min}) = (0.01, 1.5°)$; rotation thresholds further carry a scaling constant $\kappa = 1.8$.

Instruction Distribution. We converted the data from RealEstate10K [91] and DL3DV [45] into camera instructions and observed noticeable biases in their distributions, as shown in Figure 21. For instance, the dominant instruction in RealEstate10K is *Dolly In*, since this dataset largely consists of room-tour videos. In

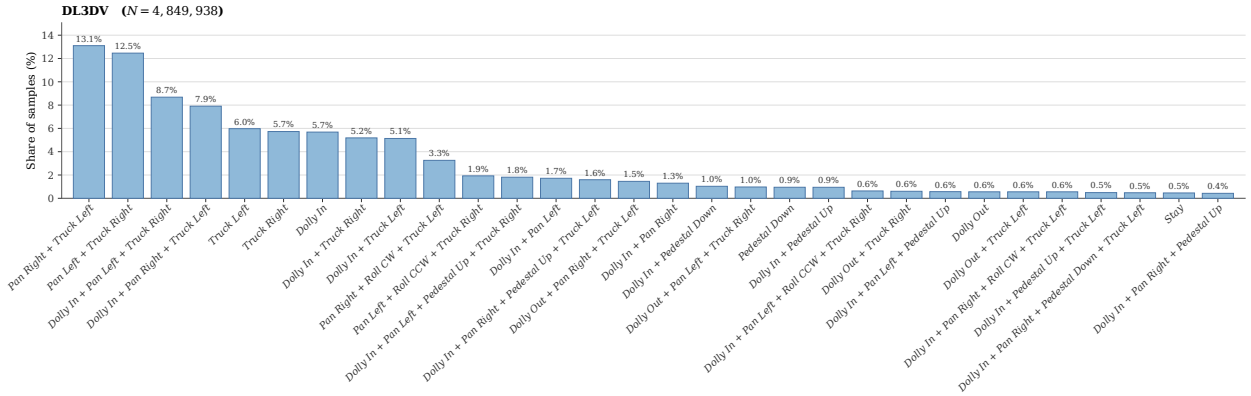
Algorithm 1: Pose → Instruction

Input: Poses $\{P_i\}$; grid $\mathcal{G}$; thresholds $\tau, \tau^{\text{abs}}, \rho, \tau^{\min}$
Output: Labeled segments $\mathcal{S} = \{([a,b), \mathcal{Y}, \mathbf{m})\}$

$P^\star \leftarrow \text{QuatEMA}(P; \alpha_0)$ `// reference trajectory`
foreach $(\Delta, \alpha) \in \mathcal{G}$ **do** $L^{\Delta,\alpha} \leftarrow \text{Thresh}\Big(\Delta\text{Pose}(\text{QuatEMA}(P;\alpha), \Delta);\ \tau\sqrt{\Delta}\Big)$
$\{y_i\} \leftarrow \text{TopK} \circ \text{Mutex} \circ \text{Vote}\Big(\bigcup_{\Delta,\alpha} L^{\Delta,\alpha}\Big)$
$\{[a_k, b_k), \mathcal{Y}_k\} \leftarrow \text{Merge}(\{y_i\})$
foreach *sub-interval* $[a,b) \subseteq [a_k, b_k)$ **do**
  $\mathbf{m} \leftarrow \text{Project}(\Delta\text{Pose}(P^\star_a, P^\star_b),\ \mathcal{Y}_k)$
  $\mathcal{Y} \leftarrow \text{Filter}_{\text{cons}\wedge\text{abs}\wedge\text{rel}}\big(\mathcal{Y}_k, \mathbf{m}; \tau^{\text{abs}}, \rho\big)$
  if $\mathcal{Y} \neq \emptyset \ \wedge\ \max|\mathbf{m}| \geq \tau^{\min}$ **then** $\mathcal{S} \leftarrow \mathcal{S} \cup \{([a,b), \mathcal{Y}, \mathbf{m})\}$
end
return $\mathcal{S}$



(a) Camera instruction distribution of RealEstate10K [91]



(b) Camera instruction distribution of DL3DV [45]

Figure 21 Camera instruction distribution in RealEstate10K [91] and DL3DV [45] dataset.

contrast, DL3DV is dominated by *Pan* and *Truck*. This is because DL3DV is designed for 3D reconstruction, which typically involves moving the camera around a central object.

To address this issue, we sample the data to maintain a balanced distribution across different instructions. Additionally, we noticed that combined camera motions are much more frequent than single camera motions. Therefore, we resampled the single-motion data to train our model using a curriculum learning strategy. Specifically, we first train the model to learn single camera motions before introducing combined motions.

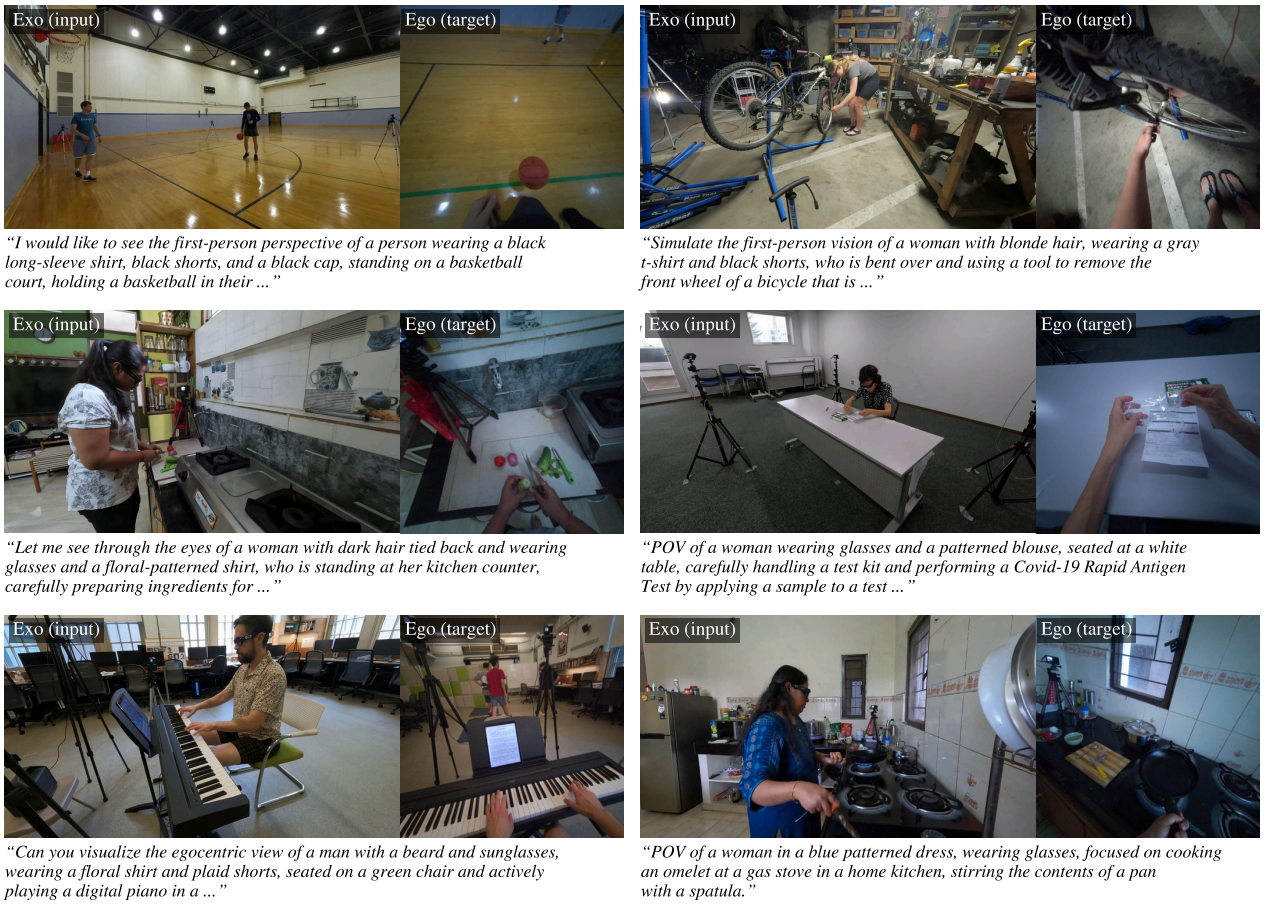


Figure 22 Exo-to-ego training pairs. Each pair shows the exo input and the synchronized ego target from Ego-Exo4D [27]; the instruction below each pair is truncated.

D.3 Exo-to-Ego Generation Data

Exo-to-ego generation is a form of multi-view generation that requires perspective taking: the target viewpoint is specified by another observer in the scene rather than by a camera-motion instruction, and the model must imagine what that observer sees. We build exo-to-ego pairs from Ego-Exo4D [27], which records each take with static third-person (exo) cameras and a head-mounted first-person (ego) camera. Each pair consists of an exo frame and the ego frame captured at the same time step. A VLM describes the camera wearer in the exo frame, including appearance, action, and surroundings, and turns the description into an instruction that asks for that person's first-person view. We remove pairs whose instruction contains unparsed VLM output, leaked templates, or references to markers that are not drawn on the image. The resulting set contains 171,558 pairs covering sports, bike repair, cooking, health care, music, and dance. The ego target is resized to 512×512. Figure 22 shows examples.

The task requires the model to locate the described person in the exo view, infer where that person is looking, and render the scene from the new viewpoint. The ego target therefore contains the objects in front of the person, often at a large viewpoint change from the exo camera, rather than the person themselves.

E Benchmark

E.1 Multi-view and Exo-to-Ego Generation

Multi-view generation. The benchmark contains 1,500 pairs, 500 each from RealEstate10K [91], DL3DV [45], and ScanNet++ [81]. Each pair consists of a source view, a target view, and a camera-motion instruction

built from the relative pose of the two views with the procedure of Appendix D.2, for example "Move the camera forward 1.81 m and rotate the camera left 36 degrees". The model generates the target view from the source view and the instruction.

Exo-to-ego generation. The benchmark contains 469 pairs from Ego-Exo4D [27]. Each pair consists of an exo frame, the synchronized ego frame, and an instruction of the form "Show me the ego view of ⟨person description⟩". The model generates the 512×512 ego view.

Metrics. We compare each generated image with the ground-truth target using PSNR, SSIM, and LPIPS [84] with an AlexNet backbone. Generated images are resized to the target resolution before scoring. Generation uses 50 denoising steps with the default classifier-free guidance of BAGEL [21] (text scale 4.0, image scale 2.0).

Tables 7 and 8 compare the original BAGEL with our models, which are the same models shown in Figures 10 and 11. On multi-view generation, our model improves all three metrics on every source, with the largest gain on RealEstate10K (+1.25 dB PSNR, −0.071 LPIPS). On exo-to-ego generation, BAGEL scores much lower (7.75 dB PSNR), consistent with its tendency to depict the described person rather than that person's view (Figure 11). Our model raises PSNR to 12.41 dB and lowers LPIPS from 0.839 to 0.670. These pixel and perceptual metrics reward matching the exact target view, and absolute values remain low because a single source image and instruction do not fully determine the target.

Table 7 Multi-view generation benchmark. PSNR (dB), SSIM, and LPIPS between generated and ground-truth target views; the number of pairs is given in parentheses.

	RealEstate10K (500)			DL3DV (500)			ScanNet++ (500)			All (1,500)		
Model	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
BAGEL	13.68	0.411	0.463	12.05	0.317	0.580	13.75	0.700	0.518	13.16	0.476	0.520
Ours	**14.93**	**0.452**	**0.392**	**12.41**	**0.321**	**0.543**	**14.22**	**0.712**	**0.497**	**13.85**	**0.495**	**0.477**

Table 8 Exo-to-ego generation benchmark. PSNR (dB), SSIM, and LPIPS between generated and ground-truth ego views on 469 Ego-Exo4D pairs.

Model	PSNR↑	SSIM↑	LPIPS↓
BAGEL	7.75	0.312	0.839
Ours	**12.41**	**0.398**	**0.670**

E.2 3DCueBench

Section 4.1 introduces the tasks and cue-balancing design. Here we provide construction details, audit procedures, and supplementary results.

E.2.1 Data and question format

The benchmark contains 400 Depth and 400 Distance questions from ten datasets (Table 9). The two tasks are sampled, balanced, and audited independently. Each question has two answer options; objects are identified by a short model-generated description and a point in the question text. Coordinates are normalized so that the image's longer side spans 0–1000. Distance questions identify the anchor in the same format. Images supplied to models contain no added target markers; the markers in the figures are for illustration only. Figure 23 shows the question template and one question per gap interval for each task.

Table 9 3DCueBench sources. Domains follow the indoor/outdoor balancing assignment.

Domain	Datasets
Indoor	CA-1M [38], ScanNet++ [81], Hypersim [55], iBims [37], Booster [54], HAMMER [34]
Outdoor	nuScenes [15], WildDet3D [33] (in-the-wild, spans both), Waymo [59], SYNS [1]

Consider the relationship in the 3D world. Which is closer to ⟨*target*⟩?
(A) ⟨*object A*⟩ (represented by the point <point>⟨*x*⟩ ⟨*y*⟩</point>)
(B) ⟨*object B*⟩ (represented by the point <point>⟨*x*⟩ ⟨*y*⟩</point>)
Please select the correct answer from the options above.

Depth: ⟨*target*⟩ = the camera.
Distance: ⟨*target*⟩ = ⟨*anchor*⟩ (represented by the point <point>⟨*x*⟩ ⟨*y*⟩</point>).

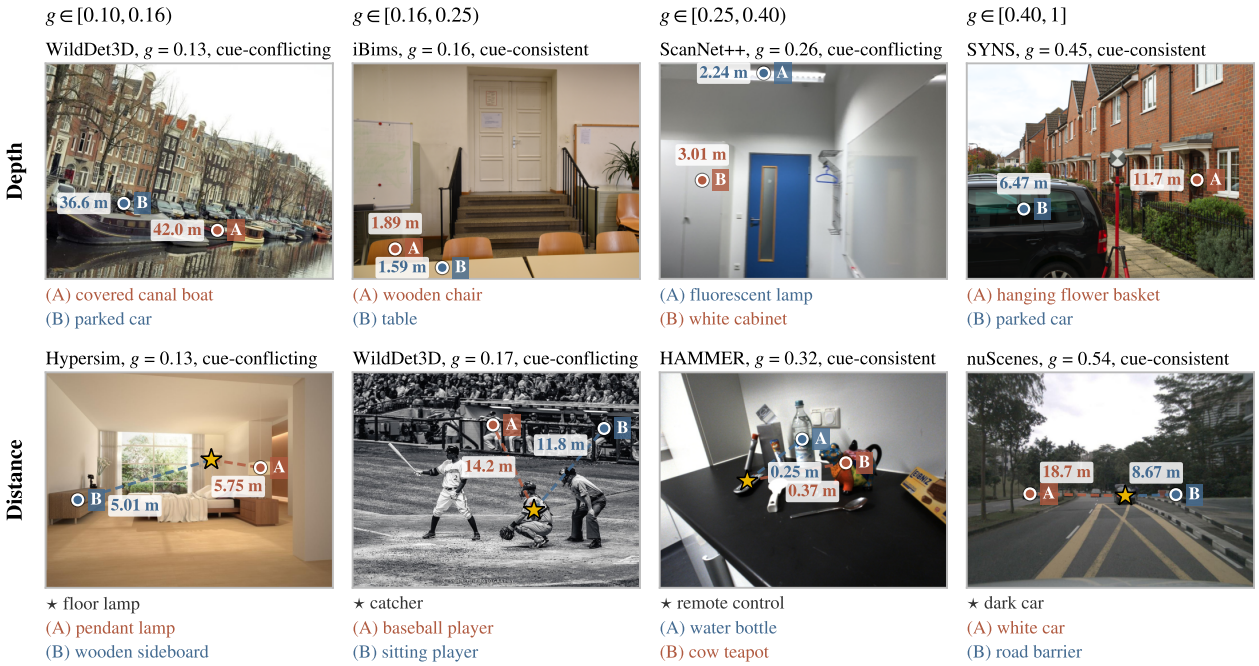


Figure 23 3DCueBench examples. Top: the question template shared by all questions; (x, y) are point coordinates normalized so that the longer image side spans 0–1000, and the image is supplied without marks. Bottom: one question per gap interval for each task. Under each image, ⋆ gives ⟨*anchor*⟩ and (A)/(B) give ⟨*object A*⟩/⟨*object B*⟩; the correct option is shown in blue. Numbers in the images are ground-truth camera depths (Depth) or 3D distances to the anchor (Distance).

E.2.2 Sampling and cue selection

Gap intervals. Using the relative gap g defined in Equation 4, we sample four intervals: $[0.10, 0.16)$, $[0.16, 0.25)$, $[0.25, 0.40)$, and $[0.40, 1]$. Each interval contains 100 Depth and 100 Distance questions, with equal indoor and outdoor representation within each task.

Ground-truth sensitivity. The lower cutoff avoids comparisons at which different depth definitions frequently disagree. We compare the 3D box-center depth with the 10th percentile and median of depth within the instance mask. In the smallest-gap CA-1M group, all three definitions agree on 44.7% of pairs, while the two surface-based definitions agree on 75.7%. These discrepancies indicate sensitivity to the depth convention; they do not by themselves establish which convention gives the correct label.

Cue signatures and pairing. We retain candidates only when the selected task-specific 2D rules agree with one another. Each retained question is therefore *cue-consistent* (all selected rules favor the answer) or *cue-conflicting* (all favor the other option). We pair the two types within the same source and gap interval, preserving the balance within each source and domain. Depth cues use vertical image-plane relationships, including box-bottom height; Distance cues use image-plane proximity to the anchor. The point-coordinate relation used in the qualitative analysis below is reported separately from the box-based cue signature.

Table 10 3DCueBench results of Seed-2.0-Lite [13] and GPT-5.5 [51]. Ordinary accuracy (%) by gap interval and cue group. All: all questions of the task; agr.: agreement with the selected 2D rule. Twelve unparseable GPT-5.5 answers are excluded.

		accuracy by interval of g					cue group		rule
Model	Task	$[0.10, 0.16)$	$[0.16, 0.25)$	$[0.25, 0.40)$	$[0.40, 1]$	all	Consistent	Conflicting	agr.
Seed-2.0-Lite	Depth	75.0	88.0	94.0	95.0	88.0	92.5	83.5	54.5
Seed-2.0-Lite	Distance	56.0	54.0	69.0	79.0	**64.5**	92.0	**37.0**	**77.5**
GPT-5.5	Depth	62.6	70.4	87.9	90.9	78.0	86.2	69.8	58.2
GPT-5.5	Distance	62.0	59.2	66.3	71.1	**64.6**	93.4	**35.7**	**78.8**

E.2.3 CV-Bench-3D audit

The rule accuracies in Figure 5(c) are computed from the bounding boxes and answer labels of 600 Depth and 600 Distance questions of CV-Bench-3D [62]. For Depth, the rule selects the object with the larger box-bottom y coordinate; for Distance, it selects the candidate whose box center has the smaller Euclidean image-plane distance to the anchor's box center. These are deterministic rule predictions, not VLM outputs.

For the difficulty audit, we match the question boxes to Omni3D test annotations [10] from Hypersim [55], SUN RGB-D [58], and nuScenes [15] by minimizing the sum of absolute differences between box coordinates within each source image. Camera-space box centers provide the depths and anchor distances used to compute g. None of the 1,200 questions has $g < 0.16$; median gaps are 0.49 for Depth and 0.76 for Distance, and 75% and 98.8%, respectively, fall in $g \geq 0.40$.

E.2.4 Blind-text audit

To test for residual cues in the question text, we evaluate Qwen2.5-VL-7B [5] without image inputs and Qwen2.5-7B-Instruct [53] on both option orders. Their accuracies range from 48.9% to 50.4%. On Distance, their predictions agree with the coordinate-proximity rule on 99.6% and 100.0% of examples, respectively. Gradient-boosted trees over numeric question fields achieve 56.1% on Depth and 53.5% on Distance under scene-level cross-validation. We keep scenes disjoint across the probe's training and evaluation folds: splitting by image instead inflates accuracy by 5.5 percentage points. These probe results measure residual text-only predictability rather than an upper bound on all possible shortcuts.

E.2.5 Results and failure analysis

We report supplementary results for Seed-2.0-Lite [13] and GPT-5.5 [51] in Table 10. These use ordinary accuracy, rather than the question-level circular accuracy in Table 1. Depth accuracy decreases as the relative gap shrinks, reaching 75.0% and 62.6% in the hardest interval. Distance accuracy is lower overall and varies less consistently with the gap.

GPT-5.5 uses the `gpt-5.5-2026-04-24` version. Excluding its 12 unparseable answers can change the evaluated interval sizes from the balanced construction.

Performance by cue group. On Distance, both models reach at least 92% on cue-consistent questions but score 37.0% and 35.7% on cue-conflicting questions (Figure 24), falling to 22% and 30% in the hardest interval. Their predictions agree with the selected rule on 77.5% and 78.8% of questions. This asymmetry is consistent with substantial reliance on the 2D cues, which aggregate accuracies of 64.5% and 64.6% obscure. Depth shows lower model-to-rule agreement, at 54.5% and 58.2%.

What the failures look like. Figure 25 shows Depth questions in which the higher target point is nearer to the camera. Grouping questions by point height, rather than the box-based signature in Table 10, gives accuracies of 82.4% versus 93.5% for Seed-2.0-Lite and 69.0% versus 86.9% for GPT-5.5 when the higher point is nearer versus farther. This separate analysis also reveals sensitivity to vertical image-plane cues.

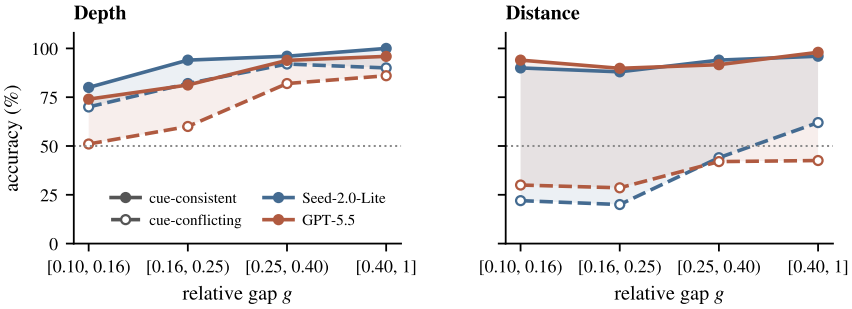
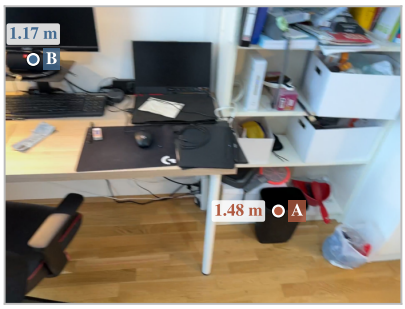


Figure 24 Sensitivity to 2D cues. Ordinary (non-circular) accuracy of Seed-2.0-Lite and GPT-5.5 on cue-consistent (solid) and cue-conflicting (dashed) questions in each gap interval. Shading spans the difference for each model; the dotted line marks chance.



Which is closer to the camera?
(A) tea cup
(B) stacked plastic containers
Answer: A
Seed-2.0-Lite: B ✗ GPT-5.5: B ✗



Which is closer to the camera?
(A) trash bin
(B) speaker
Answer: B
Seed-2.0-Lite: A ✗ GPT-5.5: A ✗



Which is closer to the camera?
(A) person
(B) suitcase
Answer: A
Seed-2.0-Lite: B ✗ GPT-5.5: B ✗

Figure 25 Counterexamples to vertical-position cues. In each Depth question the correct option (blue) is the higher point in the image yet nearer to the camera, and both models choose the other, farther option. Under each image, a red cross after a model's answer marks it as incorrect. Numbers in the images are ground-truth camera depths; marks and model answers are shown for the reader only.

E.2.6 Difficulty analysis

Tables 11 and 12 separate the 3DCueBench circular results of Tables 1 and 2, respectively, by the benchmark's ground-truth relative-gap intervals; smaller gaps indicate more difficult comparisons.

Comparison across models. In Table 11, every open-source VLM loses accuracy as the gap shrinks. The strongest, VST-7B-SFT, drops from 88.0% in the easiest interval to 61.0% in the hardest, whereas GeoUG drops from 97.0% to 84.0%. GeoUG's margin over VST-7B-SFT therefore grows from 9.0 points in the easiest interval to 23.0 points in the hardest. DA3 answering from its own depth follows a similar trend to GeoUG: it is 1.5 and 1.0 points higher in the two hardest intervals, 2.5 points lower in the third, and 0.5 points higher in the easiest.

Draft-conditioned policy. In Table 12, the two option-order variants of a question share one generated depth map. Always generating improves over direct answering by 23.0 points in the two hardest intervals and by 12.5 and 5.0 points in the easier ones. The initial policy falls furthest behind always generating in the two hardest intervals (9.5 and 7.0 points). The $\lambda = 0.1$ policy stays within 1.0 point of always generating in every interval, and the random gate at the same per-interval rate is 2.0–4.6 points below the policy.

Table 11 3DCueBench by difficulty: models in Table 1. Circular accuracy (%) in each relative-gap interval (smaller is harder), pooled over Depth and Distance (200 questions per interval).

Model / configuration	Relative gap g				Overall
	$[0.10, 0.16)$	$[0.16, 0.25)$	$[0.25, 0.40)$	$[0.40, 1]$	
BAGEL-7B-MoT [21]	52.0	57.0	61.5	76.0	61.6
Qwen2.5-VL-7B [5]	42.0	45.0	51.0	56.5	48.6
Qwen3-VL-8B [4]	54.0	61.5	71.5	80.0	66.8
InternVL3.5-8B [72]	48.0	52.5	60.0	70.0	57.6
SenseNova-SI-1.4-8B [16]	60.5	67.0	79.0	82.0	72.1
SenseNova-SI-1.5-8B [16]	57.5	67.0	78.0	83.5	71.5
VST-7B-SFT [79]	61.0	77.0	85.5	88.0	77.9
DA3 [43] (depth readout)	85.5	89.0	91.5	97.5	90.9
GeoUG (Ours)	84.0	88.0	94.0	97.0	90.8

Table 12 3DCueBench by difficulty: configurations in Table 2. Circular accuracy (%), pooled as in Table 11. The random gate uses the $\lambda = 0.1$ policy's rate within each interval.

Model / configuration	Relative gap g				Overall
	$[0.10, 0.16)$	$[0.16, 0.25)$	$[0.25, 0.40)$	$[0.40, 1]$	
Direct	61.5	66.0	81.0	93.0	75.4
Always-generate	84.5	89.0	93.5	98.0	91.3
Initial policy (SFT)	75.0	82.0	90.0	94.5	85.4
Random gate	80.8	84.2	89.4	95.0	86.1
RL, $\lambda = 0.05$	83.5	87.0	94.5	97.5	90.6
Ours ($\lambda = 0.1$)	84.0	88.0	94.0	97.0	90.8

F Additional Experimental Results

F.1 Generation Pretraining Followed by Understanding SFT

Generation pretraining alone does not reliably provide a better initialization for downstream understanding SFT. Together with the task-dependent gains from co-training, these results suggest that jointly optimizing generation and understanding may be important for transferring geometric knowledge.

We compare three initializations: the original BAGEL [21] model, a model pretrained for 6,000 steps on depth generation, and a model pretrained for 6,000 steps on multi-view generation. All three then undergo the same understanding-only SFT, using the same data and hyperparameters, with no generation objective during this stage. At steps 1,000–6,000, cumulative SFT CE tokens are 6.29M, 12.48M, 18.77M, 25.00M, 31.10M, and 37.34M for each arm. The preceding generation stage is excluded from the plotted token budget; total training compute is therefore not matched.

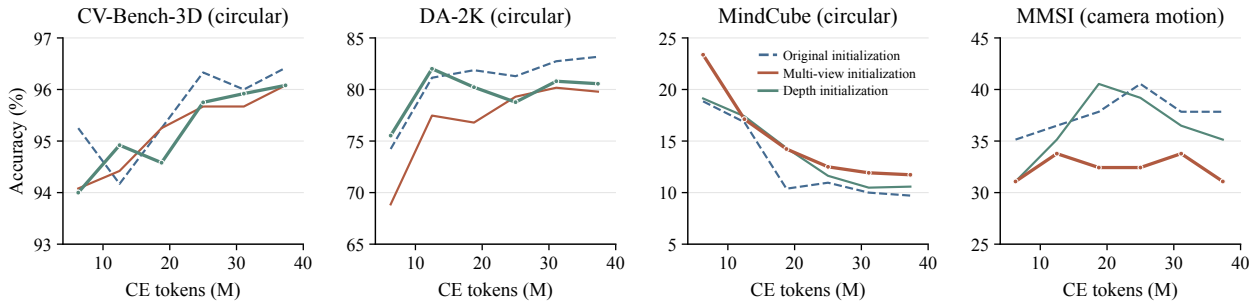


Figure 26 Generation pretraining followed by understanding SFT. Six EMA checkpoints per run, plotted against SFT CE tokens. CV-Bench-3D, DA-2K, and MindCube use circular accuracy; MMSI camera motion uses ordinary accuracy ($n = 74$).

At the final SFT checkpoint, both generation-initialized models are 0.34 percentage points below the original initialization on CV-Bench-3D [62]. Depth and multi-view initialization are lower by 2.61 and 3.38 points on DA-2K [77], and by 2.70 and 6.76 points on MMSI [80] camera motion, respectively. MindCube [70] shows a different pattern: the generation-initialized models retain small advantages of 0.87 and 2.02 points, but accuracy declines after the first evaluated SFT checkpoint for all three arms. These results concern transfer under the chosen SFT procedure, rather than the quality of the learned representations in isolation. Since the sequential and co-training experiments are not fully matched, they do not establish that co-training is necessary.

F.2 Visual and Textual Referring Formats

We report all seven evaluated open-source models under both visual and textual referring formats. CV-Bench-3D [62] uses overlaid boxes in its visual version and textual box coordinates in its textual version; both contain the same 1,200 questions (600 Depth and 600 Distance). DA-2K [77] uses colored point markers in the visual version and textual point coordinates in the textual version, with 2,068 questions. Textual coordinates follow each model's input convention, including conversion to resized-image pixels where required.

Sensitivity varies across models and tasks. Textual referring does not uniformly increase or decrease accuracy. On CV-Bench-3D, five models score lower with textual coordinates, while both SenseNova configurations score higher. Referring format therefore affects measured performance and should be held fixed when comparing systems.

DA-2K also probes marker and answer consistency. Qwen2.5-VL's 9.8% visual circular accuracy is accompanied by a strong preference for the green point (93.5% of responses), while its mean accuracy over visual variants is 53.3%. Its large Textual-minus-Visual difference therefore cannot be interpreted as an equally large gain in geometric understanding. DA-2K changes both the referring format and the circular transformation;

Table 13 Visual versus textual referring. Circular accuracy (%); both variants must be correct. Δ: Textual minus Visual (percentage points). CV-Bench-3D swaps options; DA-2K swaps colors for Visual and options for Textual.

	CV-Bench-3D			DA-2K		
Model	Visual	Textual	Δ	Visual	Textual	Δ
BAGEL-7B-MoT [21]	81.9	70.4	−11.5	58.4	55.1	−3.3
Qwen2.5-VL-7B [5]	79.9	70.8	−9.1	9.8	45.3	+35.5
Qwen3-VL-8B [4]	91.6	87.1	−4.5	67.8	44.6	−23.2
InternVL3.5-8B [72]	84.7	78.9	−5.8	53.7	31.6	−22.1
SenseNova-SI-1.4-8B [16]	87.3	89.3	+2.0	61.6	63.9	+2.3
SenseNova-SI-1.5-8B [16]	85.8	87.4	+1.6	65.5	61.1	−4.4
VST-7B-SFT [79]	94.9	93.2	−1.8	75.9	63.5	−12.4

this comparison measures sensitivity to that combined evaluation setup rather than a controlled effect of textual coordinates alone.

Result provenance. Textual results use the corrected, model-specific coordinate conventions of the September 2026 baseline evaluation. BAGEL's Visual CV-Bench-3D and DA-2K results are reused from the August 2026 runs. The table reproduces the reported aggregate scores; displayed differences are computed before rounding to one decimal place. Cross-run differences are descriptive comparisons, since equivalence of preprocessing and evaluator versions has not been established.

F.3 External Depth Tools

We also evaluate whether supplying an existing VLM with external depth measurements yields comparable performance. DA3 [43] (DA3NESTED-GIANT-LARGE-1.1) predicts depth, a shared measurement procedure extracts object depths and pairwise 3D distances, and the measurements are supplied as text to the answering VLM. Table 14 reports the resulting system performance alongside the corresponding direct-answer baselines. Qwen2.5-VL uses different input resolutions with and without DA3, and BAGEL prompts also differ; these are reference system comparisons rather than controlled estimates of the tool effect.

Table 14 External depth tools. Circular accuracy (%); both option orders must be correct. All benchmarks use textual referring formats. The last row answers from the DA3 measurements alone, without a VLM, as in Table 1.

Answering model	Depth tool	CV-Bench-3D	3DCueBench	DA-2K
BAGEL-7B-MoT [21]	None	70.4	61.6	55.1
BAGEL-7B-MoT [21]	DA3 [43]	83.0	67.8	54.5
Qwen2.5-VL-7B [5]	None	70.8	48.6	45.3
Qwen2.5-VL-7B [5]	DA3 [43]	79.3	54.4	48.4
None (depth readout)	DA3 [43]	95.3	90.9	92.1

The tool-assisted systems score higher on CV-Bench-3D and 3DCueBench than their corresponding direct-answer baselines, but gains do not hold consistently on DA-2K. Their scores remain below GeoUG and below answering from the DA3 measurements alone (last row) on all three benchmarks. This comparison shows that the evaluated combination of external measurements and an existing VLM is insufficient to match GeoUG; it does not establish that self-generated depth is intrinsically more accurate than DA3 or isolate the effect of tool use from prompting, resolution, and training differences.