Imaginative Perception Tokens Enhance Spatial Reasoning in Multimodal Language Models

Mahtab Bigverdi[1,2*], Linjie Li[1*], Weikai Huang[1,2*], Yiming Liu[1], Jaemin Cho[1,2,5], Tuhin Kundu[3], Chris Dongjoo Kim[2], Zelun Luo[4], Jieyu Zhang[1,2], Linda Shapiro[1], and Ranjay Krishna[1]

[1] University of Washington, Seattle, WA, USA
[2] Allen Institute for AI, Seattle, WA, USA
[3] Microsoft, Seattle, WA, USA
[4] OpenAI, San Francisco, CA, USA
[5] John Hopkins University, Baltimore, Maryland, USA
{mahtab,linjli,weikaih}@cs.washington.edu

Abstract. Vision-language models (VLMs) excel at many tasks, yet continue to struggle with spatial reasoning, problems where the key information is not directly observable in the input. Many spatial questions require *imaginative perception*: simulating an unseen viewpoint, tracing a trajectory through an occluded space, or integrating partial views into a coherent spatial map. Humans naturally support this kind of reasoning through imagination. Prior work has introduced intermediate visual representations (e.g., visual thoughts, depth, or box tokens), but these intermediates often refine structure already visible rather than predicting the missing spatial structure implied by the evidence. We introduce **Imaginative Perception Tokens (IPT)**, intermediate perceptual representations that externalize what a VLM would perceive under an alternative spatial configuration while remaining consistent with the observed input. To study this capability, we formulate three tasks that require imaginative perception: **Perspective Taking (PET)**, **Path Tracing (PT)**, and **Multiview Counting (MVC)**. For each task, we construct datasets of ∼20K examples spanning simulated and real-world settings, paired with ground-truth intermediate imaginations, final answers, and curated evaluation benchmarks. Using the unified VLM BAGEL [12] as our backbone, IPT supervision improves spatial reasoning across several settings and often outperforms textual chain-of-thought training, even when no image is generated at inference time. For example, on MVC, IPT improves accuracy by 3.4% and achieves performance competitive with strong closed-source models on Path Tracing. We also find that mixed training with IPT and label-only data can further improve performance. In contrast, textual chain-of-thought can be detrimental on these tasks, substantially degrading performance in some cases, highlighting a modality mismatch when forcing spatial computation through language. Overall, IPT provides a principled supervision signal for reasoning over unobserved structure, yielding stronger spatial generalization and a more interpretable intermediate aligned with the underlying geometry of the task. Code will be released at the project page.

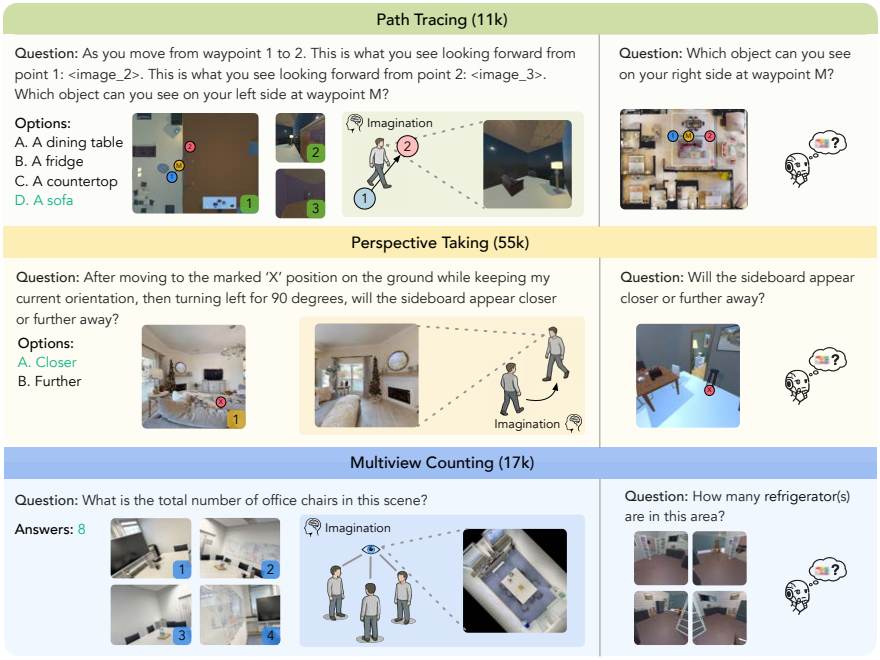

Fig. 1: Overview of the three spatial imagination tasks. The left columns show training examples with ground-truth imaginative perception; the right columns show evaluation examples.

Keywords: Vision Language Models · Perception Tokens · Visual Thought · Imagination

1 Introduction

Spatial reasoning remains a persistent challenge for vision-language models (VLMs) [1, 8,10]. These models struggle particularly with tasks that require reasoning about 3D spatial relationships, how objects relate within three-dimensional environments, how these relationships transform under viewpoint changes, or how to integrate information from multiple partial observations into a coherent scene representation [19,47]. While current VLMs can recognize objects and attributes effectively, they frequently fail when spatial reasoning demands manipulating structure, such as predicting scene appearance from novel viewpoints [23, 25] or aggregating information across multiple views [43].

A key reason for this difficulty is that many spatial reasoning problems cannot be solved by analyzing the input alone. Instead, they require constructing a spatial representation that is not directly observed. Humans naturally address

such problems through imagination: when asked what lies to the left after moving to a new position, or how many objects exist in a room seen from several viewpoints, we mentally simulate the scene from unseen perspectives or integrate partial observations into a unified spatial map [39, 47, 48]. In other words, spatial reasoning often depends on imagining missing spatial structure that proceed despite incomplete observations.

Existing approaches provide only partial solutions. Recent work teaches models to generate intermediate visual thoughts alongside language [15,17,22], while others introduce structured perceptual intermediates, such as depth maps or bounding boxes represented as tokens [2,29,44]. Although these methods demonstrate that intermediate visual representations can support reasoning, they primarily operate over information already present in the input observation, refining visible structures or extracting perceptual attributes. However, as discussed above, many spatial reasoning problems arise precisely because the required spatial information is not directly observable, and therefore requires imagination.

To address this gap, we propose **Imaginative Perception Tokens** for VLMs. When VLMs are trained with them, they enable intermediate reasoning steps that represent novel spatial views. Unlike standard perceptual intermediates that describe structures visible in the input, imaginative representations correspond to what the model would perceive if it were observing the input from a different spatial configuration, such as from an unseen viewpoint or after integrating multiple partial observations into one. At the same time, they are not unconstrained imagination: the predicted percept must remain consistent with the observed scene. These tokens externalize the model's prediction of what would be perceived given incomplete spatial evidence.

To study this capability, we propose three spatial reasoning tasks that fundamentally require imaginative perception. (1) Perspective Taking requires predicting how a scene would appear from a new viewpoint given a single first-person observation ("If you move to the marked position and turn left, will the chair appear on your left or right?"); (2) Path Tracing requires inferring what an agent would see along a navigation path based on a top-down view ("If you walk along the marked path, which object will you see on your side?"); Finally, (3) Multi-view Counting requires integrating multiple partial observations into a top-down view to determine the number of objects present in the scene. These tasks would be made easy when correctly predicting what would be perceived in a different spatial configuration. For each task we construct a dataset of approximately 20k examples each drawn from both real-world and synthetic simulated environments, with ground-truth intermediate spatial imaginations paired with final answers. Each dataset is accompanied by a human-filtered benchmark for evaluation. Together these constitute the first datasets designed explicitly to train and evaluate visually-grounded intermediate spatial reasoning in models.

Empirically, we find that training with imaginative perception supervision can improve performance on these spatial reasoning tasks compared to answer-only supervision, and often compares favorably to textual chain-of-thought approaches. These improvements can persist even when the model does not ex-

plicitly generate intermediate images at inference time, suggesting that such supervision may help models develop stronger internal spatial representations. At the same time, we observe that the benefits vary across tasks and settings, indicating that imagination quality and task structure both play important roles.

Overall, our results suggest that supervising models with intermediate perceptual predictions offers a useful direction for improving spatial reasoning, particularly in settings where the required structure is not directly observable from the input.

2 Related Works

Evaluation of VLMs' spatial reasoning. A growing body of benchmarks has established that spatial reasoning remains a persistent weakness of modern vision-language models. Early datasets target brittleness in basic spatial predicates: SpatialSense [40] reduces language priors through adversarial crowdsourcing, while VSR [24] scales relation types in a caption-verification format, and What'sUp [19] uses minimal-pair testing to reveal systematic failures on left-/right and above/below distinctions. More recent work shifts from 2D relations to viewpoint and 3D structure. 3DSRBench [25] shows that models fail under modest changes in perspective, depth, and occlusion, and ViewSpatial-Bench [23] identifies a "perspective gap": models often succeed in camera-centered views but break when asked to adopt human-centered viewpoints.

Benchmarks have also expanded to multi-image and video settings where maintaining a consistent spatial state is essential. VSI-Bench [39] tests whether models can build a persistent mental map from videos, while MMSI-Bench [43] reports large human–model gaps on cross-view scene reconstruction. MindCube [47] is closely aligned with our motivation, targeting spatial mental modeling from limited views, including perspective-taking and "what-if" scene dynamics. Counting Stacked Objects [13] studies 3D object counting under heavy occlusion across multiple views, directly analogous to our Multiview Counting setting. Finally, benchmark design work emphasizes that shortcuts remain pervasive: Brown *et al.* [3] construct VSI-Bench-Debiased by iteratively pruning samples solvable via priors, reinforcing the need for evaluations where success requires genuine spatial computation.

Collectively, these benchmarks diagnose *where* VLMs fail spatially, but they typically evaluate *discriminative* understanding—reading off a relation from an observed view—rather than *constructive* spatial imagination. Our work is complementary: we isolate *imaginative perception* as a standardized intermediate substrate, and pair each task with a ground-truth intermediate spatial imagination rather than only a final answer label.

Intermediate representations for spatial reasoning. Chain-of-thought prompting [35] can improve multi-step reasoning, but serializing viewpoint transformations, occlusions, and geometric constraints into language is often awkward and error-prone. This motivates intermediate representations in modalities better aligned with spatial computation. One direction externalizes reasoning into ex-

plicit visual buffers: Visual Sketchpad [17] equips models with drawing actions for iterative refinement, and MVoT [22] introduces visualization-of-thought traces that help on dynamic spatial tasks where text CoT struggles. ThinkMorph [15] studies interleaved text–image reasoning traces, and OpenAI describes o3/o4-mini as using chains-of-thought that include simple image transformations during reasoning [27]. A complementary line introduces latent visual scratchpads: Mirage [44] frames latent tokens as "machine mental imagery," and Mull-Tokens [29] generalizes to modality-agnostic latent thinking tokens.

Our work differs in what the intermediate is meant to represent. Many prior approaches treat intermediate images or latents as optional visualizations of *visible* structure. We instead target *imaginative perception*: predicting what would be perceived under an unobserved spatial configuration (e.g., a rotated viewpoint or a top-down path state), a representation constrained by the input but not present in it. This framing provides a principled criterion for when intermediate visual thoughts are necessary and a controlled way to supervise them.

Unified multimodal models for interleaved understanding and generation. Producing imaginative perceptual intermediates within a single model requires the ability to both understand and generate images. Unified decoder-only architectures treat image tokens as first-class sequence elements, enabling arbitrary text–image interleaving. Chameleon [5] is an early example, while Show-o2 [38] and Janus [7] offer alternative unified designs that balance understanding and generation. We build on BAGEL [12], a unified model pretrained on large interleaved corpora that exhibits strong spatial capabilities, making it a natural substrate for producing intermediate spatial imaginations. Crucially, however, a unified architecture alone does not guarantee that intermediate images are *used* in a way that supports reasoning; our work provides task constructions and supervision that make imaginative perception the relevant computational substrate.

3 Spatial Imagination: Tasks and Datasets

We introduce three spatial reasoning tasks that require constructing a missing spatial representation from incomplete inputs (single-view, partial-view, or map inputs). For each task, we build a 10k–50k training set with paired *ground-truth spatial imaginations* (task-specific intermediate visual supervision) and final answers, and we release a human-filtered benchmark for controlled evaluation. All datasets will be released publicly; for consistency across tasks, we train our models on the AI2-THOR [21] subset of each training set. Table 1 and Fig 1 summarize the training data and evaluation benchmarks.

3.1 Perspective Taking

Given a first-person view of an indoor scene with target positions marked, the model must answer a spatial question (*e.g.*, "After moving to 'X' and turning left 90°, will the {object} be on your left or right?") about the scene from the

Table 1: Dataset and benchmark statistics. All experiments use the AI2-THOR subset for training; additional data sources are released for future research. †: human-verified subset.

Task		
Perspective Taking	Source (samples)	AI2-THOR (20,531) + Habitat (19,998) + Real images (15,000)
	Train	55,529
	Eval	AI2-THOR† (238), Habitat† (300)
	IPT Format	Novel-viewpoint image
Path Tracing	Source (samples)	AI2-THOR (11,204)
	Train	11,204
	Eval	AI2-THOR† (329), Real† (332)
	IPT Format	Sideview image
Multiview Counting	Source (samples)	AI2-THOR (17,079) + MessyTable (1,880) + ScanNet (540)
	Train	19,499
	Eval	AI2-THOR† (260)
	IPT Format	Top-down BEV map

new viewpoint. Since the target view is never provided, the model must mentally simulate the spatial transformation rather than read off the answer directly.

Sub-categories. Questions span two spatial relation types across six balanced sub-categories. *Distance change* asks whether a target object becomes closer or further after the viewpoint shift: (1) *closer* and (2) *further*. *Relative position* asks whether the object falls to the left or right in the new view, defined by the object's lateral position before and after the transformation: (3) *left→left*; (4) *left→right*; (5) *right→left*; (6) *right→right*. Overall accuracy is the unweighted mean across all six sub-categories so that each spatial relationship contributes equally, preventing models from gaming the metric by over-predicting common cases.

Imaginative perception target. A novel-viewpoint rendering of the scene from the target position, directly supervised against ground-truth renders from the 3D scene.

Data. Synthetic data is generated from AI2-THOR [21] and Habitat [26, 28, 31] by sampling source/target camera pairs, rendering first-person views, and annotating the source view with a red "X" marking the target. Questions cover two relation types (distance change, relative position) across six balanced sub-categories. A *mixed* training data variant additionally incorporates real-world examples from the Visual Spatial Tuning dataset [42] (camera motion subset) as a synthetic-to-real bridge. The base training set contains 20,531 AI2-THOR examples; the mixed variant totals 55,529. We evaluate on held-out human-verified AI2-THOR (238) and Habitat (300) benchmarks. Full sub-category breakdowns and data generation details are provided in the Appendix.

3.2 Path Tracing

Given a top-down map with a marked path $1 \to 2$, a midpoint M_1, and egocentric forward views at waypoints 1 and 2, the model must identify which object is visible on a queried side at M_1. Neither the top-down map nor the endpoint views

reveal first-person visibility at the midpoint, requiring the model to imagine what the agent would see from ground level.

We evaluate under three input settings of increasing spatial cues: *Path* (map only), *PathArr* (map + query direction arrow), and *EgoDir* (map + egocentric endpoint views).

Imaginative perception target. A sideview image — a first-person rendering from M_1 — that externalizes the 3D visibility reasoning the top-down input cannot support. Ground-truth sideviews are rendered directly from the simulator at M_1.

Data. Synthetic data is generated from AI2-THOR [21] and ProcTHOR [11], sampling feasible two-waypoint paths balanced across room types and distance bins. Questions are template-generated with four answer choices and quality-filtered via TIFA-style verification [16] using GPT-4.1 majority voting; samples answerable from endpoint views alone are removed to ensure genuine imagination is required. The synthetic training set contains 11,204 examples. A real-world benchmark of 332 human-verified questions is constructed from Matterport3D [6] top-down views and evaluated on Path and PathArr settings only. Full filtering criteria and real-world annotation pipeline details are in the Appendix.

3.3 Multiview Counting

Given several first-person frames of the same environment, the model must select the correct count of a queried object (*e.g.*, "How many chairs are in this area?"). Since no single view reveals the full layout, and the same object often appears across multiple frames, the model must construct a unified spatial representation that resolves both occlusions and cross-view duplicates.

Imaginative perception target. A top-down bird's-eye view (BEV) map aggregating all input views, making de-duplication explicit by mapping each object to a single spatial location. Ground-truth BEV maps are rendered from an overhead camera in the 3D scene.

Data. Synthetic examples are generated via multi-camera and rotation trajectory types. Real-world data is sourced from MessyTable [4] (fixed multi-camera rig; overhead image as ground-truth BEV) and ScanNet++ [46] (point-cloud BEV maps converted to photorealistic overhead images via Qwen Edit [36]). Questions are four-choice MCQ with distractors sampled near the true count. The base training set contains 17,079 synthetic examples; the mixed variant totals 19,499. We evaluate on a human-verified benchmark of 260 samples. Details on trajectory types, BEV rendering, and distractor sampling are in the Appendix.

4 Method: Imaginative Perception Tokens

The core of our approach is to enable Multimodal Language Models (MLLMs) to externalize spatial reasoning through **Imaginative Perception Tokens**. Unlike standard textual chain-of-thought or methods outsourcing visual imagination

with an external visual generation model, our method requires the model to generate a visual representation of a *non-observed* spatial configuration—such as an unseen viewpoint or an integrated top-down map—as a functional prerequisite for answering a spatial query.

4.1 Problem Formalization

Given an input context $\mathcal{C}$ consisting of one or more observed images $\mathcal{I}_{obs} = \{I_1, \ldots, I_k\}$ and a spatial language query Q, the goal is to predict the correct answer A. We decompose this into a two-stage generative process. First, the model generates **imaginative perception tokens** $\hat{I}_{imag}$, representing the implied spatial structure requested by the task (e.g., the view from a new coordinate): $P(\hat{I}_{imag}|\mathcal{I}_{obs}, Q)$ Second, conditioned on this imaginative perception tokens $\hat{I}_{imag}$, the model produces the final answer: $P(A|\mathcal{I}_{obs}, Q, \hat{I}_{imag})$.

4.2 Architecture

We implement this approach using BAGEL [12], a unified decoder-only transformer that natively supports interleaved multimodal understanding and generation. BAGEL employs a Mixture-of-Transformer-Experts (MoT) design: the model utilizes two transformer experts, one optimized for multimodal understanding and another for generation. Both operate on the same token sequence through shared self-attention at every layer. Images are represented via two distinct paths. *Understanding tokens* (U) are extracted via a SigLIP2 [32] ViT encoder to capture semantic content, while *Generation tokens* (G) are latent representations from a FLUX VAE used for high-fidelity synthesis. Because all tokens (text, U, and G) coexist in a single shared context window, the model maintains lossless interaction between understanding and generation modules.

While BAGEL's standard generation tokens are typically used for open-ended text-to-image generation or editing, we repurpose this generative capacity for **spatial reasoning**. In our framework, the generation target is not a stylistic output but a precise **view imagination**—a visually grounded intermediate that represents the unobserved 3D structure of the scene.

4.3 Training and Inference

Training Objective. We optimize the framework using a multi-task loss $\mathcal{L}_{total} = \lambda_{fm}\mathcal{L}_{fm} + \lambda_{lm}\mathcal{L}_{lm}$. The model is trained to jointly produce the imaginative perception and the final answer:

1. **Flow-Matching Loss ($\mathcal{L}_{fm}$):** For the imaginative intermediate, BAGEL adopts the **Rectified Flow** method. The model learns to predict the velocity field v_t required to transform Gaussian noise into the target latent G_{gt} representing the unobserved view, conditioned on the preceding context $\mathcal{C}$:

$$\mathcal{L}_{fm} = \mathbb{E}_{t,G_0,\mathcal{C}}\left[\|v_t(G_t|\mathcal{C}) - (G_{gt} - G_0)\|^2\right] \quad (1)$$

2. **Language Modeling Loss ($\mathcal{L}_{lm}$):** We minimize the negative log-likelihood of the final VQA answer tokens A, conditioned on the observed context and the ground-truth imaginative tokens:

$$\mathcal{L}_{lm} = -\sum_{i=1}^{|A|} \log P(a_i|\mathcal{C}, U_{gt}, G_{gt}, a_{<i}) \tag{2}$$

Inference. At inference time, the model operates in one of two modes depending on the task and configuration. In the **text-only** mode, the model produces only a textual answer without generating any visual intermediate $A \sim P(A \mid \mathcal{C})$, serving as a baseline. In the **imagination** mode, the model first performs iterative denoising over VAE tokens to produce the imaginative latent: $\hat{G}_{imag} = \int_0^1 v_t(G_t \mid \mathcal{C})\, dt$ The decoded image $\hat{I}_{imag}$ is immediately re-encoded and appended to the context as both ViT understanding tokens and VAE generation tokens: $\mathcal{C}' = \Big[\mathcal{C}, \text{ViT}(\hat{I}_{imag}), \text{VAE}(\hat{I}_{imag})\Big]$ The model then attends to its own imagination to predict the final answer $A \sim P(A \mid \mathcal{C}')$.

5 Experiments

We evaluate *imaginative perception tokens* on the three spatial reasoning tasks introduced in Sec. 3: **Perspective Taking (PET)**, **Path Tracing (PT)**, and **Multiview Counting (MVC)**. To enable controlled comparisons, we train all task-specific models on the AI2-THOR subset of each dataset. We additionally report transfer to cross-environment benchmarks (Habitat), real-world images, and external datasets. All tasks use multiple-choice evaluation with balanced answer distributions.

PT is evaluated under three input variants that provide increasing spatial cues: **EgoDir** (egocentric direction only), **Path** (top-down path overlay), and **PathArr** (path with directional arrows) and average accuracy reported. Unless otherwise stated, we report accuracy (%) and use the same prompt formatting across baselines and our models.

5.1 Setup

Baselines. We compare against two groups of models, evaluated zero-shot with task-specific prompts. *VQA models* include GPT-5, GPT-5.2, Gemini 2.5 Flash, Gemini 3 Flash, InternVL3.5-8B, Qwen2.5-VL-7B, and Qwen3-VL-8B. *Unified models* that support both understanding and generation include Janus-Pro-7B and Chameleon 7B.

Our model variants. We fine-tune BAGEL [12] under several configurations to isolate the contribution of imagination supervision. Each fine-tuned model is task-specific and trained on a single task using AI2-THOR data only (unless noted otherwise):

- **Bagel (base)**: pretrained model with no task-specific fine-tuning.

Table 2: Main results. Accuracy (%) on AI2-THOR (in-domain) and different-environment (out-of-domain) benchmarks. PT reports the average across input settings (EgoDir/Path/PathArr for AI2-THOR; Real/Real+Arr for different environments). Text CoT generates a textual chain-of-thought before answering. IPT (Imaginative Perception Token) generates an intermediate image before answering. For our models, accuracy reports the maximum between answer-only and free-generation inference. Best per group in **bold**.

	AI2-THOR			**Different Env.**	
Model	PET	PT	MVC	PET	PT
VQA Models					
GPT-5	**79.8**	**60.2**	53.5	**69.3**	**80.9**
GPT-5.2	45.5	32.9	44.2	54.0	63.0
Gemini 2.5 Flash	51.0	41.5	30.8	66.3	71.4
Gemini 3 Flash	55.0	42.3	**56.9**	51.3	83.2
InternVL3.5-8B	51.5	35.8	44.6	47.7	47.4
Qwen2.5-VL-7B	50.7	37.3	38.8	54.3	44.8
Qwen3-VL-8B	52.0	35.9	43.8	46.7	64.1
Unified Models					
Janus-Pro-7B	51.8	33.5	33.1	44.7	35.3
Chameleon 7B	34.3	16.3	5.4	47.3	24.5
Ours (fine-tuned BAGEL)					
Bagel (base)	40.3	29.9	35.4	62.7	42.7
Bagel (label-only)	97.5	65.7	63.9	82.0	54.7
+ Text CoT	83.1	49.7	62.3	70.3	52.2
+ IPT	96.8	49.0	**67.3**	87.0	57.5
+ Mixed Training	**97.8**	**66.7**	62.3	**87.7**	**58.6**

- **Bagel (label-only)**: fine-tuned with answer supervision only, with no intermediate thought.
- + **Text CoT**: trained to generate a textual chain-of-thought describing the imagined spatial configuration before answering. Training CoTs are generated by GPT-5.1 using simulator ground-truth scene metadata.
- + **IPT**: trained to generate an intermediate image (the imaginative perception token) before answering.
- + **Mixed Training**: trained on a mixture of IPT examples (image-generation targets) and label-only examples (answer supervision only).

Training details. We fine-tune BAGEL-7B-MoT with AdamW (lr 1×10^{-5}, 2,000 warmup steps) on 8 GPUs using FSDP bf16, following BAGEL [12] and ThinkMorph [15]. For multi-image inputs, each image is resized to 512×512. Unless noted, IPTs use Latent-64 resolution.

5.2 Main results

Table 2 reports results on our benchmarks.

Spatial reasoning remains difficult for current VLM and unified models.

Among the zero-shot baselines, GPT-5 is the strongest across nearly all settings, yet still trails our best fine-tuned variants on multiple in-distribution tasks. Smaller open VLM models (InternVL3.5-8B, Qwen2.5-VL-7B, Qwen3-VL-8B) hover near chance on PET (50–52%) and struggle on PT, indicating that these tasks are not solvable through superficial cues. Unified models perform worse overall: Chameleon 7B drops to 34.3% on PET and 5.4% on MVC, suggesting that current unified designs often trade away understanding robustness in exchange for generation capability.

Answer supervision alone yields large gains and transfers across environments.

Bagel (label-only) substantially improves over Bagel (base) across all tasks, rising from 40.3% to 97.5% on AI2-THOR PET, from 29.9% to 65.7% on PT, and from 35.4% to 63.9% on MVC. These improvements transfer: label-only reaches 82.0% on Habitat PET, showing that spatial reasoning can be learned in simulation and generalized to new environments.

Imagination supervision helps most when language is a poor interface.

On MVC, IPT achieves the best accuracy (67.3%), outperforming label-only (63.9%) and Text CoT (62.3%). On different-environment PET (Habitat), IPT reaches 87.0% (vs. 82.0% for label-only), and Mixed Training improves further to 87.7%. On PT, Mixed Training achieves the best results on both synthetic (66.7%) and real (58.6%) benchmarks, outperforming label-only (65.7% / 54.7%) and all baselines. IPT also improves real-world PT transfer (57.5%) over label-only (54.7%) and Text CoT (52.2%). Notably, IPT models are evaluated in *answer-only* mode: the model does not generate an image at inference, yet the imagination targets during training strengthen internal spatial representations that transfer across environments.

Text CoT underperforms label-only and IPT.

Text CoT typically falls behind label-only (e.g., PET 83.1% vs. 97.5%, PT 49.7% vs. 65.7%) and also behind IPT (e.g., MVC 62.3% vs. 67.3%, PET 83.1% vs. 96.8%). Compared to label-only, the Text CoT objective forces the model to allocate capacity to generating long spatial descriptions during fine-tuning, which competes with answer prediction. Compared to IPT, the gap reflects a modality mismatch: viewpoint changes, occlusions, and cross-view correspondences are difficult to serialize into natural language, and the resulting textual traces introduce noise rather than useful structure. IPT represents these relationships directly in the visual modality where they are naturally expressed.

5.3 Ablations

Latent resolution controls imagination quality and downstream accuracy.

Table 3 and Fig. 2 ablate IPT resolution on PET and MVC. At Latent-4 (64×64), imaginations are blurry and lose spatial detail; at Latent-64 (1024×1024),

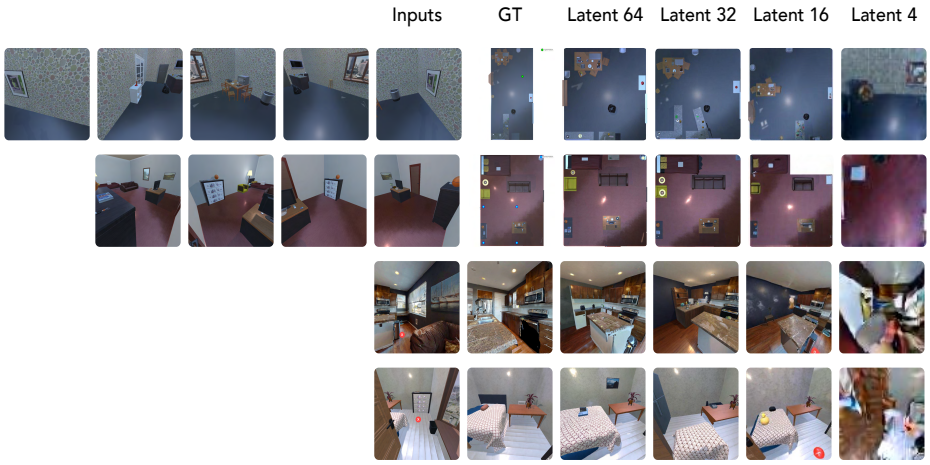

Fig. 2: Qualitative examples of model-generated imaginative perception tokens. Top two rows: MVC example showing imagined top-down BEV maps. Bottom: PET examples showing imagined novel viewpoints. From left to right, imagination resolution increases from Latent-4 (64×64) to Latent-64 (1024×1024). Higher resolution produces sharper and more spatially faithful imaginations, preserving object identities and relative positions needed for downstream reasoning.

Table 3: Ablation on latent size. Accuracy (%) with w/ Thought inference mode at different imagination resolutions. Best per column in **bold**.

		PET		MVC
Latent Size	**Resolution**	AI2-THOR	Habitat	AI2-THOR
Latent-4	64×64	87.4	73.3	53.5
Latent-16	256×256	95.3	81.0	56.2
Latent-32	512×512	95.0	**87.0**	58.9
Latent-64	1024×1024	**96.8**	83.3	**63.1**

imaginations become sharper and more spatially faithful, preserving object identities and relative positions. Quantitatively, increasing resolution from Latent-4 to Latent-64 improves AI2-THOR PET from 87.4% to 96.8% and MVC from 53.5% to 63.1%. Habitat PET peaks at Latent-32 (87.0%) and drops slightly at Latent-64 (83.3%), suggesting mild overfitting to AI2-THOR appearance statistics at the highest resolution.

Thought modality and inference mode. Table 4 ablates the training signal (Text CoT vs. IPT) and inference mode (generate thought, answer-only, or oracle GT). **IPT training builds stronger spatial representations than Text CoT.**

On PT, IPT with answer-only inference (61.1%) outperforms Text CoT with answer-only inference (55.8%) by 5.3 points. On MVC, IPT with image generation (63.1%) outperforms Text CoT with text generation (61.5%).

Table 4: Ablation on thought modality and inference mode. Accuracy (%) on AI2-THOR benchmarks (PT uses EgoDir variant). We compare Text CoT vs. IPT training and vary inference mode: generate thought then answer (w/ text/image), answer directly (answer-only), or condition on ground-truth (w/ GT image).

Training	Inference	PET	PT	MVC
Text CoT	w/ text	83.1	53.1	61.5
Text CoT	answer-only	78.3	55.8	62.3
IPT	w/ image	96.8	50.4	**67.3**
IPT	answer-only	96.8	61.1	62.3
IPT	w/ GT image	96.7	**86.7**	**67.3**

Table 5: Transfer to similar external benchmarks. Accuracy (%). SAT tests perspective taking and MessyTable tests multiview counting, both in domains unseen during training. Best in **bold**.

	PET	MVC
Model	SAT (66)	MessyTable (200)
Bagel (base)	34.9	29.0
Bagel (label-only)	59.1	32.5
+ Text CoT	50.0	30.0
+ IPT	57.6	28.5
+ Mixed Training	**63.6**	**37.0**

Imagination supervision is useful, but explicit generation is not required at inference.

For IPT models, answer-only mostly outperforms generating the imagination explicitly: on PT, answer-only reaches 61.1% vs. 50.4% with generation. For Text CoT, generating the chain-of-thought also slightly underperforms answer-only (53.1% vs. 55.8% on PT), though the gap is smaller than for IPT. This asymmetry suggests that producing faithful imaginations is harder than producing text descriptions, and imperfect generations can mislead downstream reasoning. However, training with imagination targets remains valuable: answer-only IPT matches GPT-5 on PT (61.1%).

Ground-truth imaginations reveal headroom.

When given ground-truth imaginations instead of model-generated ones, PT accuracy jumps from 50.4% to 86.7% (+36.3) and MVC rises from 63.1% to 67.3% (+4.2). The large PT gap indicates that imagination quality is the dominant bottleneck for path tracing; for PET, model-generated imaginations nearly match GT (96.8% vs. 96.7%), leaving little room for improvement.

IPT transfers to aligned external benchmarks.

Table 5 evaluates transfer to external benchmarks that test similar spatial capabilities: SAT [30] (perspective-taking subset) and MessyTable [4] (multiview counting). On SAT, Bagel (label-only) improves from 34.9% to 59.1% over Bagel

Table 6: Does our data help on other spatial tasks? Accuracy (%) on benchmarks beyond our training task categories. Fine-tuning on our AI2-THOR MVC data consistently improves over Bagel (base), suggesting that the spatial reasoning learned from our datasets transfers broadly. Best per column in **bold**.

Model	ScanNet (200)	MindCube (200)	All-Angles (170)
VQA Models			
GPT-5	58.5	**67.3**	**67.9**
GPT-5.2	48.5	37.5	29.4
Gemini 2.5 Flash	48.0	50.3	37.9
Gemini 3 Flash	62.5	56.5	64.2
InternVL3.5-8B	53.5	42.1	54.8
Qwen2.5-VL-7B	**63.5**	47.8	51.8
Qwen3-VL-8B	62.5	34.5	42.3
Unified Models			
Janus-Pro-7B	39.5	42.0	45.0
Chameleon 7B	5.5	25.4	17.7
Ours (fine-tuned BAGEL)			
Bagel (base)	40.5	39.5	40.0
Bagel (fine-tuned)	**52.0**	**47.5**	**50.0**

(base), and Mixed Training further improves to 63.6%. On MessyTable, Mixed Training reaches 37.0%, up from 29.0% for Bagel (base).

Training with our data improves performance on other spatial benchmarks.

Finally, we test whether our training data improves spatial reasoning on tasks with different structures: ScanNet [9] (in-the-wild multiview counting), MindCube [47] (abstract geometric reasoning), and All-Angles-Bench [45] (cross-view matching on EgoHumans [20]). Because IPTs are task-specific by construction (e.g., rotated views for PET, bird's-eye paths for PT), they do not directly transfer to these settings. We therefore fine-tune on AI2-THOR MVC using answer supervision only. Bagel (fine-tuned) consistently improves over Bagel (base) across all three benchmarks (40.5%→52.0% on ScanNet, 39.5%→47.5% on MindCube, 40.0%→50.0% on All-Angles), indicating that our simulator data builds broadly useful spatial representations even when the specific imaginative token target changes.

6 Conclusion

We introduced Imaginative Perception Tokens, intermediate visual representations that externalize spatial reasoning about unobserved structure in multimodal language models. We designed three tasks, Perspective Taking, Path Tracing, and Multiview Counting, that specifically require constructing missing spatial information, and built datasets with ground-truth intermediate imaginations for each. Experiments on a unified MLLM backbone show that training

with imagination supervision consistently improves spatial reasoning over both label-only and text chain-of-thought baselines, even when the imagination is not explicitly generated at inference. Ablations further reveal that imagination quality directly governs downstream accuracy, with ground-truth intermediates yielding large headroom over current generation quality.

Acknowledgements

This work was partially supported by Samsung and CoCoSys. We thank Zhe Zhou for helping clean and filter our datasets and benchmarks.

References

1. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., Zhu, K.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025), https://arxiv.org/abs/2511.21631
2. Bigverdi, M., Luo, Z., Hsieh, C.Y., Shen, E., Chen, D., Shapiro, L.G., Krishna, R.: Perception tokens enhance visual reasoning in multimodal language models. arXiv preprint arXiv:2412.03548 (2024)
3. Brown, E., Yang, J., Yang, S., Fergus, R., Xie, S.: Benchmark designers should "train on the test set" to expose exploitable non-visual shortcuts. arXiv preprint arXiv:2511.04655 (2025)
4. Cai, Z., Zhang, J., Ren, D., Yu, C., Zhao, H., Yi, S., Yeo, C.K., Change Loy, C.: Messytable: Instance association in multiple camera views. In: European Conference on Computer Vision. pp. 1–16. Springer (2020)
5. Chameleon Team: Chameleon: Mixed-modal early-fusion foundation models. arXiv preprint arXiv:2405.09818 (2024)
6. Chang, A., Dai, A., Funkhouser, T., Halber, M., Nießner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments (2017), https://arxiv.org/abs/1709.06158
7. Chen, X., Wu, Z., Liu, X., Pan, Z., Liu, W., Xie, Z., Yu, X., Ruan, C.: Janus-pro: Unified multimodal understanding and generation with data and model scaling. arXiv preprint arXiv:2501.17811 (2025)
8. Clark, C., Zhang, J., Ma, Z., Park, J.S., Salehi, M., Tripathi, R., Lee, S., Ren, Z., Kim, C.D., Yang, Y., Shao, V., Yang, Y., Huang, W., Gao, Z., Anderson, T., Zhang, J., Jain, J., Stoica, G., Han, W., Farhadi, A., Krishna, R.: Molmo2: Open weights and data for vision-language models with video understanding and grounding. arXiv preprint arXiv:2601.10611 (2026), https://arxiv.org/abs/2601.10611
9. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes (2017), https://arxiv.org/abs/1702.04405
10. Deitke, M., Clark, C., Lee, S., Tripathi, R., Yang, Y., Park, J.S., Salehi, M., Muennighoff, N., Lo, K., Soldaini, L., Lu, J., Anderson, T., Bransom, E., Ehsani, K., Ngo, H., Chen, Y., Patel, A., Yatskar, M., Callison-Burch, C., Head, A., Hendrix, R., Bastani, F., VanderBilt, E., Lambert, N., Chou, Y., Chheda, A., Sparks, J., Skjonsberg, S., Schmitz, M., Sarnat, A., Bischoff, B., Walsh, P., Newell, C., Wolters, P., Gupta, T., Zeng, K.H., Borchardt, J., Groeneveld, D., Nam, C., Lebrecht, S., Wittlif, C., Schoenick, C., Michel, O., Krishna, R., Weihs, L., Smith, N.A., Hajishirzi, H., Girshick, R., Farhadi, A., Kembhavi, A.: Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In: CVPR (2025), https://arxiv.org/abs/2409.17146
11. Deitke, M., VanderBilt, E., Herrasti, A., Weihs, L., Salvador, J., Ehsani, K., Han, W., Kolve, E., Farhadi, A., Kembhavi, A., Mottaghi, R.: Procthor: Large-scale embodied ai using procedural generation (2022), https://arxiv.org/abs/2206.06994

12. Deng, C., Zhu, D., Li, K., Gou, C., Li, F., Wang, Z., Zhong, S., Yu, W., Nie, X., Song, Z., Shi, G., Fan, H.: Emerging properties in unified multimodal pretraining. arXiv preprint arXiv:2505.14683 (2025)
13. Dumery, C., Etté, N., Fan, A., Li, R., Xu, J., Le, H., Fua, P.: Counting stacked objects. In: ICCV (2025)
14. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12873–12883 (2021)
15. Gu, J., Hao, Y., Wang, H.W., Li, L., Shieh, M.Q., Choi, Y., Krishna, R., Cheng, Y.: ThinkMorph: Emergent properties in multimodal interleaved chain-of-thought reasoning. arXiv preprint arXiv:2510.27492 (2025)
16. Hu, Y., Liu, B., Kasai, J., Wang, Y., Ostendorf, M., Krishna, R., Smith, N.A.: Tifa: Accurate and interpretable text-to-image faithfulness evaluation with question answering (2023), https://arxiv.org/abs/2303.11897
17. Hu, Y., Shi, W., Fu, X., Roth, D., Ostendorf, M., Zettlemoyer, L., Smith, N.A., Krishna, R.: Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. arXiv preprint arXiv:2406.09403 (2024)
18. Jia, M., Qi, Z., Zhang, S., Zhang, W., Yu, X., He, J., Wang, H., Yi, L.: Omnispatial: Towards comprehensive spatial reasoning benchmark for vision language models. arXiv preprint arXiv:2506.03135 (2025)
19. Kamath, A., Hessel, J., Chang, K.W.: What's "up" with vision-language models? Investigating their struggle with spatial reasoning. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) (2023)
20. Khirodkar, R., Bansal, A., Ma, L., Newcombe, R., Vo, M., Kitani, K.: Egohumans: An egocentric 3d multi-human benchmark (2023)
21. Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Deitke, M., Ehsani, K., Gordon, D., Zhu, Y., et al.: Ai2-thor: An interactive 3d environment for visual ai. arXiv preprint arXiv:1712.05474 (2017)
22. Li, C., Wu, W., Zhang, H., Xia, Y., Mao, S., Dong, L., Vulić, I., Wei, F.: Imagine while reasoning in space: Multimodal visualization-of-thought. arXiv preprint arXiv:2501.07542 (2025)
23. Li, L., Chen, X., Chen, P., et al.: ViewSpatial-Bench: Evaluating multi-perspective spatial understanding of vision-language models. arXiv preprint arXiv:2505.21500 (2025)
24. Liu, F., Emerson, G., Collier, N.: Visual spatial reasoning. arXiv preprint arXiv:2205.00363 (2022)
25. Ma, W., Chen, H., Zhang, G., Chou, Y.C., Chen, J., de Melo, C.M., Yuille, A.: 3DSRBench: A comprehensive 3D spatial reasoning benchmark. In: ICCV (2025)
26. Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., Parikh, D., Batra, D.: Habitat: A Platform for Embodied AI Research. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
27. OpenAI: Thinking with images. OpenAI Blog (2025), https://openai.com/index/thinking-with-images/
28. Puig, X., Undersander, E., Szot, A., Cote, M.D., Partsey, R., Yang, J., Desai, R., Clegg, A.W., Hlavac, M., Min, T., Gervet, T., Vondruš, V., Berges, V.P., Turner, J., Maksymets, O., Kira, Z., Kalakrishnan, M., Malik, J., Chaplot, D.S., Jain, U., Batra, D., Rai, A., Mottaghi, R.: Habitat 3.0: A co-habitat for humans, avatars and robots (2023)

29. Ray, A., Abdelkader, A., Mao, C., Plummer, B.A., Saenko, K., Krishna, R., Guibas, L., Chu, W.S.: Mull-tokens: Modality-agnostic latent thinking. arXiv preprint arXiv:2512.10941 (2025)
30. Ray, A., Duan, J., Brown, E., Tan, R., Bashkirova, D., Hendrix, R., Ehsani, K., Kembhavi, A., Plummer, B.A., Krishna, R., Zeng, K.H., Saenko, K.: Sat: Dynamic spatial aptitude training for multimodal language models (2025), https://arxiv.org/abs/2412.07755
31. Szot, A., Clegg, A., Undersander, E., Wijmans, E., Zhao, Y., Turner, J., Maestre, N., Mukadam, M., Chaplot, D., Maksymets, O., Gokaslan, A., Vondrus, V., Dharur, S., Meier, F., Galuba, W., Chang, A., Kira, Z., Koltun, V., Malik, J., Savva, M., Batra, D.: Habitat 2.0: Training home assistants to rearrange their habitat. In: Advances in Neural Information Processing Systems (NeurIPS) (2021)
32. Tschannen, M., Gritsenko, A., Wang, X., Naeem, M.F., Alabdulmohsin, I., Parthasarathy, N., Evans, T., Beyer, L., Xia, Y., Mustafa, B., et al.: Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. arXiv preprint arXiv:2502.14786 (2025)
33. Wang, S., Pei, M., Sun, L., Deng, C., Shao, K., Tian, Z., Zhang, H., Wang, J.: Spatialviz-bench: An mllm benchmark for spatial visualization. arXiv preprint arXiv:2507.07610 (2025)
34. Wang, W., Tan, R., Zhu, P., Yang, J., Yang, Z., Wang, L., Kolobov, A., Gao, J., Gong, B.: Site: towards spatial intelligence thorough evaluation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9058–9069 (2025)
35. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., Zhou, D.: Chain-of-thought prompting elicits reasoning in large language models. arXiv preprint arXiv:2201.11903 (2022)
36. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., ming Yin, S., Bai, S., Xu, X., Chen, Y., Chen, Y., Tang, Z., Zhang, Z., Wang, Z., Yang, A., Yu, B., Cheng, C., Liu, D., Li, D., Zhang, H., Meng, H., Wei, H., Ni, J., Chen, K., Cao, K., Peng, L., Qu, L., Wu, M., Wang, P., Yu, S., Wen, T., Feng, W., Xu, X., Wang, Y., Zhang, Y., Zhu, Y., Wu, Y., Cai, Y., Liu, Z.: Qwen-image technical report (2025), https://arxiv.org/abs/2508.02324
37. Wu, H., Huang, X., Chen, Y., Zhang, Y., Wang, Y., Xie, W.: Spatialscore: Towards unified evaluation for multimodal spatial understanding. arXiv e-prints pp. arXiv–2505 (2025)
38. Xie, J., Yang, Z., Shou, M.Z.: Show-o2: Improved native unified multimodal models. arXiv preprint arXiv:2506.15564 (2025)
39. Yang, J., Yang, S., Gupta, A.W., Han, R., Fei-Fei, L., Xie, S.: Thinking in space: How multimodal large language models see, remember, and recall spaces. arXiv preprint arXiv:2412.14171 (2025)
40. Yang, K., Russakovsky, O., Deng, J.: SpatialSense: An adversarially crowdsourced benchmark for spatial relation recognition. In: ICCV (2019)
41. Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth anything: Unleashing the power of large-scale unlabeled data. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10371–10381 (2024)
42. Yang, R., Zhu, Z., Li, Y., Huang, J., Yan, S., Zhou, S., Liu, Z., Li, X., Li, S., Wang, W., et al.: Visual spatial tuning. arXiv preprint arXiv:2511.05491 (2025)
43. Yang, S., Xu, R., Xie, Y., et al.: MMSI-Bench: A benchmark for multi-image spatial intelligence. arXiv preprint arXiv:2505.23764 (2025)

44. Yang, Z., Yu, X., Chen, D., Shen, M., Gan, C.: Machine mental imagery: Empower multimodal reasoning with latent visual tokens. arXiv preprint arXiv:2506.17218 (2025)
45. Yeh, C.H., Wang, C., Tong, S., Cheng, T.Y., Wang, R., Chu, T., Zhai, Y., Chen, Y., Gao, S., Ma, Y.: Seeing from another perspective: Evaluating multi-view understanding in mllms. arXiv preprint arXiv:2504.15280 (2025)
46. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: Scannet++: A high-fidelity dataset of 3d indoor scenes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12–22 (2023)
47. Yin, B., Wang, Q., Zhang, P., et al.: Spatial mental modeling from limited views. arXiv preprint arXiv:2506.21458 (2025)
48. Zhang, P., Huang, Z., Wang, Y., Zhang, J., Xue, L., Wang, Z., Wang, Q., Chandrasegaran, K., Zhang, R., Choi, Y., Krishna, R., Wu, J., Fei-Fei, L., Li, M.: Theory of space: Can foundation models construct spatial beliefs through active exploration? In: International Conference on Learning Representations (ICLR) (2026)

A Training Details and Hyperparameters

A.1 Training Setup

We fine-tune BAGEL-7B-MoT [12] using PyTorch FSDP (Fully Sharded Data Parallel) with `bf16` mixed precision on 8 NVIDIA A100 80 GB GPUs. Table 7 summarizes the key hyperparameters.

Table 7: Training hyperparameters.

Parameter	**Value**
Optimizer	AdamW (β_1=0.9, β_2=0.95, ϵ=10^{-15})
Learning rate	1×10^{-5} (constant after warmup)
Warmup steps	2,000
Gradient clipping	max norm = 1.0
EMA decay	0.9999
Max tokens per batch	32,768
Max tokens per sample	24,576
Input image resolution	1024×1024 (PET) / 512×512 (PT, MVC)
IPT latent resolution	64×64 (Latent-64)
Flow-matching loss weight (λ_{fm})	1.0
Language modeling loss weight (λ_{lm})	1.0
Frozen modules	VAE encoder & decoder
Fine-tuned modules	LLM (all layers), ViT encoder, connector

System prompts. All training modes share one of two system prompts prepended to every input:

- **Thinking prompt** (used for IPT, Text CoT, and label-only):
  Let's think step by step to answer the question. For text-based thinking, enclose the process within `<think> </think>`*. For visual thinking, enclose the content within* `<image_start> </image_end>`*. Finally conclude with the final answer wrapped in* `<answer></answer>` *tags.*
- **Answer-only prompt** (used for the answer-only portion of mixed training):
  Answer the question directly. Wrap your answer in `<answer></answer>` *tags. Do not think or generate any images.*

Training modes. Table 8 summarizes the five training configurations evaluated in this work. Each mode differs in the output format the model is trained to produce, and correspondingly in which loss terms are active. In IPT mode, both $\mathcal{L}_{fm}$ and $\mathcal{L}_{lm}$ are active; in Text CoT and label-only, only $\mathcal{L}_{lm}$ is used.
Mixed training. Mixed training combines 50% IPT examples (with the thinking prompt and visual generation targets) and 50% answer-only examples (with the answer-only prompt and direct answers). The two data subsets are mixed

Table 8: Training modes and their output formats. `[IMG]` denotes the generated intermediate image tokens.

Mode	Prompt	Training target
Label-only	Think	`<answer>A</answer>`
Text CoT	Think	`<think>`*reasoning*`</think>` `<answer>A</answer>`
IPT	Think	`<think>`*task prompt*`</think>` `<image_start>[IMG]<image_end>` `<answer>A</answer>`
Mixed	50% Think / 50% Ans-only	50% IPT + 50% answer-only

at the dataloader level: both dataset names and sample counts are specified in the training configuration, and the dataloader interleaves batches from both sources. The model learns to switch between generating imaginative perception tokens and producing direct answers based on which system prompt is provided, enabling a single checkpoint to operate in either mode at inference time.

Text CoT generation. Text chain-of-thought training targets are generated by GPT-5.1 using ground-truth scene metadata from the simulator. For each training example, GPT-5.1 receives the input image, the ground-truth answer, and a task-specific instruction, and produces a step-by-step textual reasoning trace (100–300 words). Below we describe the task-specific prompts.

Path tracing CoT prompt. The system prompt instructs the model to act as a spatial reasoning AI that solves indoor navigation questions by analyzing top-down views step by step. The task instruction is:

> *You are navigating along a numbered path through an indoor scene. The top-down view shows the path with waypoints and midpoints. You need to determine what object is visible from a specific side (left or right) at a midpoint.*
> *Reason step by step: (1) Identify the path direction (which waypoint to which). (2) Determine your orientation at the midpoint. (3) Figure out what "left" or "right" means given that orientation. (4) Analyze the top-down layout to identify objects on that side. (5) Compare against the answer choices and eliminate wrong ones.*

Figure 3 shows three Text CoT training examples for path tracing in the EgoDir setting, illustrating how the generated reasoning traces analyze the top-down layout, determine the agent's orientation at the midpoint, and systematically eliminate incorrect answer choices.

Perspective Taking CoT prompt. For Perspective Taking, the model receives the input image together with privileged hidden information, the ground-truth answer and a second image showing the target view after the camera motion, to ensure correctness. The generated chain-of-thought must be written as if only

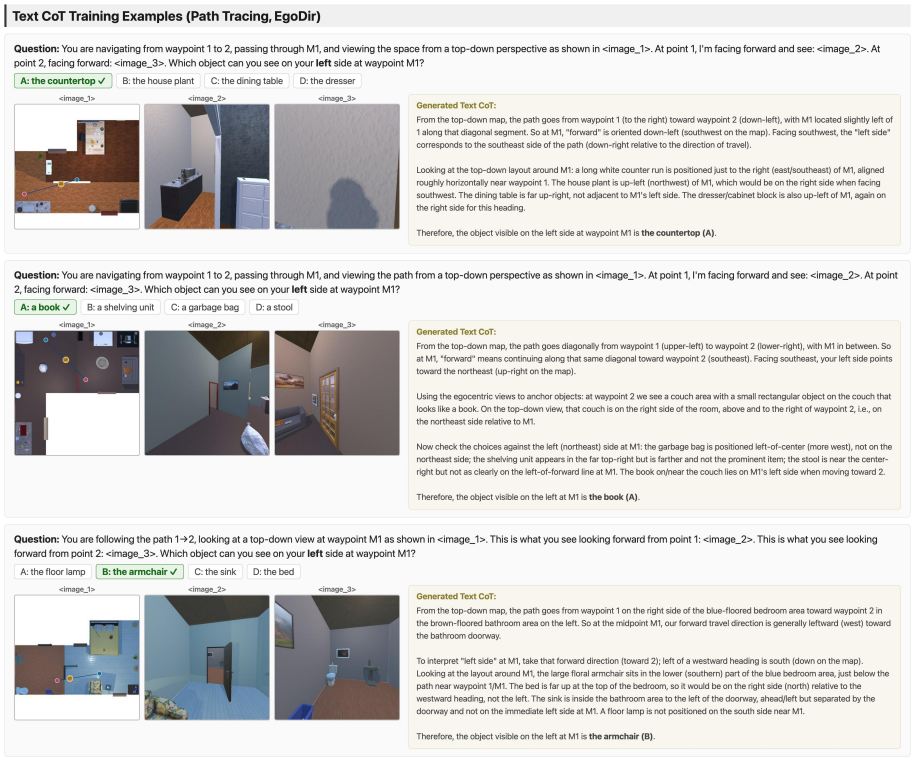

Fig. 3: Text CoT training examples for path tracing (EgoDir setting). Each example shows the input images (top-down map and egocentric views at endpoints), the question with answer choices, and the GPT-5.1-generated reasoning trace. The reasoning follows the structured prompt: identifying path direction, determining orientation at M_1, interpreting left/right relative to that orientation, and eliminating distractors.

the original image and question were available, without referencing any hidden information. The prompt is:

> *You generate student-facing chain-of-thought explanations for visual navigation/spatial reasoning from a single image. You may receive hidden privileged information (metadata, ground-truth answer, and a second image) to ensure correctness, but your explanation must be written as if you only saw the original image and question. Do not mention, quote, paraphrase, or allude to any hidden information.*
>
> *Student-visible input (ONLY what the explanation may reference): (1) Image A (original scene with a red "X" on the floor). (2) Question: {Q}*
> *Hidden privileged information (FOR CORRECTNESS ONLY — MUST NOT APPEAR IN THE EXPLANATION): (1) Correct answer: {A} (2) Image B (ground-truth final view after moving and turning)*
>
> *Task: Write a chain-of-thought that a student could produce from Image A and the question alone (not more than 10 sentences).*
> *Hard constraints: Do NOT mention Image B, metadata, coordinates, angles, yaw/pitch/roll, agent states, movement info, IDs, or any numbers from the hidden block. The explanation must be plausible using only Image A: talk only about visible objects, their relative positions, and the effect of turning left/right. Keep it short (3–6 steps). No extra commentary. End with the correct final answer:* My answer is [A]/[B]. *Do not answer based on your own perspective; stick to the ground-truth answer provided.*
>
> *Output format:*
> *Step 1: Identify the object in the question and describe its relative location in the original view.*
> *Step 2: Describe the walk-to-X action (translation without changing facing direction).*
> *Step 3: Describe the turn action (update egocentric left/right).*
> *Step 4: Conclude whether the object is on the left or right relative to the agent.*

An example of a generated thought:

> *Step 1: The oven is on the right wall of the kitchen, a bit ahead of the current viewpoint, and the red X is on the floor between the viewer and that oven wall. Step 2: Walking straight to the X without turning keeps me facing the same direction, so I move forward toward the area in front of the oven and end up nearer that wall. Step 3: From the X, turning right 90 degrees means I would now face the wall with the oven and cabinets. Step 4: Since I walked closer to that wall before turning, I end up standing nearer to the oven than I was at the start. My answer is [A].*

Multiview Counting CoT prompt. For Multiview Counting, the model receives per-frame metadata listing which object instances (by ID) are visible in each frame, and is asked to produce a frame-by-frame reasoning trace without directly referencing the hidden instance IDs. The prompt is:

> *You are given an object-counting question based on multiple images from the same scene. You will be provided with: the target object, the correct total count,*

and frame-level information listing which object instances are visible in each frame (this information is hidden and should not be referenced directly).

IMPORTANT: An empty list for a frame means that no objects of the target type are visible in that frame.

Write a brief, frame-by-frame explanation describing what is visible in each frame. Do not mention object IDs or refer to them explicitly. When explaining each frame, do not count objects that were already visible in previous frames. After the frame-by-frame explanation, conclude with: "The total number is X."

Object: {O}
Correct total count: {answer}
Frames: {frames_ text}

An example of a generated thought:

Frame 1: No bowls are visible. Frame 2: A bowl appears and is visible for the first time. Frame 3: No new bowls appear; the same bowl from before may still be present. The total number is 1.

A.2 Evaluation Setup

Inference parameters. Table 9 lists the inference hyperparameters used across all evaluations.

Table 9: Inference hyperparameters for evaluation.

Parameter	**Value**
Text temperature	0.3
Sampling	do_sample = True
Max thinking tokens	4,096
Diffusion timesteps	50
Timestep shift	3.0
Text CFG scale	4.0
Image CFG scale	2.0
CFG interval	$[0.0, 1.0]$
Generated image resolution	1024×1024
Max generation rounds	1
GPU allocation	2× A100 80 GB (model parallelism)

Evaluation modes. At inference, each model variant is evaluated in the mode that matches its training configuration. IPT models are evaluated in two settings: (1) *imagination mode*, where the model generates an intermediate image before answering, and (2) *answer-only mode*, where the model produces only a text answer without generating any image. For models trained with visual generation (IPT, Mixed), the VAE weights are always loaded (`visual_gen=True`), and input

images are encoded through both the ViT and VAE pathways (`vae_input=True`) to match the training-time encoding. Without setting `vae_input=True`, a train–eval mismatch would occur: during training, input images pass through both VAE and ViT, but the default evaluation behavior sends inputs through ViT only.[6]

Answer extraction. We extract the predicted answer letter from model outputs using a cascading rule-based procedure: (1) parse `<answer>X</answer>` tags; (2) extract from `\boxed{X}` format; (3) match patterns such as "the answer is X"; (4) detect bold letter formatting (`**X**`); (5) fall back to the last single letter in the response. All benchmarks use the same unified scoring function that compares the extracted letter against the ground-truth answer.

B Data Curation Details

B.1 Path Tracing

We generate path tracing data from two sources: AI2-THOR (synthetic) and Matterport3D (real-world).

AI2-THOR

Scene selection. We use 120 standard iTHOR [21] scenes spanning four room types (kitchens, living rooms, bedrooms, bathrooms), with 30 scenes per type, split into train (20), val (5), and test (5) per type. The training set additionally incorporates procedurally generated houses from ProcTHOR-10k [11], which include a fifth room type (hallways, offices, dining rooms).

Path sampling. For each scene, we sample feasible two-waypoint paths on the navigation mesh, balanced across room types and three distance bins: short (1–2 m), medium (2–4 m), and long ($\geq$4 m). Grid-based path sampling uses a spacing of 0.5 m with a minimum waypoint separation of 1.0 m.

Camera configuration. All views are rendered at 1024×1024 resolution. To increase viewpoint diversity, we randomize the camera height (sampled from seven values between 1.4 and 1.8 m), field of view (75–120), and pitch (-5 to 5).

Rendering. At each sample we render top-down views, egocentric forward views at both endpoints, and a sweep of candidate sideviews at the midpoint M_1. The sideview sweep covers 7 yaw angles $\times$ 7 horizontal offsets $\times$ 3 pitch values, yielding 147 candidate views per midpoint. We select the sideview that best exposes the queried object using simulator segmentation masks, requiring a minimum object coverage of 0.15% of the image area and a maximum view angle of 90 relative to the path direction.

[6] For Path-Tracing, we found that setting `vae_input=False` actually improves generalization to real environments.

Question generation. Questions are generated from templates ("Which object can you see on your {side} at waypoint M1?") with four choices: the correct answer drawn from verified visible objects and three distractors drawn from the opposite side or a global object pool. Each base MCQ is expanded into eight input variants by combining different image types (top-down path, top-down with arrow, top-down with midpoint marker, dollhouse view) and egocentric cue availability (with/without endpoint views).

TIFA filtering. We apply TIFA-style filtering [16] to ensure question quality. Each candidate is decomposed into binary visibility queries and verified by GPT-4.1 with three-round majority voting, using early exit after round 2 when unanimous. Samples are dropped if (1) the correct answer is not visible in the sideview, (2) a distractor is also visible in the sideview, or (3) the model answers incorrectly even when provided the sideview. We further remove samples where GPT-4.1 answers correctly from the top-down view and egocentric endpoint views alone, ensuring the benchmark requires genuine spatial imagination.

Debiasing. Answer choices are reshuffled with per-sample deterministic seeds to remove positional bias. We additionally verify that per-object and per-room answer distributions remain approximately uniform.

Statistics. The synthetic training set contains 11,204 examples.

Real-World Data (Matterport3D) To evaluate cross-domain transfer, we construct a real-world test set from Matterport3D [6] top-down views.

Image collection. We collect top-down screenshots from Matterport 3D indoor tours, capturing per-floor views with UI elements removed and dark borders cropped.

Auto-annotation. We annotate walking paths on each image using a two-pass GPT pipeline. In the first pass, GPT proposes N candidate walking paths, each defined by three waypoints (start "1", midpoint "M1", end "2") placed on open floor areas. In the second pass, the proposed waypoints are drawn on the image and GPT is asked to (a) verify that M_1 lies on walkable floor, adjusting its position if necessary, and (b) identify 2–5 visible furniture items or objects on each side (left/right) of the path at M_1.

Post-processing. Several geometric filters are applied to ensure path quality. Waypoints are clamped to image bounds, and paths shorter than 30% of the shorter image dimension are rejected. M_1 is snapped onto the line segment between waypoints 1 and 2 and constrained to the $[0.2, 0.8]$ interval to avoid proximity to endpoints. Paths that traverse dark or background regions (more than 20% dark pixels along the path) are discarded.

TIFA filtering. Because no side-view images exist for real environments, TIFA verification operates on the top-down image only. We verify three properties: (1) all three waypoints lie on walkable floor (not on furniture, walls, or background); (2) each annotated object is visible in the image; and (3) each object is on the correct side of the path. Paths with fewer than 2 verified objects on either side are dropped. Majority voting across up to 3 rounds is used, with early exit when the first two rounds agree.

Human review. After automated filtering, all surviving annotations undergo human review, where annotators can approve, delete, or edit individual paths and their associated object lists.

Question generation. Each verified path yields eight question variants (forward/reverse $\times$ left/right $\times$ arrow/no-arrow). When the walking direction is reversed ($2 \rightarrow 1$), left and right swap because the agent faces the opposite direction. Distractors are drawn from opposite-side objects first, then supplemented from a global pool of 50 common indoor objects. Answer choices are shuffled with per-sample deterministic seeds to eliminate positional bias. Because the real environments lack egocentric viewpoints, we evaluate on the Path and PathArr settings only. The real-world benchmark contains 332 human-verified questions.

B.2 Perspective Taking

We generate perspective taking data from three sources: AI2-THOR (synthetic), Habitat (photorealistic scans), and Visual Spatial Tuning (real-world images). All sources share the same task structure—given a first-person view with a marked target position, answer a spatial question about the scene from the new viewpoint—but differ in visual domain and 3D engine.

AI2-THOR

Scene selection. We use procedurally generated indoor scenes from ProcTHOR [11], which provides diverse house layouts with varied room configurations. For each scene, we sample multiple camera positions from the navigable area and generate perspective-taking examples at each position.

Target position placement. The target position (marked with a red "X" on the input image) is determined by raycasting from the camera into the scene. To ensure physically plausible movement targets, we filter ray hits to *ground-only* surfaces: floors, carpets, rugs, and tiles. Hits on furniture, tabletops, or other elevated surfaces are rejected. The hit point must lie within $\pm 0.2\,\text{m}$ of the ground-level height. Up to 40 raycasting attempts are made per sample; if no valid ground hit is found, the sample is skipped.

Viewpoint transformation. The agent is teleported to the target position and rotated 90 in a randomly chosen direction (left or right with equal probability), simulating a realistic movement-and-turn action. A ground-truth novel-viewpoint image is rendered at this new pose to serve as the imaginative perception target.

Object filtering. To ensure well-defined questions, target objects must satisfy several criteria:

- Visible in *both* the original and new viewpoint (dual-view visibility).
- Occupy at least 0.4% of the image area.
- Lie within 5 m of the camera.
- Fall at least 150 px from the image edge (to avoid partially visible objects).
- For relative position questions: the object must be unambiguously on one side of the image center, with a 150 px margin from the center line.

Additionally, we enforce a *left-right eligibility* constraint: an object is only used for relative position questions if it appears on at most one side of the image (not straddling the center), ensuring that the left/right answer is unambiguous.
Question generation. We generate two types of questions with 10 template variants each, varying in person perspective (first, second, third person) and formality level:

- **Distance change**: "After moving to 'X' and turning {direction} 90, will the {object} get closer or further?" Requires a minimum distance change of ± 0.5 m between the old and new camera positions.
- **Relative position**: "After moving to 'X' and turning {direction} 90, will the {object} be on your left or right?" Left/right is determined by the object's 2D position in the new-viewpoint image.

Sub-categories. As described in Sec. 3.1, the six balanced sub-categories arise from the combination of question type and answer: two for distance change (*closer*, *further*) and four for relative position (*left*→*left*, *left*→*right*, *right*→*left*, *right*→*right*), where the notation indicates the object's lateral position before and after the viewpoint transformation. The training set is balanced across all six sub-categories.
Image annotation. Each input image is annotated in two versions: (1) with only a red "X" marking the target position, and (2) with the "X" plus a blue directional arrow indicating the agent's facing direction after rotation. The arrow version provides an additional spatial cue at evaluation time.

Statistics. The AI2-THOR training set contains 20,531 examples across 98 scenes, with an average of ∼210 questions per scene.

Habitat Scene source. We use photorealistic 3D scans from HM3D (Habitat-Matterport 3D) [6] with semantic annotations. Only single-floor scenes are selected (floor-level Y-variance < 2.0 m) to avoid cross-level ambiguities.

Camera configuration. Images are rendered at 1024×1024 resolution with a horizontal field of view of 90. The sensor height is set to 1.25 m above the navigable floor surface, matching a standing human eye level.
Target position and viewpoint. Camera A (original viewpoint) is placed at a random navigable point with a random yaw. The target position (Camera B) is

determined by selecting a visible object as an anchor: Camera B is placed at the object's XZ coordinates, offset slightly along the facing direction to avoid clipping into geometry, and snapped to the nearest navigable point on the mesh (within a 1.0 m snap radius). The ground-truth novel-viewpoint image is rendered at Camera B's position and orientation.

Object filtering. We apply a strict whitelist of ∼70 mainstream furniture categories (seating, tables, beds, storage, appliances, bathroom fixtures) and exclude structural elements (walls, floors, doors). Objects must occupy at least 0.8% of the image area in the original frame and at least 0.5% in the imagined frame. To avoid ambiguous references, only objects whose category is *unique* in the frame are used (*e.g.*, if two chairs are visible, neither is selected as a question target). An edge margin of 200 px is applied.

Left/right determination. Object laterality is determined by the mean x-coordinate of the object's semantic segmentation mask in the rendered image, with a 180 px margin from the image center (512 px). Objects falling in the center zone ($332 < x < 692$) are excluded as ambiguous.

Statistics. The Habitat training set contains 19,998 examples balanced across the six sub-categories.

Real-World Data (VST) To bridge the synthetic-to-real domain gap, the *mixed* training variant incorporates 15,000 real-world examples drawn from the camera motion subset of the Visual Spatial Tuning (VST) dataset [42]. Each example contains a pair of multi-view images captured from different viewpoints in real indoor scenes, along with a question about the camera motion between them and a corresponding answer.

Filtering uncertain answers. We first filter out examples whose answers are uncertain or underspecified using GPT-5.1, prompting it as a binary classifier. Any example whose answer contains phrases such as "cannot be determined," "unknown," or "insufficient information" is removed. The filtering prompt is:

> *You are a binary classifier.*
> *Proposed Answer:*
> *<start_answer>{A}<end_answer>*
> *If the answer contains ANY of the following phrases or meanings, output "NO":*
> - *cannot be determined*
> - *insufficient information*
> - *not enough information*
> - *unknown*
> - *unclear*
> - *cannot tell*
> - *impossible to determine*
> - *indeterminate*
>
> *Only output "YES" if the answer clearly states a specific, determined choice (e.g., a direction, location, label, or concrete option).*
> *Output exactly one token: YES or NO.*

Rewriting into generation prompts. After filtering, we use GPT-5.1 to rewrite each question, answer pair into a generation prompt describing the camera motion *from the first image to the second*. This requires careful handling of reference frame direction: if the original question asks where the first camera is relative to the second image, the motion direction must be inverted before constructing the prompt. The rewriting prompt is:

> *You are given a question about camera motion between two images and its correct answer.*
>
> *IMPORTANT INVERSION RULE:*
> - *If the question asks "where is the FIRST camera relative to the SECOND image" (or uses the second image as reference), then the answer describes motion FROM second TO first.*
> - *In this case, you MUST invert the direction to get motion FROM first TO second.*
> - *If the question asks "where is the SECOND camera relative to the FIRST image" (or uses the first image as reference), NO inversion is needed.*
>
> *STEPS:*
> 1. *Determine which image is the reference point in the question.*
> 2. *If the reference is the second image, invert the direction in the answer.*
> 3. *Using the final motion FROM first TO second, create a generation prompt.*
>
> *INVERSION EXAMPLES:*
> - *"right" → "left"*
> - *"left" → "right"*
> - *"front" → "back"*
> - *"back" → "front"*
> - *"front left" → "back right"*
> - *"back right" → "front left"*
>
> *Your output should start with "generate" and describe viewing the scene from the new camera position after applying the motion from the first image.*
>
> *Question: <start_question>{Q}<end_question>*
> *Answer: <start_answer>{A}<end_answer>*
>
> *First, identify the reference image. Then apply inversion if needed. Then generate your output.*

The resulting generation prompts condition the model on the first view and the inferred camera motion, with the second view serving as the imaginative perception target. Because no programmatic 3D annotation is available for these real scenes, this data serves as a domain bridge rather than a source of the full six-sub-category question format.

Mixed training composition. The mixed PET training variant combines AI2-THOR (20,531), Habitat (19,998), and VST (15,000) examples, totaling 55,529 samples.

B.3 Multiview Counting

We construct multiview counting data from both synthetic and real-image sources. Our main training set is generated from ProcTHOR/AI2-THOR environments, which provide full 3D supervision for both egocentric observations and top-down bird's-eye-view (BEV) targets. To complement this synthetic source, we additionally curate two real-image multiview counting sets from MessyTable and ScanNet++, which expose the model to real visual appearance and partial observability under natural image statistics.

ProcTHOR / AI2-THOR We generate the main multiview counting training set from AI2-THOR environments, using two trajectory types that capture complementary modes of partial observability.

Trajectory types.

- **Rotation**: The agent remains at a fixed position and rotates in 90° increments through four cardinal directions (0°, 90°, 180°, 270°), producing four frames that together cover a 360° panorama of the surrounding area.
- **Multi-camera**: The agent traverses a square path, capturing one frame at each of four corners. This setup simulates a multi-camera rig where viewpoints are spatially distributed around the scene.

Both trajectory types produce exactly four input frames per sample.

Bird's-eye view (BEV) generation. The ground-truth intermediate image is a top-down BEV map rendered from an overhead camera in the 3D scene. To ensure the BEV only covers the *explored* area (the region visible from the input frames), we crop the map with trajectory-aware padding: 5 m around the agent position for rotation trajectories and 4 m around the traversed path for multi-camera trajectories. Object counts in the cropped BEV are validated against the segmentation maps to ensure consistency.

Object filtering and category balancing. Structural elements (walls, floors, ceilings, doorways) are excluded from counting targets. Target objects must be visible in both the first-person frames and the cropped top-down segmentation map (with a minimum coverage of 0.1%). Because initial generation heavily favors count= 1 questions (∼82%), we apply iterative rebalancing: high-frequency categories are capped at 9.9% of the dataset, and count= 1 samples are downsampled. This produces a more uniform distribution across object categories and count values.

Question format and distractor generation. Questions follow the template: "How many {category}(s) are in this area?" with four answer choices (A–D). Distractors are sampled from ±1 and ±2 of the correct count, producing plausible alternatives. Negative counts are removed, and the four options are shuffled with a per-sample deterministic seed to eliminate positional bias.

Statistics. The synthetic training set contains 17,079 examples generated from ProcTHOR [11] scenes, covering both trajectory types.

MessyTable To expose the model to real tabletop imagery with severe clutter and occlusion, we construct an additional multiview counting set from MessyTable [4]. Each scene contains multiple camera views of the same tabletop arrangement together with instance-level annotations.

Scene-level counting targets. For each scene, we aggregate annotations across all cameras and de-duplicate instance IDs across views, so that the ground-truth answer corresponds to the number of unique physical objects rather than the sum of per-view detections. Counting targets are defined at the subclass level and mapped to readable category names.

Target and view sampling. For each scene, we sample one target category from the categories present in the scene. To reduce the dominance of trivial singleton cases, sampling is biased toward categories with count ≥ 2: when both singleton and multi-instance categories are available, 90% of samples are drawn from the multi-instance bucket and 10% from the singleton bucket. Input images are selected from the eight surrounding non-top cameras, with priority given to views in which the target category is absent. When too few such views exist, they are supplemented with additional non-adjacent views to maintain viewpoint diversity. This makes the final count require aggregation across multiple views rather than inspection of a single image.

Top-view supervision and question generation. Each sample also stores the canonical top-view image as the reasoning target. In the exported JSONL format, image inputs use centered crops derived from the union of all annotated object boxes in each selected camera view, which reduces empty borders while preserving the visible object layout. Questions are instantiated from a diverse pool of natural-language counting templates such as "How many {object} are in this scene?"

ScanNet++ We further construct a real indoor multiview counting set from ScanNet++ [46], using iPhone image trajectories paired with labeled 3D scene reconstructions. Compared with MessyTable, this set covers larger indoor spaces, more varied viewpoints, and more realistic household layouts.

Top-down map and candidate view generation. For each scene, we first generate a top-down map from the labeled 3D reconstruction. Because raw point-cloud renderings are visually sparse and unrealistic, we further use Qwen-Image-Edit [36] to transform the rendered top-down visualization into a more realistic top-down image while preserving the scene layout. We then build a candidate egocentric view pool by combining a small set of canonical iPhone views with additional randomly sampled frames, requiring each extra frame to differ from the canonical views by at least a minimum yaw angle.

Visibility estimation and target selection. We estimate which object instances are visible in each candidate frame by projecting the labeled 3D scene into the camera views using mesh ray-casting, and use the semantic annotations to obtain scene-level category counts. Top-down maps are filtered by automatic quality rules to remove blurry, blank, or low-texture renderings. Candidate counting targets are restricted to non-structural object categories with bounded scene-level counts and sufficient visible support in the candidate views. To avoid metadata leakage, target selection is performed in a blind setting: the model is shown the top-down image and candidate labels, but not their annotated counts, and we keep only categories whose visually predicted count matches the ground truth. When multiple valid categories remain, we preferentially sample categories with counts greater than one.

Final evidence image selection. The final evidence set is selected in two stages. We first greedily choose images that jointly cover all instances of the target category while enforcing a minimum yaw-separation constraint. We then fill the remaining slots with views that add new foreground content and viewpoint diversity. Each final sample contains 5–8 egocentric images together with the top-down map as the reasoning image, and the question is rewritten into a natural counting form.

C Comparison with Related Spatial Benchmarks

IPT uniquely combines multiple components that existing spatial reasoning benchmarks lack. While prior work has focused on specific aspects—such as evaluation metrics, real-world data collection, or thought reasoning—IPT simultaneously provides training data, a scalable generation pipeline, paired ground-truth visual and textual thoughts, and a complete model training framework. Table 10 summarizes how IPT compares to related work across these key dimensions.

Table 10: Comparison of IPT with related spatial reasoning benchmarks.

	Ours	**SpatialViz** [33]	**SITE** [34]	**OmniSpatial** [18]	**SpatialScore** [37]	**MMSI** [43]
Training data	✓	×	×	×	✓	×
Scalable gen.	✓	×	×	×	✓	×
GT visual thoughts	✓	×	×	partial	×	×
GT text thoughts	✓	×	×	×	×	✓
Model framework	✓	×	×	×	✓	×
Real-world samples	✓	×	✓	✓	✓	✓

D Additional Results

Table 2 in the main paper reports path tracing accuracy averaged across input settings. Table 11 provides the full per-split breakdown for both AI2-THOR

(EgoDir, Path, PathArr) and different-environment (Real, Real+Arr) benchmarks.

Table 11: Path tracing per-split results. Accuracy (%) broken down by input setting. The main paper reports the average across these splits. For our models, accuracy reports the maximum between answer-only and free-generation inference. Best per group in **bold**.

	AI2-THOR			**Different Env.**	
Model	EgoDir	Path	PathArr	Real	Real+Arr
VQA Models					
GPT-5	**61.1**	**56.8**	**62.6**	**74.5**	87.3
GPT-5.2	22.1	40.2	36.3	57.5	68.4
Gemini 2.5 Flash	45.1	37.9	41.5	58.0	84.8
Gemini 3 Flash	48.7	39.1	39.2	70.1	**96.2**
InternVL3.5-8B	43.4	29.0	35.1	45.4	49.4
Qwen2.5-VL-7B	44.2	32.5	35.1	47.1	42.4
Qwen3-VL-8B	31.9	33.7	42.1	52.9	75.3
Unified Models					
Janus-Pro-7B	36.3	30.2	33.9	34.5	36.1
Chameleon 7B	5.3	23.7	19.9	23.0	25.9
Ours (fine-tuned BAGEL)					
Bagel (base)	36.3	26.0	27.5	39.7	45.6
Bagel (label-only)	**73.5**	61.5	62.0	46.6	62.7
+ Text CoT	53.1	47.9	48.0	**52.3**	51.3
+ IPT	61.1	43.2	42.7	46.6	**68.4**
+ Mixed Training	71.7	**65.1**	**63.2**	50.6	66.5

E Additional Baseline Models

E.1 Model Selection

We report results for **Show-o2 7B** [38] and **ThinkMorph 7B** [15], a variant of Bagel with alternative fine-tuning configurations, alongside several additional baselines to provide comprehensive evaluation coverage.

E.2 Comprehensive Baseline Comparison

In addition to the primary baselines, we evaluate several variants:

- *Human baseline*: Expert performance on each task
- *Random baseline*: Chance-level performance
- *Bagel (text-only)*: Bagel without access to image prompts

Results across all baselines are presented in Table 12.

Table 12: Performance comparison across baseline models and variants.

Model	PT.EgoDir	MVC	Hab. PET
Human	80.3%	73.8%	96.3%
Random	25.0%	25.0%	50.0%
Bagel (text-only)	15.0%	28.8%	51.1%
ThinkMorph	40.7%	38.8%	46.6%
Show-o2 7B	51.0%	35.4%	48.6%
IPT (ours)	**61.1%**	**67.3%**	**87.0%**

F Visualizations

F.1 Path Tracing

Inference with imaginative perception. Figure 4 shows examples of path tracing inference with imaginative perception tokens in the EgoDir setting. Path tracing presents a particularly challenging imagination target: the model must synthesize a first-person sideview at midpoint M_1 from a top-down map and two egocentric endpoint views, requiring accurate reasoning about camera height, 3D object layout, and occlusion from a bird's-eye representation. As shown in the figure, the generated visual thoughts are often spatially imprecise, with noticeable artifacts and layout errors compared to ground-truth sideviews. Despite this, the model frequently arrives at the correct answer (rows 1 and 2). This observation suggests that the value of imaginative perception training lies not in producing pixel-accurate intermediate images, but in encouraging the model to internalize spatial reasoning during training. The imagination supervision acts as an auxiliary signal that shapes the model's internal spatial representations, enabling it to reason about 3D visibility even when the externalized image is imperfect. This is further supported by the finding in our main experiments that IPT-trained models achieve strong performance in answer-only mode, where no image is generated at inference time.

Dataset examples. Figure 5 shows representative examples from both the real-world (Matterport3D) and synthetic (AI2-THOR/ProcTHOR) path tracing datasets across all input settings. The real-world examples (top) use photographic top-down views and are evaluated in the Path and PathArr settings only, since real environments do not provide egocentric viewpoints at arbitrary positions. The synthetic examples (bottom) include ground-truth sideview images rendered at the midpoint M_1, which serve as the imaginative perception target during training. The PathArr setting provides a directional arrow at M_1 indicating the query direction, while the EgoDir setting provides egocentric forward views at both endpoints in addition to the top-down map.

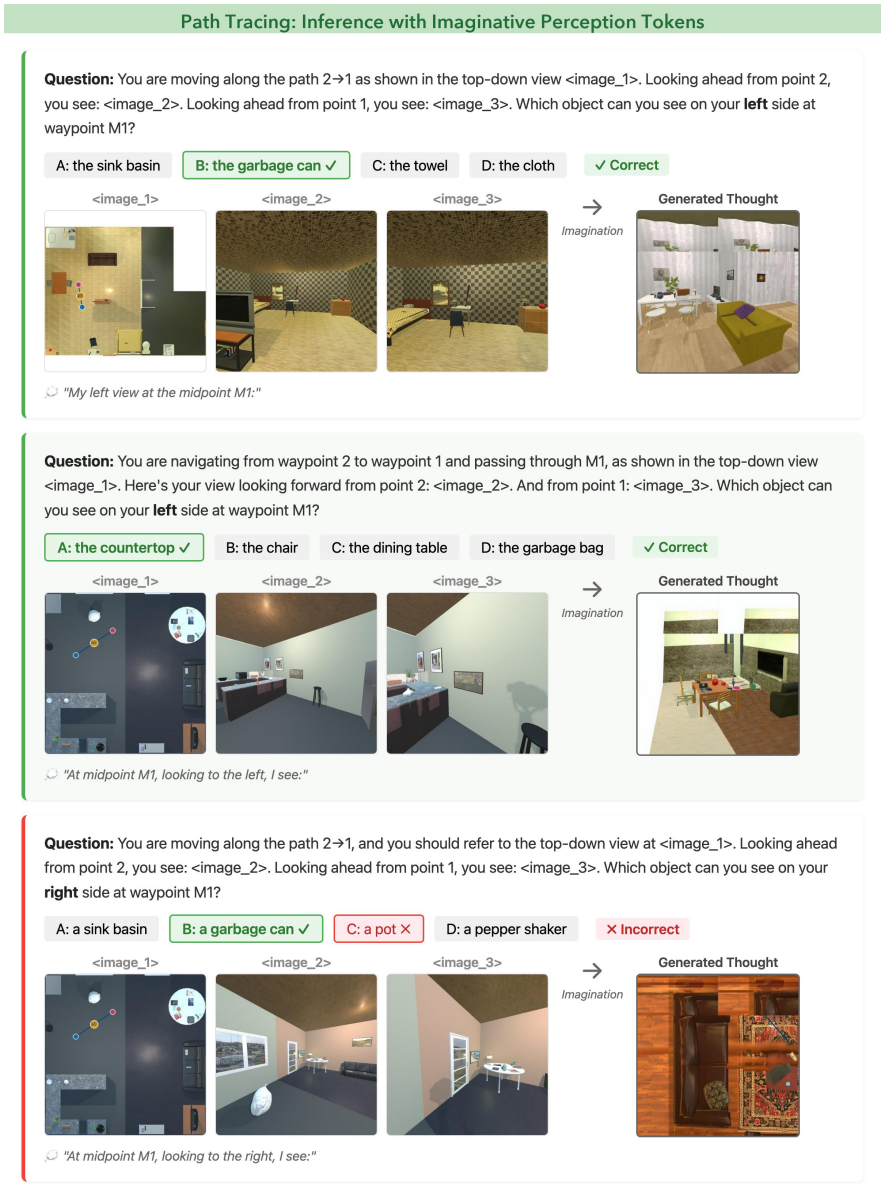

Fig. 4: Path tracing with imaginative perception tokens (EgoDir setting). The model receives a top-down map (`<image_1>`) and egocentric views at the two endpoints (`<image_2>`, `<image_3>`), generates a visual thought (imagined sideview at M_1), and predicts an answer. Although the generated thoughts exhibit spatial imprecision and artifacts, the model still arrives at the correct answer in the first two examples, suggesting that imagination training encourages internalized spatial reasoning rather than reliance on pixel-accurate intermediate outputs. The third row shows a failure case. Correct answers are highlighted in green; incorrect predictions in red.

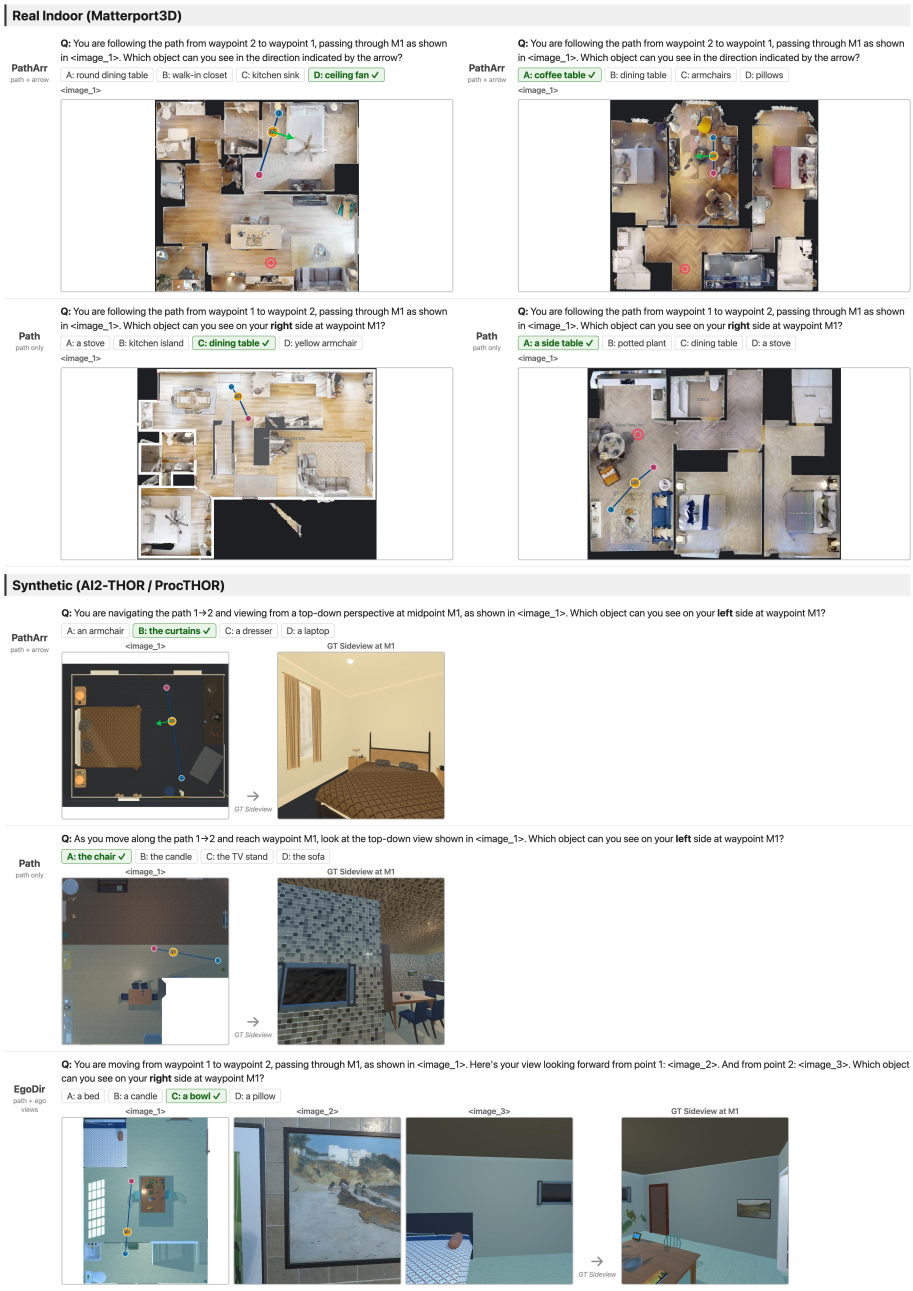

Fig. 5: Path tracing dataset examples. Top: real-world examples from Matterport3D in the PathArr and Path settings. Bottom: synthetic examples from AI2-THOR/ProcTHOR in the PathArr, Path, and EgoDir settings, with ground-truth sideviews at midpoint M_1 shown on the right. Each example shows the input image(s), question, and four answer choices with the correct answer highlighted.

F.2 Perspective Taking

Inference with imaginative perception. Figure 6 shows examples of perspective taking inference with imaginative perception tokens. The model receives a first-person view with an "X" mark indicating the target position, generates an imagined novel viewpoint as a visual thought, and predicts whether an object is closer/further or on the left/right. As with path tracing, the generated visual thoughts are not pixel-perfect but capture the essential spatial layout. The first two rows show correct predictions where the model successfully imagines the scene from the new viewpoint. The third row shows a failure case where the model incorrectly predicts the relative position.

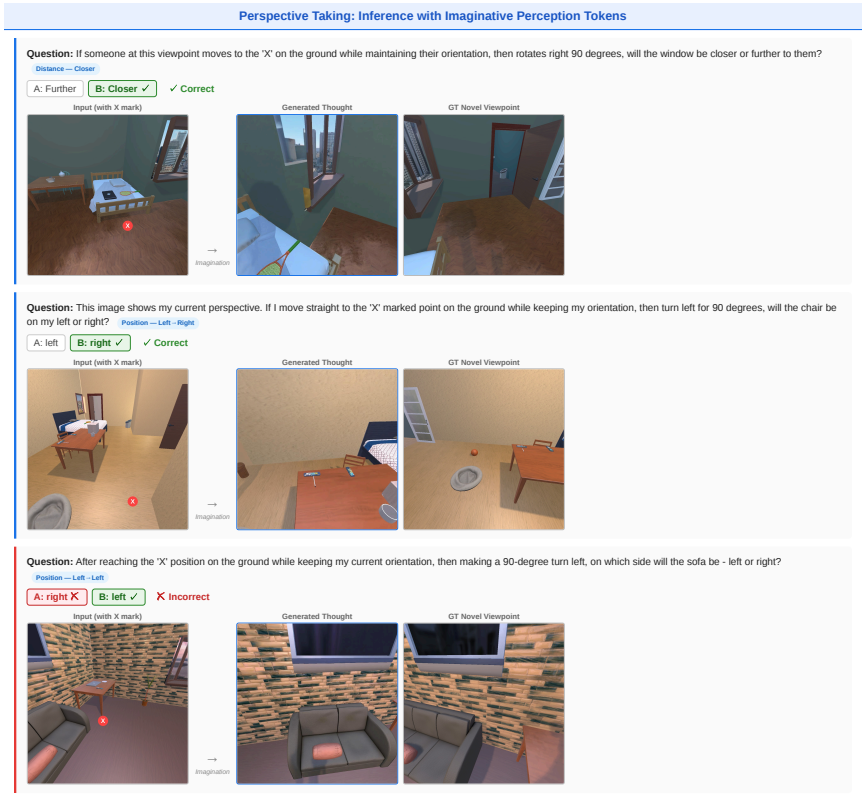

Fig. 6: Perspective taking with imaginative perception tokens. The model receives an input view with an "X" mark on the ground, imagines the scene from the target viewpoint, and predicts spatial relationships. Generated thoughts are compared against ground-truth novel viewpoints. Correct answers are highlighted in green; incorrect predictions in red.

Dataset examples. Figure 7 shows representative examples from the AI2-THOR/ProcTHOR perspective taking dataset across all four sub-categories: distance change (closer/-further) and relative position (left→right / right→left). Each example shows the input image with the target position marked by "X" and the ground-truth novel viewpoint after moving to "X" and turning.

F.3 Multiview Counting

Inference with imaginative perception. Figure 8 shows examples of multiview counting inference with imaginative perception tokens. The model receives four egocentric views from a rotation or multi-camera trajectory, generates a top-down BEV map as its visual thought, and counts the target objects. The generated top-down maps show that the model learns to synthesize a bird's-eye view from multiple perspectives, capturing the approximate room layout and object placements. The first two rows show correct predictions; the third row shows a failure case.

Dataset examples. Figure 9 shows representative examples from the AI2-THOR/ProcTHOR multiview counting dataset for both trajectory types: rotation (four cardinal directions from a fixed position) and multi-camera (four cameras placed at the corners of a square path). Each example shows the four input views and the ground-truth top-down map with agent/camera positions annotated.

G Imaginative Token Exploration with Different VLMs

Prior to adopting unified models like BAGEL for imaginative perception token generation, we investigated adding discrete imaginative perception tokens directly to the language model vocabulary of state-of-the-art vision-language models. Inspired by Aurora [2], we first trained a VQ-VAE from scratch on the intermediate RGB images in our datasets, including novel viewpoint renders, top-down BEV maps, and sideview images. However, the reconstruction quality of these simple VQ-VAEs was insufficient for supervising models on their intermediate token sequences as image outputs.

We therefore switched to off-the-shelf pretrained VQ-GANs [14] with varying configurations. These configurations differ along two axes: codebook size (e.g., 1K, 8K, and 16K entries) and spatial downsampling ratio ($f = 8$ vs. $f = 16$). Each choice involves a tradeoff: a larger codebook improves representational fidelity but inflates the model vocabulary, while a smaller downsampling ratio yields higher reconstruction quality at the cost of a longer token sequence per image, increasing context length. Figure 10 illustrates reconstruction quality across these settings.

We selected Qwen2.5-VL [36] in two sizes (3B and 7B) as our backbone for discrete token finetuning experiments. We note that the training data at this stage was of lower quality than our final datasets described in main text; these

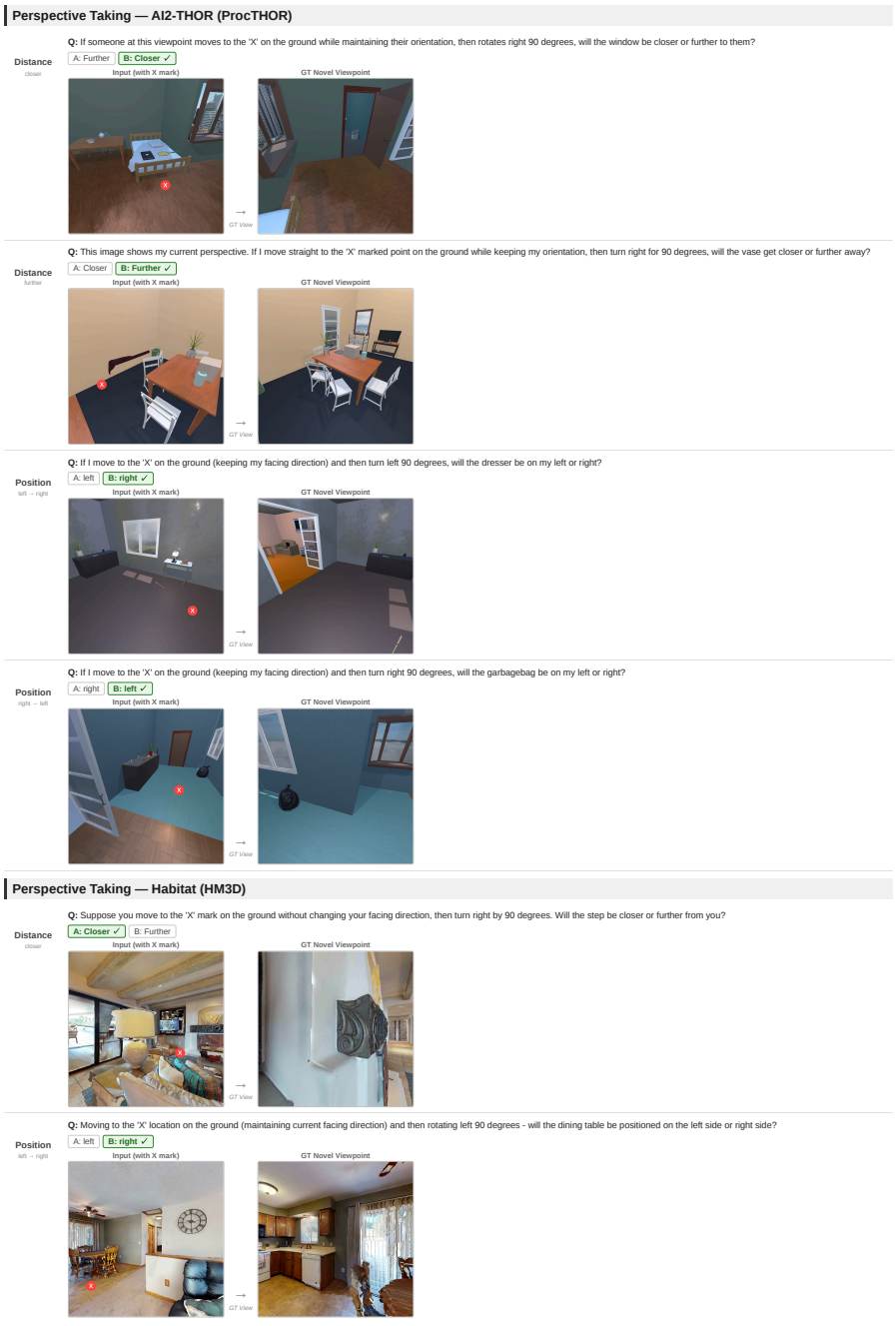

Fig. 7: Perspective taking dataset examples. Examples from AI2-THOR/ProcTHOR across the four sub-categories. Each example shows the input view with "X" mark and the ground-truth novel viewpoint. The correct answer is highlighted in green.

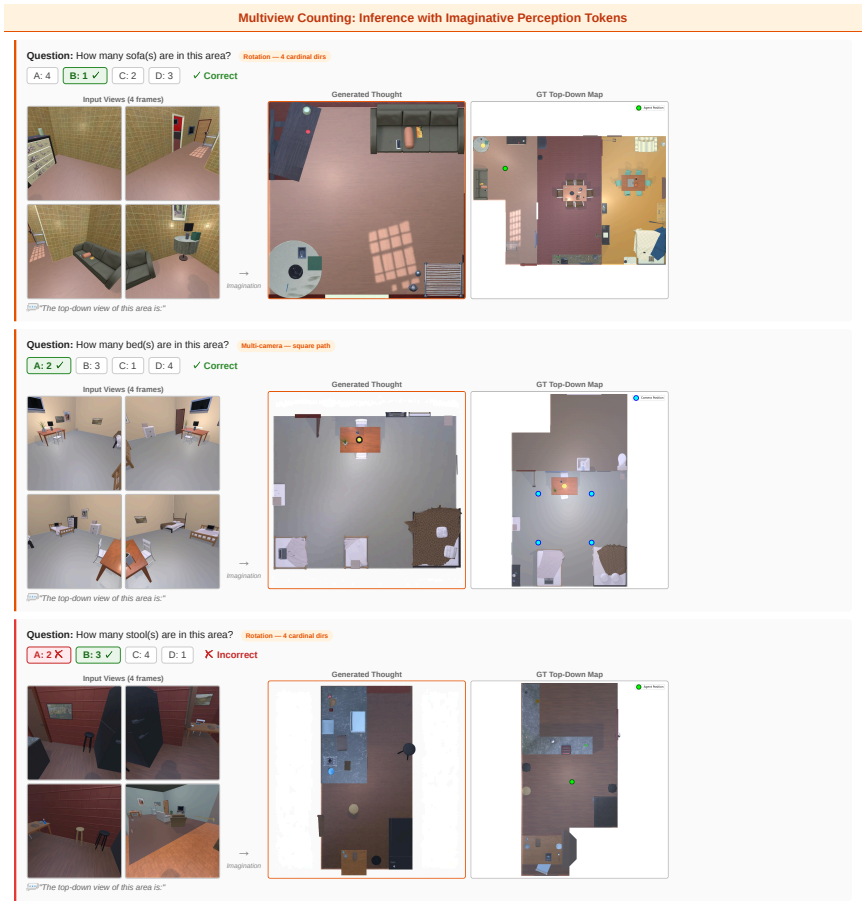

Fig. 8: Multiview counting with imaginative perception tokens. The model receives four egocentric views, imagines a top-down BEV map, and counts target objects. Generated thoughts are compared against ground-truth top-down maps. Correct answers are highlighted in green; incorrect predictions in red.

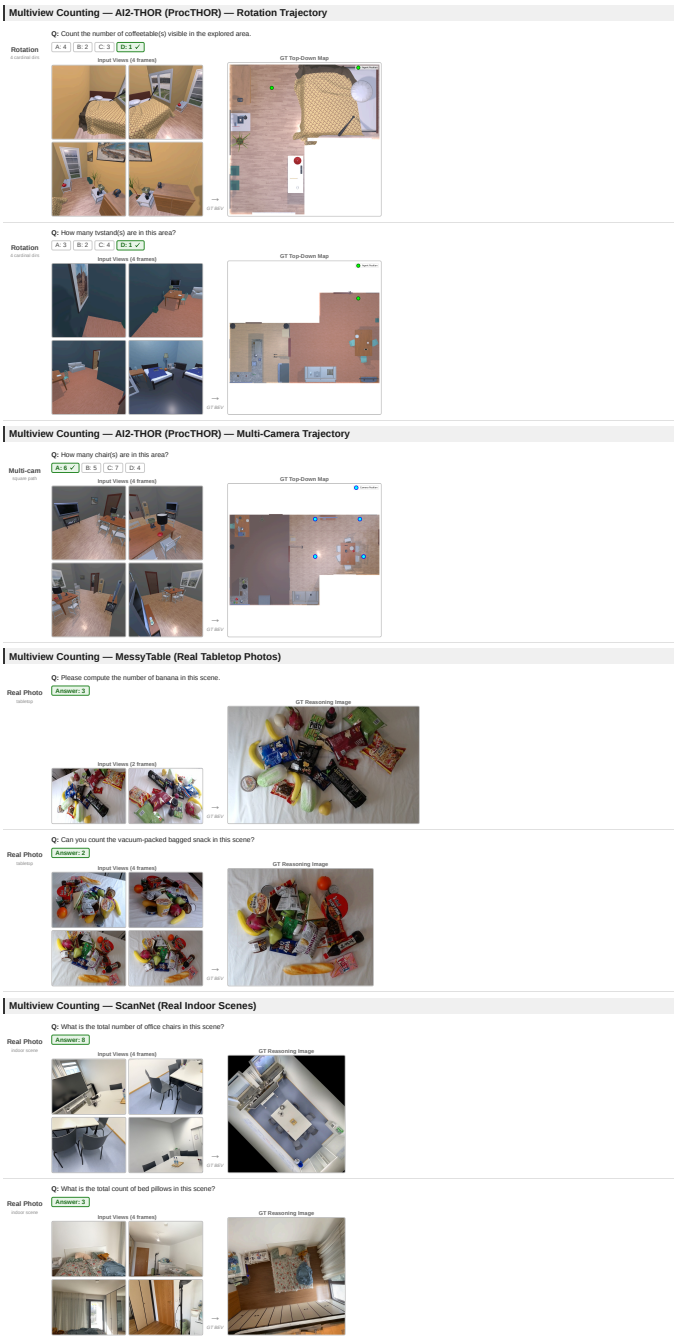

Fig. 9: Multiview counting dataset examples. Examples from AI2-THOR/ProcTHOR showing both rotation (top) and multi-camera (bottom) trajectories. Each example shows four input views in a 2×2 grid and the ground-truth top-down map. The correct answer is highlighted in green.

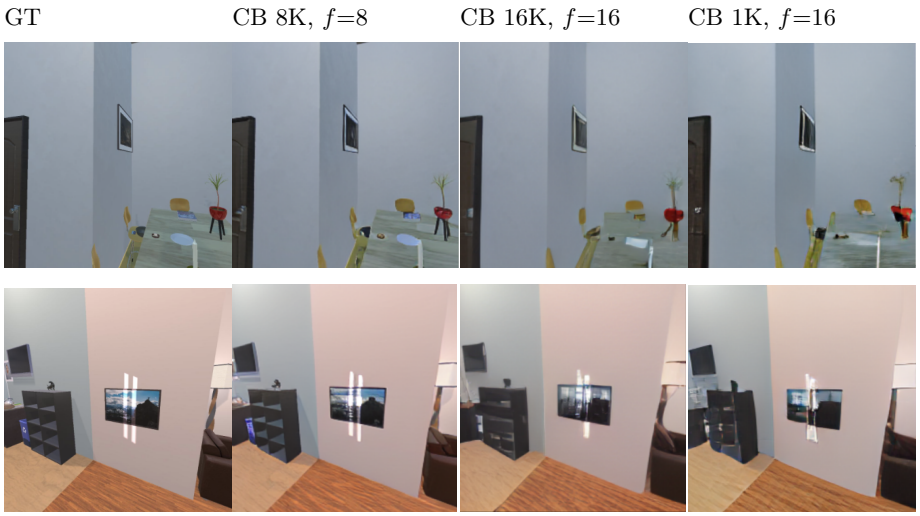

Fig. 10: VQGAN reconstruction quality across codebook and downsampling settings. Larger codebooks and smaller f improve fidelity but increase vocabulary size and sequence length respectively.

experiments were intended solely to probe whether discrete imaginative perception tokens can serve as a useful intermediate, not to achieve peak performance. We focused on Path Tracing (PT) and Perspective Taking (PET).

For each model we trained three variants: answer-only finetuning, Text CoT, and image chain-of-thought with discrete IPTs, where the VQGAN codebook tokens are appended to the model vocabulary and the model first autoregressively generates the imaginative perception token sequence before predicting the final answer plus a zeroshot baseline. For IPT variants we tested two VQGAN settings that keep sequence length manageable: CB 16K f=16 and CB 1K f=16. Results are shown in Table 13.

IPT consistently outperforms both answer-only finetuning and Text CoT on Path Tracing, with CB 1K f=16 yielding the best results for both model sizes (55.0 for 3B, 55.9 for 7B). On Perspective Taking, gains are modest and near the zero-shot baseline, suggesting that lower data quality and the representational limitations of discrete token reconstruction are a bottleneck for this more visually demanding task. A substantial gap remains relative to our final BAGEL-based results, motivating the move to a unified model.

Figure 11 shows ground-truth imagination images alongside the corresponding IPTs decoded from the Qwen2.5-VL 3B model. The decoded outputs are visually degraded, lacking the spatial structure and object detail present in the ground truth, which helps explain the remaining performance gap and further motivated our switch to continuous latent representations.

We further investigated alternative intermediate image representations. Instead of training the model to generate imaginative perception tokens for RGB

Table 13: Discrete IPT experiments on Qwen2.5-VL. Accuracy (%) on Path Tracing (PT) and Perspective Taking (PET). Training data at this stage was of lower quality than the final datasets. "–" denotes experiments not conducted.

Model	Method	PT	PET
Qwen2.5-VL 3B	Zero-shot	33.0	50.0
	Answer-only	48.6	50.5
	Text CoT	43.0	–
	IPT (CB 16K, f=16)	50.5	48.5
	IPT (CB 1K, f=16)	**55.0**	50.0
Qwen2.5-VL 7B	Zero-shot	38.5	47.2
	Answer-only	37.6	–
	Text CoT	35.7	–
	IPT (CB 16K, f=16)	55.0	–
	IPT (CB 1K, f=16)	**55.9**	–

Fig. 11: Ground-truth vs. decoded IPTs from Qwen2.5-VL 3B. The model-generated imagination tokens decode into visually degraded images that fail to preserve the spatial structure of the ground truth, highlighting the limitations of discrete token generation in non-unified VLMs.

thought images, we replaced them with tokens for grayscale images and tokens for pseudo depth maps obtained from the DepthAnything model [41]. The intuition is that simplifying the generation target, from full RGB to grayscale, reduces the difficulty of the token prediction task and may improve spatial reasoning downstream. Results in Table 14 show that switching from RGB to grayscale does boost performance (55.0→59.6 on PT, 50.0→55.5 on PET), while depth tokens perform comparably to RGB. Nevertheless, a substantial gap remains, and Figure 12 shows that the decoded grayscale outputs are still visually degraded, indicating that generation quality rather than representation type is the primary bottleneck.

Figure 12 shows ground-truth imagination images alongside the corresponding IPTs decoded from the Qwen2.5-VL 3B model. The decoded outputs are

Table 14: Effect of intermediate image representation on Qwen2.5-VL 3B. Accuracy (%) on Path Tracing (PT) and Perspective Taking (PET).

Method	**PT**	**PET**
IPT RGB (CB 1K, f=16)	55.0	50.0
IPT Grayscale (CB 1K, f=16)	**59.6**	**55.5**
IPT Depth (Aurora VQVAE)	55.0	54.7

visually degraded, lacking the spatial structure and object detail present in the ground truth, which helps explain the remaining performance gap and further motivated our switch to continuous latent representations.

These findings collectively motivated us to move away from discrete token generation in non-unified VLMs and instead adopt a unified model: BAGEL, that natively supports interleaved image understanding and generation through continuous latent representations.

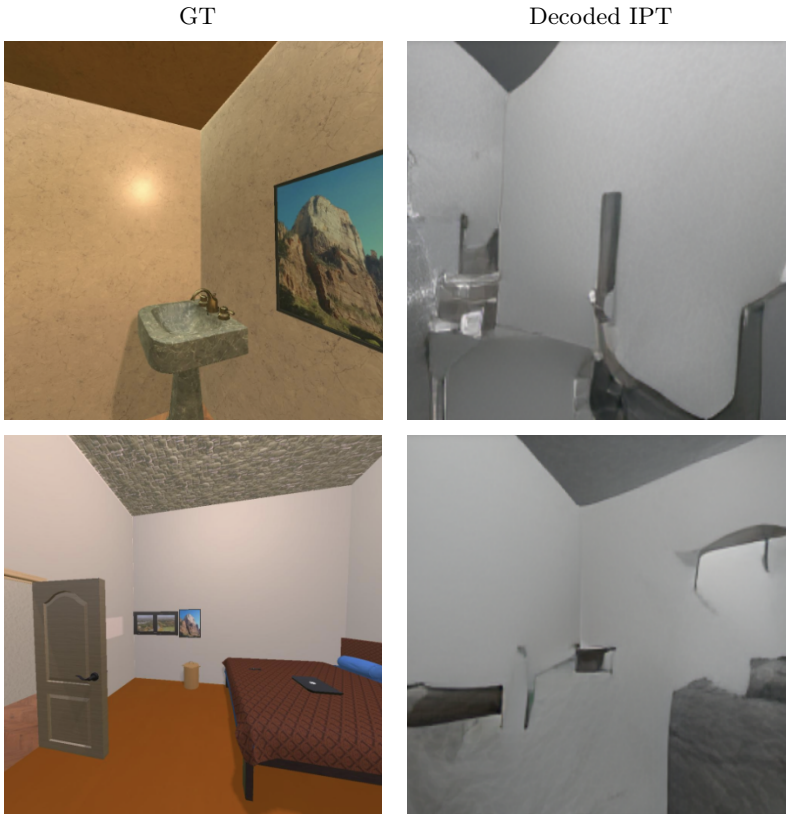

Fig. 12: Decoded IPTs from Qwen2.5-VL 3B for grayscale representations. Despite the simpler generation target, grayscale decoded outputs remain visually degraded and fail to preserve spatial structure.