REASON THE SCENE, RECONSTRUCT THE SPACE, RENDER THE UNSEEN: TINKERCUBE FOR SPATIAL INTELLIGENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Spatial reasoning, 3D reconstruction, and novel-view synthesis describe complementary aspects of the physical world, yet are typically isolated, limiting exchange between semantic and geometric evidence. We present TinkerCube, a coordinated multimodal model connecting spatial understanding, target-view geometry prediction, and controllable visual generation. TinkerCube augments a Mixture-of-Transformer-Experts backbone with a multi-view geometry pathway and role-specific interfaces that align 2D appearance, camera states, and cross-view structure in a shared context. Geometry-aware reasoning supplies explicit spatial evidence to the language expert; language-conditioned geometry converts motion and content instructions into target cameras, depths, and point maps; render-guided generation projects a reference-only point cloud into the target view to constrain completion. Three-stage training progressively aligns spatial tokens with language, learns motion-conditioned target geometry, and adapts coverage-aware view synthesis while preserving earlier capabilities. Across spatial question answering, depth prediction and camera control, point-map reconstruction, and novel-view synthesis, TinkerCube delivers consistent performance, with leading results on major spatial-reasoning benchmarks and strong view-synthesis performance on DL3DV and RealEstate10K. These results show that integrating semantic reasoning, 3D geometry, and visual generation through joint training and interaction effectively strengthens spatial intelligence. Code is available at: `https://anonymous.4open.science/r/TinkerCube`.

1 INTRODUCTION

Spatial intelligence is emerging as an objective of multimodal learning. Spatial-perception benchmarks and spatially supervised vision–language models ask systems to infer distances, directions, correspondences, and cross-view relations from images rather than merely recognize content (Chen et al., 2024a; Cai et al., 2025; Ji et al., 2025). In parallel, geometry-centric models reconstruct depth, cameras, and point-based scene structure from reference views (Wang et al., 2026b; 2025a), while novel-view synthesis (NVS) renders how the scene should appear under a new camera (Mildenhall et al., 2020; Kerbl et al., 2023). These lines of work emphasize complementary facets of the same underlying 3D world: spatial reasoning interprets its organization, whereas reconstruction and view synthesis recover and visualize that organization. The growing ability of multimodal models to accommodate both understanding and generation suggests a basis for making these capabilities mutually useful (Deng et al., 2025; Zhao et al., 2025).

Despite this progress, three gaps prevent existing systems from fully exploiting that complementarity. **Geometry-light reasoning.** Most spatial VLMs encode images with a generic 2D visual backbone and resolve questions primarily in a language-centric representation space; neither component explicitly preserves 3D structure or cross-view camera geometry, leaving projection-induced ambiguities to be recovered indirectly. **Rigid scene control.** Reconstruction pipelines typically derive static scene estimates from RGB observations and offer limited support for scene expansion; NVS systems additionally depend on precisely specified numerical camera trajectories, limiting semantic control over camera motion and target content. **Weak task coupling.** Existing integrations often introduce geometry as an auxiliary cue for one downstream task, or expose generation as an

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

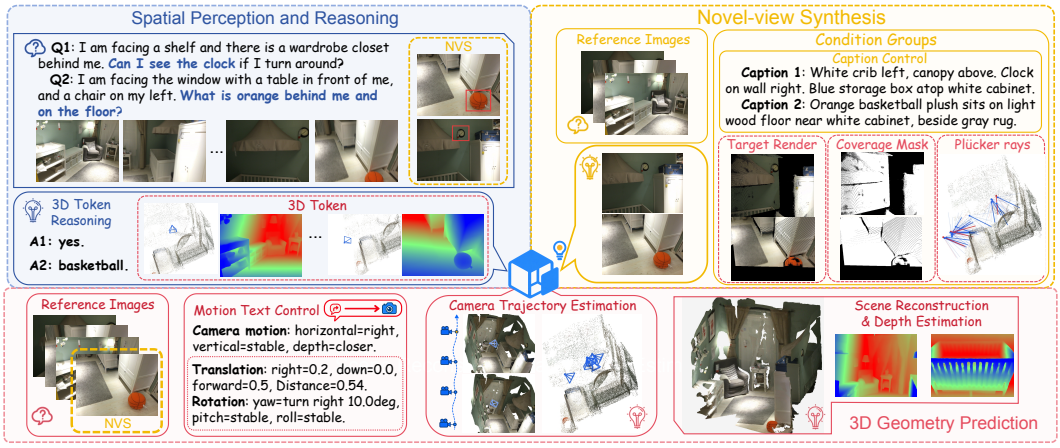


Figure 1: **TinkerCube's spatial capabilities.** Visual and 3D token fusion enables spatial question answering; language-conditioned geometry supports camera control, depth estimation, and point-cloud reconstruction; render-guided generation synthesizes novel views using text and reference images. Generated views enrich references for spatial reasoning and iterative scene reconstruction.

isolated output, without systematically organizing how semantic reasoning, 3D scene structure, and 3D-to-2D rendering can support one another. These gaps motivate a training and inference framework built around shared spatial representations and explicit, reusable geometric intermediates.

These limitations reveal complementary roles for reasoning, reconstruction, and generation at the task and representation levels. Because they operate in the same physical space, global, cross-view-consistent geometry can ground spatial reasoning, while semantic reasoning can translate language-specified motion and content into camera and generation conditions. Aligning vision, geometry, and language in a shared context strengthens 3D-aware reasoning while enabling flexible semantic control without precise parameter inputs. Motivated by this complementarity, we introduce **TinkerCube**, a unified model for spatial understanding, 3D geometric modeling, and controllable novel-view generation. In Fig. 1, TinkerCube combines a Mixture-of-Transformer-Experts backbone for semantic reasoning and image generation with a spatial geometry module that extracts cross-view camera, depth, and scene-structure features, projecting 2D visual features and 3D spatial tokens into a shared representation space. In spatial VQA, the reasoning expert jointly analyzes appearance, camera relations, and multi-view geometry. For 3D scene modeling, language representations of motion and target content interact with reference-scene features through learnable target tokens to predict the target camera, depth, and point map, from which an explicit point cloud can be constructed. For NVS, a predicted pose or an externally supplied trajectory defines the target view; the reference point cloud is rendered into that view, after which the generation expert, jointly conditioned on the rendered result, coverage mask, Plücker rays, text, and reference appearance, performs generative rendering of the target scene image. A three-stage optimization progressively aligns geometric tokens with multimodal reasoning, learns language-conditioned target geometry, and adapts coverage-aware view generation. This design establishes a directed form of cooperation rather than merely placing multiple task modules side by side: spatial representations ground reasoning, semantic states condition geometric prediction, and reconstructed geometry constrains visual generation.

We evaluate TinkerCube across spatial question answering, depth prediction and camera control, point-map reconstruction, and NVS against general-purpose multimodal models and task-specific systems. It achieves leading results on major spatial-reasoning and NVS evaluations while remaining highly competitive across geometric reconstruction tasks. These results show that a single coordinated system can sustain strong reasoning, reconstruction, and generation performance rather than improving one task at the expense of others. Our work makes the following contributions:

- **Unified spatial framework.** TinkerCube integrates spatial understanding, 3D scene modeling, and controllable novel-view generation through coordinated multimodal interactions.
- **3D-informed multimodal reasoning.** Scene structure and camera information strengthen vision–language reasoning about spatial relations, multi-view evidence, and 3D layouts.
- **Language-directed generation.** Natural-language motion and content instructions jointly guide geometric prediction, viewpoint transformation, and novel-view generation.

2 Related Work

Spatial Reasoning and Geometry Awareness. Spatial reasoning research strengthens vision–language models along several complementary directions. Data-centric methods construct spatial instruction corpora and use staged supervision or reinforcement learning to progress from localization to relational inference (Li et al., 2026a; Wu et al., 2025a; Zhang et al., 2025a). Geometry-aware models introduce structural features, camera tokens, or reconstruction-aligned objectives (Wu et al., 2025a; Fan et al., 2026). More recent spatial systems couple understanding with reconstruction or generation: geometry-grounded VLMs align 3D predictions with linguistic outputs (Fan et al., 2026; Hu et al., 2026b), while generative reasoners use drawings, imagined views, or native multimodal generation to preserve spatial states (Wu et al., 2025b; Li et al., 2026b; Quan et al., 2026). However, these approaches largely treat spatial question answering as the endpoint or optimize only one direction of transfer; the inferred structure is rarely exposed as a reusable interface for camera, depth, point-map prediction, and view synthesis.

3D Reconstruction and View Generation. 3D scene modeling spans monocular geometry, multi-view reconstruction, and generative view synthesis. Generalist geometry models recover metric depth or point maps from single or variable-view inputs (Wang et al., 2025c; Lin et al., 2026), whereas feed-forward multi-view transformers jointly estimate cameras, depth, and dense scene structure (Wang et al., 2024a; 2025a). NVS methods condition image or video generators on rendered point clouds, persistent 3D caches, and prescribed camera trajectories, combining geometric projection with learned completion (Yu et al., 2024; Ren et al., 2025). Despite these advances, reconstruction and generation systems remain largely dependent on visual observations and numerical camera inputs, with limited semantic reasoning over desired motion and content. This leaves language instructions insufficiently connected to target-camera control, dense reconstruction, and controllable NVS.

3 Method of TinkerCube

TinkerCube supports spatial reasoning, target-view geometry prediction, and novel-view synthesis. It comprises three functional modules. The ***Geometry-Aware Reasoner*** **(GAR)** aligns 2D appearance features with 3D spatial tokens. ***Language-Conditioned Geometry*** **(LCG)** realizes a motion instruction m_t as a target camera and predicts dense geometry conditioned on reference observations and a target description c_t. The ***Render-Guided Generator*** **(RGG)** projects reference geometry into the target view and renders the scene image under semantic and appearance conditions. The modules collaborate through language hidden states, camera parameters, and point clouds.

3.1 Model Architecture

As shown in Fig. 2, TinkerCube adopts a Mixture-of-Transformer-Experts (MoT) multimodal backbone. Understanding and generation tokens share the attention topology and contextual sequence. The understanding expert $\mathcal{U}$ jointly reasons over text, 2D visual tokens, and spatial tokens to generate text, while the generation expert $\mathcal{G}$ predicts a continuous velocity field in image latent space. The backbone further contains a 2D visual encoder $\mathcal{F}_v$ and an image latent encoder–decoder pair $(\mathcal{E}, \mathcal{D})$. For spatial tasks, we introduce a multi-view spatial encoder $\mathcal{F}_s$ and a point-cloud renderer $\mathcal{R}$. $\mathcal{F}_s$ uses 16×16 patch embeddings and prepends one camera token and 16 register tokens to the patch tokens of each view: $\mathbf{X}_i^0 = [\mathbf{x}_i^{\text{cam}}; \mathbf{X}_i^{\text{reg}}; \mathbf{X}_i^{\text{patch}}]$. A multi-layer Transformer aggregates information across views. A camera head predicts $\mathbf{p}_i = (\mathbf{t}_i, \mathbf{q}_i, \mathbf{f}_i) \in \mathbb{R}^9$, where $\mathbf{t}_i$, $\mathbf{q}_i$, and $\mathbf{f}_i = (f_i^h, f_i^w)$ encode translation, quaternion rotation, and vertical/horizontal fields of view; a dense head estimates depth and confidence from patch features. $\mathcal{F}_s$ thereby provides camera states, cross-view memory, and dense structure, enabling multimodal fusion and interactive reasoning through a common token sequence and geometric intermediates.

Geometry-Aware Reasoner. GAR augments visual-language reasoning with cross-view-consistent 3D evidence. For a reference image I_i, the 2D path extracts patch features with $\mathcal{F}_v$ and maps them to the backbone using projector C_v. In parallel, $\mathcal{F}_s$ jointly encodes all views of the same scene:

$$\mathbf{V}_i = C_v(\mathcal{F}_v(I_i)) \in \mathbb{R}^{P_i \times d}, \quad \{\mathbf{g}_i^{\text{sc}}, \mathbf{g}_i^{\text{cam}}, \mathbf{G}_i^{\text{reg}}\}_{i=1}^{N} = \mathcal{F}_s(I_1, \ldots, I_N)\,. \tag{1}$$

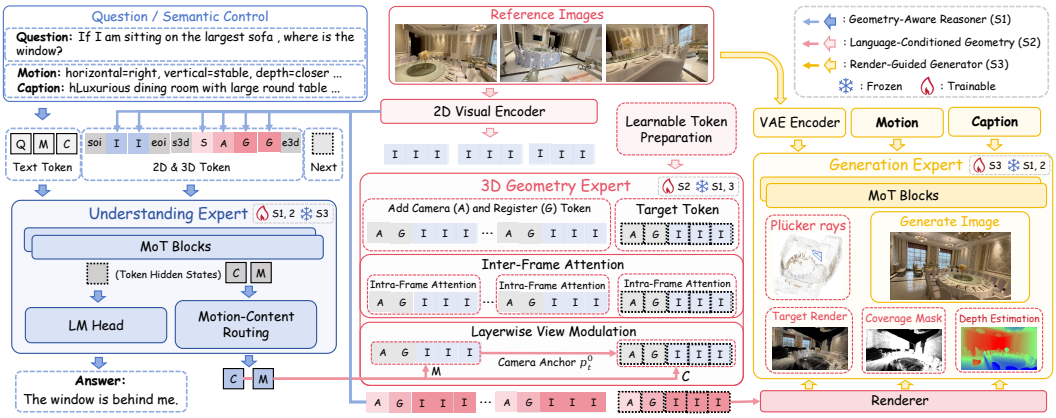


Figure 2: **TinkerCube architecture.** GAR (S1) combines spatial tokens from the geometry expert with visual tokens for spatial question answering by the understanding expert. LCG (S2) routes motion and caption hidden states into the geometry expert, where camera anchoring and layerwise modulation guide interactions between reference and learnable target tokens to predict cameras and depth; back-projection yields point maps. RGG (S3) renders reference geometry at the predicted viewpoint, then conditions the generation expert on rendered views, coverage masks, camera rays, reference appearance, and semantic context to synthesize novel views.

Here, P_i is the number of 2D patches. $\mathbf{V}_i$ preserves object identity, texture, and local 2D position; $\mathbf{g}_i^{\text{cam}}$ represents the global camera state; $\mathbf{G}_i^{\text{reg}}$ contains 16 register tokens after cross-view interaction; and $\mathbf{g}_i^{\text{sc}}$ is a scene-level summary. Together, these features encode both per-view content and shared multi-view structure.

We introduce *Role-Aware Spatial Projection* to inject scene, camera, and register representations into the reasoning space: $\mathbf{S}_i = C_{\text{sc}}(\mathbf{g}_i^{\text{sc}}), \mathbf{A}_i = C_{\text{cam}}(\mathbf{g}_i^{\text{cam}}), \bar{\mathbf{G}}_i = C_{\text{reg}}(\mathbf{G}_i^{\text{reg}})$. The three projectors map scene, camera, and register representations into the reasoning space while preserving their distinct roles. Their outputs form a *Spatial View Block*, which places the 2D appearance and 3D evidence of the same image in adjacent but distinguishable token segments:

$$\mathbf{B}_i = [\, e_{\text{soi}}; \mathbf{V}_i; e_{\text{eoi}}; e_{\text{s3d}}; \mathbf{S}_i; \mathbf{A}_i; \bar{\mathbf{G}}_i; e_{\text{e3d}} \,]. \tag{2}$$

$e_{\text{soi}}/e_{\text{eoi}}$ and $e_{\text{s3d}}/e_{\text{e3d}}$ mark the boundaries of the 2D image and spatial segments, respectively. Boundary, patch, and spatial tokens from the same view share its 2D positional encoding, while block order retains the view order. Given textual context x, the understanding expert produces

$$\mathbf{H}^u = \mathcal{U}([\mathbf{B}_1; \ldots; \mathbf{B}_N; x]), \tag{3}$$

where $\mathbf{H}^u$ denotes the unified contextual states. This design lets 2D tokens identify scene content, while camera and register tokens provide structural evidence for cross-view correspondence, enabling question-adaptive interaction among object semantics, camera relations, and multi-view structure.

Language-Conditioned Geometry. Given ordered reference views $\mathcal{I}_r = \{I_i\}_{i=1}^N$, motion text m_t, and target description c_t, LCG interprets the intended viewpoint change and uses reference-scene context to resolve unspecified viewing degrees of freedom. It then realizes the target camera and jointly predicts target-view depth and a point map. To learn and predict this frame, we introduce *Learnable Target-View Tokens* in $\mathcal{F}_s$: $\mathbf{T}_t^0 = [\mathbf{q}_t^{\text{cam}}; \mathbf{Q}_t^{\text{reg}}; \mathbf{Q}_t^{\text{patch}}]$. The sequence contains one target camera token, 16 target register tokens, and target patch tokens on a regular 2D grid. The camera and register tokens respectively capture viewpoint and global information, while the patch tokens are interpolated to different resolutions. Together, they learn a target representation from the reference structure and language conditions.

Let $\mathcal{M}$ and $\mathcal{C}$ index the motion and caption spans in the unified sequence. *Motion–Content Routing* extracts two conditions from $\mathbf{H}^u$:

$$\mathbf{Q}_m = \text{Pool}_K(\mathbf{H}^u[\mathcal{M}]), \quad \mathbf{Q}_c = C_c(\text{Pool}_K(\mathbf{H}^u[\mathcal{C}])), \tag{4}$$

where $K = 32$ is the fixed number of condition tokens and C_c is the content projector. $\mathbf{Q}_m$ encodes translation, rotation, and field-of-view changes, whereas $\mathbf{Q}_c$ describes the target content. The camera, register, and pooled patch tokens of the reference views form reference memory $\mathbf{M}_r$. A motion projector C_m maps $\mathbf{Q}_m$ to the pose space and combines it with $\mathbf{M}_r$ to predict a base camera anchor:

$$\mathbf{p}_t^0 = \mathcal{P}_{\text{anchor}}(C_m(\mathbf{Q}_m), \mathbf{M}_r), \quad \mathbf{p} = (\mathbf{t}, \mathbf{q}, \mathbf{f}) \in \mathbb{R}^9, \quad \mathbf{f} = (f^h, f^w). \tag{5}$$

$\mathcal{P}_{\text{anchor}}$ is a three-layer pose decoder that cross-attends the motion queries to reference memory, providing a preliminary parse of the target camera pose. The resulting anchor specifies an initial position, orientation, and field of view, and is converted to camera center $\mathbf{o}_t^0$, rotation matrix $\mathbf{R}_t^0$, and field of view $\mathbf{f}_t^0$. A *Pose–Ray Encoder* maps these quantities to global pose tokens $\mathbf{Q}_p$ and patch-wise Plücker ray tokens $\mathbf{Q}_\rho$:

$$\mathbf{Q}_p = C_p\big(\text{Fourier}([\mathbf{o}_t^0; \text{vec}(\mathbf{R}_t^0); \log \mathbf{f}_t^0])\big), \quad \mathbf{Q}_{\rho,j} = C_\rho^s\big(\text{Fourier}([\mathbf{d}_j; \mathbf{o}_t^0 \times \mathbf{d}_j])\big). \tag{6}$$

Here, $\mathbf{d}_j$ is the ray direction of target patch j, $\mathbf{Q}_p$ constrains the global viewpoint, $\mathbf{Q}_{\rho,j}$ represents the corresponding local viewing direction, and $\mathbf{Q}_\rho = \{\mathbf{Q}_{\rho,j}\}_j$. The spatial encoder is a multi-layer Transformer. Across layers, the learnable target tokens first absorb reference structure from observed reference tokens through within-view and cross-view attention. At selected layers, we apply *Layerwise View Modulation*, whose cross-attention injects motion, caption, pose, and ray conditions into the learnable target tokens across successive depths:

$$\mathbf{T}_t = \Phi_{\text{target}}\big(\mathbf{T}_t^0, \mathcal{I}_r \mid \mathbf{Q}_c, \mathbf{Q}_p, \mathbf{Q}_\rho\big). \tag{7}$$

The target camera and register tokens emphasize $\mathbf{Q}_p$, while target patch token j integrates $\mathbf{Q}_c$, $\mathbf{Q}_p$, and $\mathbf{Q}_{\rho,j}$. Repeated injection provides fine-grained modulation of target content, global viewpoint, and local viewing direction, updating $\mathbf{T}_t^0$ into $\mathbf{T}_t$ to represent the target observation pose and its spatial structure. Finally, a *Target Structure Decoder* lets the learnable target tokens further retrieve reference and condition memory. Its camera branch predicts bounded residuals in camera-center, axis–angle, and log-field-of-view spaces to refine the base anchor. Its dense branch fuses multi-level target patch tokens with $\mathbf{Q}_c$ to estimate depth and uses reprojection to the references to correct local structure. The target point map is derived from the final camera and depth rather than an independent prediction head:

$$\hat{\mathbf{p}}_t = \mathbf{p}_t^0 \oplus \mathcal{H}_{\text{cam}}(\mathbf{T}_t, \mathbf{M}_r), \quad \hat{D}_t = \mathcal{H}_{\text{depth}}(\mathbf{T}_t, \mathbf{M}_r, \hat{\mathbf{p}}_t, \mathbf{Q}_c), \quad \hat{P}_t(u) = \pi_{\hat{\mathbf{p}}_t}^{-1}\Big(u, \hat{D}_t(u)\Big). \tag{8}$$

Here, u denotes a target pixel, $\oplus$ is a residual update in camera-parameter space, and $\pi_{\mathbf{p}}^{-1}$ denotes back-projection under camera $\mathbf{p}$. Motion provides the initial camera cue, the caption supplies content semantics, and the layerwise-refined target tokens are decoded into a coordinate-consistent camera, depth map, and point map.

Render-Guided Generator. RGG uses geometric projection under predicted poses to provide a spatial scaffold for novel-view generation. It constructs a colored point cloud $\mathcal{P}_r$ from the reference images and their predicted cameras and depths, and renders it into the target view:

$$\mathcal{P}_r = \text{Fuse}_{i=1}^N \, \pi_{\mathbf{p}_i}^{-1}(I_i, D_i), \qquad (\widetilde{I}_t, M_t) = \mathcal{R}(\mathcal{P}_r, \hat{\mathbf{p}}_t). \tag{9}$$

$\pi_{\mathbf{p}_i}^{-1}(I_i, D_i)$ denotes colored back-projection under camera $\mathbf{p}_i$. The rendered image $\widetilde{I}_t$ carries the reference-supported target structure, while coverage map M_t marks valid projections. The point cloud is also re-rendered into selected reference cameras to form a *Reference Render Cache* $(I_i, \widetilde{I}_i, M_i)$, which captures the correspondence between real appearance and rendering degradation. RGG then organizes its controls into semantic, visual, and spatial condition groups:

$$\mathcal{C}_t = \{\mathcal{C}_{\text{sem}}, \mathcal{C}_{\text{vis}}, \mathcal{C}_{\text{spatial}}\}. \tag{10}$$

$\mathcal{C}_{\text{sem}}$ is the GAR context of the references, motion, and caption. $\mathcal{C}_{\text{vis}}$ uses a frozen VAE to encode the real references, reference re-renderings, and target rendering as *Dual-Latent Conditioning*. $\mathcal{C}_{\text{spatial}}$ contains reference and target Plücker rays together with *Local Coverage Encoding*, which summarizes each coverage map into token-aligned support codes. These conditions respectively preserve scene semantics and appearance, specify viewing directions, and distinguish reliable projections from regions requiring generative rendering. Conditioned on $\mathcal{C}_t$, the generation expert predicts the latent flow from initial noise $\mathbf{z}_1$, and the VAE decoder produces the target image: $\hat{I}_t = \mathcal{D}(\text{Flow}_{\mathcal{G}}(\mathbf{z}_1 \mid \mathcal{C}_t))$. The rendered view anchors observable geometry, while the semantic, appearance, ray, and coverage controls guide artifact correction and scene rendering beyond the projected support. Task-specific inference reuses these pathways for reasoning, geometry, and NVS (Appendix B).

3.2 TRAINING PIPELINE

TinkerCube is trained in three progressive stages. Stage 1 aligns spatial tokens with the language reasoning space. Stage 2 preserves the learned understanding capability while learning the target camera and dense structure. Stage 3 trains novel-view generation from reference point-cloud renderings. Across stages, spatial tokens, camera estimates, and point clouds carry forward capabilities learned earlier.

Stage 1: Spatial–Language Alignment. Stage 1 uses spatial visual-question-answering data in single-image, multi-image, and video formats, covering object attributes, spatial relations, referring expressions, camera motion, depth relations, and cross-view reasoning. Each sample is represented as $\mathcal{X}^{(1)} = (\{I_i\}_{i=1}^{N}, x^{\text{human}}, y^{\text{assistant}})$, where x^{human} and $y^{\text{assistant}}$ are the question and supervised answer. Each image yields both 2D visual tokens and spatial tokens. GAR places these tokens and the question in the language context, after which the understanding expert generates the answer using camera relations, depth cues, and cross-view structure. Autoregressive cross-entropy is applied to answer tokens:

$$\mathcal{L}_{\text{S1}} = -\sum_{j} \log p_{\mathcal{U}}\big(y_j \mid y_{<j}, \{I_i\}, x^{\text{human}}\big)\,. \tag{11}$$

Here, j indexes answer positions. This stage jointly updates the understanding expert and role-aware spatial projectors, aligning spatial tokens with the reasoning space while adapting the language layers to geometric evidence.

Stage 2: Target-View Structure Learning. Stage 2 trains LCG to predict a target camera and dense structure from reference images, a motion description, and target semantics. We sample several reference frames and one target frame from a scene to form

$$\mathcal{X}^{(2)} = (\{I_i\}_{i=1}^{N}, m_t, c_t; I_t, \mathbf{p}_t^*, D_t^*). \tag{12}$$

The semicolon separates the conditioning inputs from the target frame and supervisory geometry. Given $\mathcal{I}_r = \{I_i\}_{i=1}^{N}$, m_t, and c_t, LCG predicts

$$(\hat{\mathbf{p}}_t, \hat{D}_t, \hat{P}_t) = \Phi_{\text{LCG}}(\mathcal{I}_r, m_t, c_t). \tag{13}$$

Supervision covers the camera, dense structure, and cross-view projection. $\mathcal{L}_{\text{pose}}$ constrains position, rotation, and field of view; $\mathcal{L}_{\text{struct}}$ constrains depth, point maps, and surface orientation, with $P_t^*(u) = \pi_{\mathbf{p}_t^*}^{-1}(u, D_t^*(u))$; and $\mathcal{L}_{\text{view}}$ imposes observational consistency through reprojection. The combined objective is

$$\mathcal{L}_{\text{S2}} = \lambda_{\text{pose}} \underbrace{\big[\ell_1(\hat{\mathbf{t}}_t, \mathbf{t}_t^*) + d_R(\hat{\mathbf{R}}_t, \mathbf{R}_t^*) + \ell_1(\hat{\mathbf{f}}_t, \mathbf{f}_t^*)\big]}_{\mathcal{L}_{\text{pose}}\ \text{(camera)}} + \lambda_{\text{view}} \underbrace{\sum_{i=1}^{N} \ell_{\text{proj}}(\hat{P}_t; D_i^*, \mathbf{p}_i^*)}_{\mathcal{L}_{\text{view}}\ \text{(cross-view)}} + \lambda_{\text{struct}} \underbrace{\big[\ell_1(\hat{D}_t, D_t^*) + \ell_1(\hat{P}_t, P_t^*) + \ell_{\text{normal}}(\hat{P}_t, P_t^*)\big]}_{\mathcal{L}_{\text{struct}}\ \text{(dense structure)}}. \tag{14}$$

Here, $\hat{\mathbf{t}}_t$, $\hat{\mathbf{R}}_t$, and $\hat{\mathbf{f}}_t$ denote predicted translation, rotation, and fields of view; d_R is the rotation distance, and λ controls each loss group's weight. D_i^* and $\mathbf{p}_i^*$ are reference depth and camera annotations. The ℓ_1 terms average valid residuals, the normal loss supervises local surface orientation, and ℓ_{proj} uses valid, in-frame, depth-consistent projections. Together, these terms couple camera accuracy, surface structure, and cross-view agreement. Appendix D.5 provides their detailed definitions. Optimization first adapts the understanding expert, target tokens, and conditioning interfaces, then includes the spatial encoder and prediction heads.

Stage 3: Coverage-Aware Novel-View Generation. Stage 3 uses ordered multi-view samples $\mathcal{X}^{(3)} = (\mathcal{I}_r, m_t, c_t; I_t)$, where the reference views, motion, and caption form the conditions, and I_t provides target supervision. LCG predicts the target pose used for rendering, rays, and coverage; predicted reference geometry supplies the point cloud. The semantic, visual, and spatial controls follow $\mathcal{C}_t$ in Equation 10. Training uses flow matching. Let $\mathbf{z}_0 = \mathcal{E}(I_t)$ be the target latent, and sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and time τ. The interpolated latent and target velocity are $\mathbf{z}_\tau = (1-\tau)\mathbf{z}_0 + \tau\boldsymbol{\epsilon}$, $\mathbf{v}^* = \boldsymbol{\epsilon} - \mathbf{z}_0$. The flow-matching objective supervises target generation tokens under $\mathcal{C}_t$:

$$\mathcal{L}_{\text{S3}} = \mathbb{E}\Big[\|\mathbf{v}_{\mathcal{G}}(\mathbf{z}_\tau \mid \mathcal{C}_t, \tau) - \mathbf{v}^*\|_2^2\Big]\,. \tag{15}$$

The expectation is taken over training samples, τ, and $\boldsymbol{\epsilon}$, encouraging the generation expert to follow the target latent trajectory under the rendering and multimodal controls.

Table 1: Evaluation of image- and video-based spatial reasoning capabilities. Results are reported as accuracy (%), with ViewSpatial further separated by relation type.

Model	BLINK ↑	SPAR ↑	ViewSpatial						VSI-Bench ↑
			Overall ↑	Cam. Rel. ↑	Cam.–Obj. ↑	Person–Obj. ↑	Person Rel. ↑	Person SSRD ↑	
General VLMs									
Cambrian-S	37.9	37.9	41.3	50.4	22.7	45.0	38.8	41.9	57.3
MiniCPM-V-4.5	54.1	36.0	39.0	–	–	–	–	–	34.1
Qwen2.5-VL-32B	**78.7**	33.1	36.9	57.7	31.5	56.9	39.1	28.8	37.0
Qwen3-VL-8B-Instruct	66.7	39.6	42.2	54.2	29.7	47.3	40.3	31.1	57.9
GPT-5.2	68.0	41.5	45.6	60.2	27.9	41.0	48.5	40.1	54.5
Gemini-3-Pro-Preview	76.0	43.1	50.4	61.9	31.6	41.1	**74.4**	38.9	52.5
Spatially Specialized Models									
MindCube	35.1	51.0	24.1	30.9	22.1	23.7	22.4	16.9	17.2
ViLaSR	51.4	35.0	35.7	46.8	25.3	39.1	32.7	26.6	44.6
Spatial-MLLM	40.5	35.1	43.6	35.0	23.4	40.4	40.4	34.6	48.4
SpaceR	49.6	37.6	37.3	43.2	28.9	37.5	34.1	30.2	45.6
SpatialLadder	43.0	34.4	44.2	46.2	25.7	56.2	31.9	33.6	44.9
VST	58.8	48.9	42.4	46.9	35.4	**70.3**	52.6	**62.8**	57.9
Multi-capability Models									
BAGEL	62.5	39.3	42.0	–	–	–	–	–	26.7
G^{2}VLM	23.5	41.7	16.9	2.1	30.2	27.3	31.5	8.0	–
UniUGG	44.4	47.2	–	–	–	–	–	–	40.1
VLM-3R	52.3	42.4	40.5	–	–	–	–	–	60.9
Omni-View	56.2	59.8	47.9	53.7	39.5	52.7	45.5	43.4	55.3
TinkerCube	70.8	**65.6**	**56.2**	**62.7**	**53.2**	62.5	53.6	45.1	**61.4**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Training data and details. Stage 1 uses approximately 11.02M image, multi-image, and video instruction records, including 3D-GRAND (Yang et al., 2025a), Grounded 3D-LLM (Chen et al., 2024b), Multi3DRefer (Zhang et al., 2023), HoliSpatial (Gao et al., 2026), and SPAR-7M-RGBD (Zhang et al., 2025a). Each record pairs 1–32 ordered frames with an instruction and response, covering scene QA, grounding, spatial relations, camera motion, RGB-D reasoning, and multi-view understanding. Stages 2–3 equally sample approximately 88.7K entries from ScanNet (Dai et al., 2017), ScanNet++ (Yeshwanth et al., 2023), DL3DV (Ling et al., 2024), RealEstate10K (Zhou et al., 2018), and ARKitScenes (Baruch et al., 2021). Stage 2 selects 3–16 references and one target under local, uniform, and wide-baseline strategies; Stage 3 samples 1–32 overlap-connected references while favoring larger valid motions. We initialize TinkerCube from previous work (Deng et al., 2025; Wang et al., 2026a). Details appear in Appendix D.

Benchmarks and evaluation protocol. Spatial reasoning uses BLINK (Fu et al., 2024), SPAR-Bench (Zhang et al., 2025a), ViewSpatial-Bench (Li et al., 2025a), VSI-Bench (Yang et al., 2025c), ScanQA (Azuma et al., 2022), and SQA3D (Ma et al., 2023); accuracy/EM assess correctness, and BLEU-4/CIDEr assess answers. MPI Sintel (Butler et al., 2012), NYU Depth V2 (Silberman et al., 2012), 7-Scenes (Shotton et al., 2013), and ETH3D (Schöps et al., 2017) assess target geometry (Appendix G). DL3DV (Ling et al., 2024) and RealEstate10K (Zhou et al., 2018) assess NVS fidelity (Appendix H).

Baselines. Spatial reasoning compares BAGEL (Deng et al., 2025), general VLMs (Cambrian-S-7B (Yang et al., 2026), Gemini-3-Pro-Preview (Team et al., 2023), GPT-5.2 (OpenAI, 2025), GPT-4o (OpenAI, 2024), Qwen2.5-VL-32B (Bai et al., 2025b), Qwen3-VL-8B, Qwen3-VL-8B-Instruct (Bai et al., 2025a), and MiniCPM-V-4.5-7B (Yu et al., 2026)), spatial models (MindCube-3B (Yin et al., 2025), Spatial-MLLM-4B (Wu et al., 2025a), SpatialLadder-3B (Li et al., 2026a), ViLaSR-7B (Wu et al., 2025b), VST-7B-RL (Yang et al., 2025d), SpaceR-7B (Ouyang et al., 2025), UniUGG (Xu et al., 2026), Omni-View (Hu et al., 2026a) and VLM-3R-7B (Fan et al., 2026)). Open-ended 3D QA further compares GeoVR-2B (Wang & Huang, 2026), Video-3D LLM (Zheng et al., 2025a), Chat-Scene (Huang et al., 2024), GPT4Scene-HDM (Qi et al., 2026) (Qwen2-VL-7B), LEO-VL (Huang et al., 2025a), Text-Scene (Li et al., 2025b), BridgeQA (Mo & Liu, 2024), Reg3D (Zheng et al., 2026), and SpaR3D-MoE (Feng et al., 2026). Geometry baselines comprise CUT3R (Wang et al., 2025b), DUSt3R/DUSt3R-GA (Wang et al., 2024a), Fast3R (Yang et al., 2025b), FLARE (Zhang et al., 2025c), MASt3R (Leroy et al., 2024), Metric3D v2 (Hu et al., 2024), MonST3R (Zhang et al., 2025b), π^3 (Wang et al., 2026b), Point3R (Wu et al., 2026), Spann3R (Wang & Agapito, 2025), MapAnything (Keetha et al., 2026), MegaSaM (Li et al., 2025c), VGGT (Wang et al., 2025a), Stream3R (Lan et al., 2026), StreamVGGT (Zhuo et al., 2026), Depth Anything 3 (Lin et al., 2026), VGGT-Ω (Wang et al., 2026a), G^{2}VLM (Hu et al., 2026b), and SPARGen (Quan et al., 2026). For NVS task, we compare BAGEL (Deng et al., 2025), MotionC-

trl (Wang et al., 2024b), CameraCtrl (He et al., 2024), ViewCrafter (Yu et al., 2024), See3D (Ma et al., 2025), IDESplat (Long et al., 2026), GEN3C (Ren et al., 2025), Voyager (Huang et al., 2025b), PE-Field (Bai et al., 2026), SEVA (Zhou et al., 2025) and MetaView (Cai et al., 2026).

4.2 Spatial Perception and Reasoning

Tables 1 and 2 show consistent gains across discriminative and open-ended reasoning. TinkerCube leads SPAR, ViewSpatial, and VSI-Bench while remaining competitive with substantially larger VLMs on BLINK, indicating that geometry-aware adaptation preserves semantic perception. Its ViewSpatial gains across camera- and object-centric relations, together with strong video reasoning on VSI-Bench, support GAR's role-specific camera and register tokens for viewpoint alignment and cross-view memory. Leading results on ScanQA and SQA3D further extend this advantage to free-form answers. The consistent pattern across static images, multi-view inputs, videos, and free-form responses suggests that GAR learns transferable spatial organization rather than relying on benchmark-specific answer priors or isolated geometric cues. This breadth also indicates that geometric supervision complements, rather than displaces, the backbone's visual and linguistic representations across task formats. These gains remain coherent across evaluation protocols. Overall, 2D tokens preserve semantic evidence, spatial tokens contribute view-consistent structure, and their shared language context enables query-dependent interaction.

Table 2: Open-ended 3D scene question answering on ScanQA and SQA3D.

Model	ScanQA			SQA3D
	EM ↑	BLEU-4 ↑	CIDEr ↑	EM ↑
General VLMs				
GPT-4o	–	–	72.2	47.8
Qwen3-VL-8B	28.2	–	80.5	51.1
Spatially Specialized Models				
Spatial-MLLM-4B	–	14.8	91.8	55.9
Chat-Scene	–	18.4	93.5	55.8
BridgeQA	31.3	**24.1**	83.8	52.9
Text-Scene	24.6	–	91.3	61.2
GPT4Scene-HDM	–	15.5	96.3	59.4
SpaR3D-MoE	30.4	17.1	101.5	58.3
Video-3D LLM	30.1	16.2	102.1	58.6
LEO	21.5	13.2	101.4	50.0
GeoVR	–	15.3	103.3	61.4
Reg3D	30.3	18.3	104.7	60.0
Multi-capability Models				
BAGEL	4.8	1.2	30.3	46.8
UniUGG	24.4	12.1	85.6	51.3
VLM-3R	–	15.5	101.9	60.7
Omni-View	29.5	16.2	103.0	59.2
TinkerCube	**31.5**	19.5	**113.4**	**61.5**

4.3 3D Geometry Prediction

Tables 3 and 5 demonstrate performance in target-view depth prediction and point-map reconstruction, while Table 4 confirms camera modeling. These gains indicate that progressive training transfers spatial representations across 3D tasks rather than optimizing isolated outputs. LCG training couples motion semantics with reference geometry through target tokens, while layerwise view modulation integrates pose, ray, and semantic supervision across the spatial encoder. Joint camera, depth, and cross-view objectives encourage these representations to encode viewpoint and surface structure, improving camera modeling, depth prediction, and point-map construction. This design allows each 3D objective to benefit from the geometric knowledge learned by the others, validating the effectiveness of progressive optimization and multi-modal interaction. The resulting geometry provides structural conditions for downstream RGG rendering. Appendix I.3 evaluates partial, compositional, and paraphrased camera instructions.

Table 3: Depth estimation on MPI Sintel and NYU Depth V2. AbsRel/Rel report relative depth error, and $\delta_{1.25}$ reports threshold accuracy.

Model	MPI Sintel		NYU Depth V2	
	AbsRel ↓	$\delta_{1.25}$ ↑	Rel ↓	$\delta_{1.25}$ ↑
Depth and Reconstruction Models				
FLARE	0.729	0.336	0.089	0.898
Fast3R	0.638	0.422	0.093	0.898
Spann3R	0.470	0.539	0.122	0.849
DUSt3R	0.488	0.532	0.081	0.909
MonST3R	0.399	0.519	0.094	0.887
CUT3R	0.418	0.520	0.081	0.914
MASt3R	0.340	0.604	0.129	0.849
Point3R	0.395	0.568	0.078	0.923
π^3	0.233	0.664	**0.054**	**0.956**
Multi-capability Models				
SPARGen	0.257	0.674	0.071	0.935
G^2VLM	0.297	0.589	0.062	0.954
TinkerCube	**0.209**	**0.690**	0.059	0.954

Table 4: Camera estimation on MPI Sintel and 7-Scenes. ATE measures trajectory error, and AUC summarizes pose accuracy across thresholds.

Model	MPI Sintel			7-Scenes	
	ATE ↓	AUC@3 ↑	AUC@30 ↑	AUC@3 ↑	AUC@30 ↑
MapAnything	0.202	2.9	31.6	5.8	61.4
MonST3R	0.108	4.3	45.8	9.0	68.3
CUT3R	0.213	–	–	6.5	71.8
VGGT	0.167	15.0	50.0	10.9	74.4
MegaSaM	0.074	22.5	58.3	10.6	71.8
Fast3R	0.371	–	–	19.0	78.6
π^3	0.074	14.8	53.5	13.3	77.0
StreamVGGT	0.219	–	–	13.5	81.4
Stream3R	0.213	–	–	13.1	82.1
Depth Anything 3	0.128	16.2	52.7	27.8	86.7
VGGT-Ω	0.057	35.3	73.0	29.6	83.1
TinkerCube	**0.043**	**36.5**	**75.4**	**30.4**	**86.9**

Table 5: Point-map reconstruction on 7-Scenes and ETH3D. Accuracy measures surface error; completeness measures reference coverage.

Model	7-Scenes		ETH3D	
	Acc. ↓	Comp. ↓	Acc. ↓	Comp. ↓
DUSt3R-GA	0.146	0.181	1.167	0.842
Fast3R	0.040	0.056	0.832	0.978
MASt3R	0.030	0.039	0.968	0.684
Spann3R	0.298	0.205	0.465	0.472
CUT3R	0.023	0.027	0.617	0.747
FLARE	0.019	0.026	0.464	0.664
VGGT	0.022	0.026	0.280	0.305
π^3	**0.016**	**0.022**	0.194	0.210
G^2VLM	0.062	0.031	0.539	0.355
SPARGen	0.034	0.028	0.393	0.445
TinkerCube	0.022	0.025	**0.148**	**0.142**

4.4 NOVEL-VIEW SYNTHESIS

We evaluate NVS using cameras realized by LCG under a shared trajectory protocol (Appendix H). Table 6 shows strong view synthesis on DL3DV and RealEstate10K. High photometric fidelity and structural similarity support faithful scene rendering under viewpoint changes. RGG connects these objectives through complementary conditions: target-aligned point-cloud rendering anchors scene layout, reference appearance preserves visual identity, and camera rays specify the observation direction. Coverage distinguishes supported projections from regions requiring generative rendering, allowing the generation expert to preserve geometric evidence while synthesizing newly visible content. This division of roles couples image fidelity with viewpoint-conditioned generation in a shared process. The cooperation generalizes across distinct scene distributions. Consistent gains over BAGEL across both datasets support the benefit of geometry-aware conditioning; matched-data fusion comparisons further demonstrate the value of coordinating projected structure, appearance, rays, and coverage through the proposed interfaces (Appendix I.4).

Table 6: NVS on DL3DV and RealEstate10K under the shared 25-frame trajectory protocol. Image metrics evaluate visual fidelity.

Model	DL3DV			RealEstate10K		
	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑
Camera-conditioned Generation						
MotionCtrl	13.647	0.363	0.323	16.431	0.279	0.505
CameraCtrl	13.701	0.359	0.317	17.724	0.245	0.565
ViewCrafter	13.846	0.347	0.341	16.830	0.253	0.533
See3D	14.490	0.320	0.363	17.124	0.319	0.526
Geometry-guided NVS						
IDESplat	14.599	0.307	0.401	17.464	0.229	0.572
GEN3C	15.202	0.344	0.366	18.037	0.221	0.587
Voyager	15.323	0.316	0.383	18.159	0.220	0.551
PE-Field	15.440	0.312	0.401	18.578	0.218	0.611
SEVA	15.674	0.285	0.397	18.825	0.199	0.562
MetaView	16.190	**0.259**	**0.507**	19.339	**0.198**	0.677
Multi-capability Models						
BAGEL	12.488	0.430	0.298	14.378	0.380	0.498
TinkerCube	**16.218**	0.263	0.503	**19.419**	0.204	**0.682**

4.5 ABLATION STUDIES

Tables 7–10 show complementary gains from spatial alignment, target-geometry learning, and rendering-guided adaptation. Joint semantic–geometric optimization improves reasoning and instruction-conditioned camera accuracy, while Stage 3 translates geometric evidence into high-quality NVS. Removing successive stages primarily weakens their associated capabilities, while freezing stage-specific pathways reduces cross-task transfer. Retrained ablations further show the largest reasoning degradation when appearance or register tokens are removed, whereas target rendering provides RGG's strongest geometric anchor for synthesis. GAR's appearance, camera, and register tokens supply semantic identity, viewpoint, and cross-view structure. RGG combines target rendering with reference appearance, rays, and coverage to support faithful, camera-aligned synthesis. These trends persist across all evaluated task families. The matched-data comparisons further support geometry-aware interaction beyond spatial instruction adaptation alone.

Table 7: Ablation of progressive stages. S1/S2/S3 denote spatial alignment, target geometry, and NVS training; VS, AUC30, and PSNR correspond to ViewSpatial, Sintel, and DL3DV.

S1 Align.	S2 Geom.	S3 NVS	SPAR ↑	VS Overall ↑	Sintel AUC30 ↑	DL3DV PSNR ↑
✓	✓	✓	**65.6**	**56.2**	**75.4**	**16.218**
✓	✓	×	65.6	56.2	75.4	13.440
✓	×	×	60.4	52.7	8.0	12.039
×	×	×	39.3	41.9	6.3	12.488

Table 8: Ablation of stage-specific trainable modules. UND, GEO, and COND denote the understanding expert, geometry path, and conditioning modules; metrics follow Table 7.

S1 UND	S2 UND	S2 GEO	S3 COND	SPAR ↑	VS Overall ↑	Sintel AUC30 ↑	DL3DV PSNR ↑
✓	✓	✓	✓	**65.6**	**56.2**	**75.4**	**16.218**
×	✓	–	–	43.5	44.1	–	–
✓	×	✓	–	60.4	52.7	66.5	–
✓	✓	×	–	63.7	55.8	49.7	–
✓	✓	✓	×	65.6	56.1	75.4	15.363

Table 9: Retrained component ablation of GAR.

Scene Token	Camera Token	Register Tokens	2D Appearance Tokens	SPAR ↑	ViewSpatial Overall ↑	SQA3D EM ↑	VSI-Bench ↑
✓	✓	✓	✓	**65.6**	**56.2**	**61.5**	**61.4**
×	✓	✓	✓	62.4	55.6	58.7	59.7
✓	×	✓	✓	61.5	53.8	56.5	57.6
✓	✓	×	✓	56.7	49.7	52.1	53.0
✓	✓	✓	×	33.4	28.4	4.3	27.5

Table 10: Retrained component ablation of RGG.

Target Render	Re-rendered Ref. Cache	Camera Ray	Local Coverage	PSNR ↑	LPIPS ↓	SSIM ↑
✓	✓	✓	✓	**16.218**	**0.263**	**0.503**
×	✓	✓	✓	12.230	0.456	0.273
✓	×	✓	✓	15.025	0.311	0.449
✓	✓	×	✓	15.592	0.279	0.487
✓	✓	✓	×	15.472	0.284	0.473

5 CONCLUSION

We presented TinkerCube, integrating spatial reasoning, 3D geometry prediction, and controllable novel-view synthesis. Geometry grounds reasoning, language directs camera and structure prediction, and rendering guides generation. Three-stage training coordinates these capabilities, while results validate multimodal training and 2D–3D interaction for scene understanding and exploration.

AI USE STATEMENT

Generative AI tools assisted with linguistic editing and manuscript revision. The authors developed the model, conducted the experiments, and reviewed the resulting text and analyses. Reported experimental values were obtained by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human-subject studies, the collection of personally identifiable data, or deployment in safety-critical settings. Experiments use established research datasets under their respective terms. We have reviewed the methodology and evaluation in relation to the ICLR Code of Ethics and identify no specific ethical concerns.

REPRODUCIBILITY STATEMENT

The appendix provides detailed descriptions of the implementation, data construction and processing, training objectives and hyperparameters, stage-wise procedures, inference pipelines, and evaluation protocols. These materials specify the configurations required to reproduce the reported results. The complete code is available through the anonymous link provided in the abstract.

REFERENCES

Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. ScanQA: 3d question answering for spatial scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19129–19139, 2022.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Yunpeng Bai, Haoxiang Li, and Qixing Huang. Positional encoding field. In *International Conference on Learning Representations*, volume 2026, pp. 84996–85008, 2026.

Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitScenes: A diverse real-world dataset for 3d indoor scene understanding using mobile RGB-D data. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, pp. 611–625, 2012.

Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. Spatialbot: Precise spatial understanding with vision language models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9490–9498. IEEE, 2025.

Yufei Cai, Xuesong Niu, Hao Lu, Kun Gai, Kai Wu, and Guosheng Lin. Metaview: Monocular novel view synthesis with scale-aware implicit geometry priors. *arXiv preprint arXiv:2607.12000*, 2026.

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

Yilun Chen, Shuai Yang, Haifeng Huang, Tai Wang, Runsen Xu, Ruiyuan Lyu, Dahua Lin, and Jiangmiao Pang. Grounded 3d-llm with referent tokens. *arXiv preprint arXiv:2405.10370*, 2024b.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5828–5839, 2017.

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

Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Peihao Wang, Huaizhi Qu, Shijie Zhou, Dilin Wang, Zhicheng Yan, Hongyu Xu, Justin Theiss, Tianlong Chen, Jiachen Li, Zhengzhong Tu, Zhangyang Wang, and Rakesh Ranjan. VLM-3R: Vision-language models augmented with instruction-aligned 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31054–31065, 2026.

Haida Feng, Hao Wei, Haolin Wang, Shiwei Li, Chade Li, and Yihong Wu. Spar3d-moe: Adaptive 3d spatial reasoning from sparse views meets geometry-inductive mixture-of-experts. *arXiv preprint arXiv:2607.06620*, 2026.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. BLINK: Multimodal large language models can see but not perceive. *arXiv preprint arXiv:2404.12390*, 2024.

Yuanyuan Gao, Hao Li, Yifei Liu, Xinhao Ji, Yuning Gong, Yuanjun Liao, Fangfu Liu, Manyuan Zhang, Yuchen Yang, Dan Xu, et al. Holi-spatial: Evolving video streams into holistic 3d spatial intelligence. *arXiv preprint arXiv:2603.07660*, 2026.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024.

Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3D-LLM: Injecting the 3D world into large language models, 2023. URL `https://arxiv.org/abs/2307.12981`.

Jiakui Hu, Shanshan Zhao, Qing-Guo Chen, Xuerui Qiu, Jialun Liu, Zhao Xu, Weihua Luo, Kaifu Zhang, and Yanye Lu. Omni-View: Unlocking how generation facilitates understanding in unified 3D model based on multiview images. In *International Conference on Learning Representations*, 2026a. URL `https://arxiv.org/abs/2511.07222`.

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

Wenbo Hu, Jingli Lin, Yilin Long, Yunlong Ran, Lihan Jiang, Yifan Wang, Chenming Zhu, Runsen Xu, Tai Wang, and Jiangmiao Pang. G^2VLM: Geometry grounded vision language model with unified 3d reconstruction and spatial reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9535–9546, 2026b.

Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, et al. Chat-scene: Bridging 3d scene and large language models with object identifiers. *Advances in Neural Information Processing Systems*, 37:113991–114017, 2024.

Jiangyong Huang, Xiaojian Ma, Xiongkun Linghu, Junchao He, Qing Li, Song-Chun Zhu, Yixin Chen, Baoxiong Jia, and Siyuan Huang. Leo-vl: Efficient scene representation for scalable 3d vision-language learning. *arXiv preprint arXiv:2506.09935*, 2025a.

Tianyu Huang, Wangguandong Zheng, Tengfei Wang, Yuhao Liu, Zhenwei Wang, Junta Wu, Jie Jiang, Hui Li, Rynson Lau, Wangmeng Zuo, et al. Voyager: Long-range and world-consistent video diffusion for explorable 3d scene generation. *ACM Transactions on Graphics (TOG)*, 44 (6):1–15, 2025b.

Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, et al. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1724–1734. IEEE, 2025.

Baoxiong Jia, Yixin Chen, Huangyue Yu, Yan Wang, Xuesong Niu, Tengyu Liu, Qing Li, and Siyuan Huang. SceneVerse: Scaling 3D vision-language learning for grounded scene understanding, 2024. URL https://arxiv.org/abs/2401.09340.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. Mapanything: Universal feed-forward metric 3d reconstruction; map-anything. github. io. In *2026 International Conference on 3D Vision (3DV)*, pp. 499–509. IEEE, 2026.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023.

Yushi Lan, Yihang Luo, Fangzhou Hong, Shangchen Zhou, Honghua Chen, Zhaoyang Lyu, Bo Dai, Shuai Yang, Chen Change Loy, and Xingang Pan. Stream3r: Scalable sequential 3d reconstruction with causal transformer. In *International Conference on Learning Representations*, volume 2026, pp. 42746–42768, 2026.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pp. 71–91. Springer, 2024.

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

Haoyuan Li, Rui Liu, Hehe Fan, and Yi Yang. Text-scene: A scene-to-language parsing framework for 3d scene understanding. *arXiv preprint arXiv:2509.16721*, 2025b.

Hongxing Li, Dingming Li, Zixuan Wang, Yuchen Yan, Hang Wu, Wenqi Zhang, Yongliang Shen, Weiming Lu, Jun Xiao, and Yueting Zhuang. SpatialLadder: Progressive training for spatial reasoning in vision-language models. In *International Conference on Learning Representations*, 2026a.

Yian Li, Yang Jiao, Bin Zhu, Tianwen Qian, Shaoxiang Chen, Jingjing Chen, and Yu-Gang Jiang. SpatialImaginer: Towards adaptive visual imagination for spatial reasoning. *arXiv preprint arXiv:2604.17385*, 2026b.

Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. Megasam: Accurate, fast, and robust structure and motion from casual dynamic videos. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10486–10496. IEEE, 2025c.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In *International Conference on Learning Representations*, 2026.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. DL3DV-10K: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22160–22169, 2024.

Wei Long, Haifeng Wu, Shiyin Jiang, Jinhua Zhang, Xinchun Ji, and Shuhang Gu. Idesplat: Iterative depth probability estimation for generalizable 3d gaussian splatting. *arXiv preprint arXiv:2601.03824*, 2026.

Baorui Ma, Huachen Gao, Haoge Deng, Zhengxiong Luo, Tiejun Huang, Lulu Tang, and Xinlong Wang. You see it, you got it: Learning 3d creation on pose-free videos at scale. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2016–2029. IEEE, 2025.

Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. SQA3D: Situated question answering in 3d scenes. In *International Conference on Learning Representations*, 2023.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pp. 405–421, 2020.

Wentao Mo and Yang Liu. Bridging the gap between 2d and 3d visual question answering: A fusion approach for 3d vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 4261–4268, 2024.

OpenAI. GPT-4o System Card, August 2024. URL https://arxiv.org/abs/2410.21276.

OpenAI. OpenAI GPT-5 System Card, August 2025. URL https://arxiv.org/abs/2601.03267.

Kun Ouyang, Yuanxin Liu, Haoning Wu, Yi Liu, Hao Zhou, Jie Zhou, Fandong Meng, and Xu Sun. Spacer: Reinforcing mllms in video spatial reasoning. *arXiv preprint arXiv:2504.01805*, 2025.

Zhangyang Qi, Zhixiong Zhang, Ye Fang, Jiaqi Wang, and Hengshuang Zhao. Gpt4scene: Understand 3d scenes from videos with vision-language models. In *International Conference on Learning Representations*, volume 2026, pp. 157277–157302, 2026.

Jinsheng Quan, Jianhua Li, Siyi Xie, Xuanke Shi, Kewang Deng, Zukai Chen, Feifei Shao, Lei Yang, Quan Wang, and Yawei Luo. SPARGen: Unifying spatial perception and reasoning through native multimodal generation. *arXiv preprint arXiv:2608.14138*, 2026.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. GEN3C: 3d-informed world-consistent video generation with precise camera control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6121–6132, 2025.

Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3260–3269, 2017.

Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2930–2937, 2013.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *European Conference on Computer Vision*, pp. 746–760, 2012.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Haibo Wang and Lifu Huang. Learning geometric representations from videos for spatial intelligent multimodal large language models. *arXiv preprint arXiv:2606.05833*, 2026.

Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *2025 International Conference on 3D Vision (3DV)*, pp. 78–89. IEEE, 2025.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5294–5306, 2025a.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. *arXiv preprint arXiv:2605.15195*, 2026a.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10510–10522. IEEE, 2025b.

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

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20697–20709, 2024a.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations*, volume 2026, pp. 10481–10497, 2026b.

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024b.

Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-MLLM: Boosting MLLM capabilities in visual-based spatial intelligence. In *Advances in Neural Information Processing Systems*, volume 38, 2025a.

Junfei Wu, Jian Guan, Kaituo Feng, Qiang Liu, Shu Wu, Liang Wang, Wei Wu, and Tieniu Tan. Reinforcing spatial reasoning in vision-language models with interwoven thinking and visual drawing. In *Advances in Neural Information Processing Systems*, volume 38, 2025b.

Yuqi Wu, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Point3r: Streaming 3d reconstruction with explicit spatial pointer memory. *Advances in Neural Information Processing Systems*, 38:69675–69699, 2026.

Yueming Xu, Jiahui Zhang, Ze Huang, Yurui Chen, Yanpeng Zhou, Zhenyu Chen, Yu-Jie Yuan, Pengxiang Xia, Guowei Huang, Xinyue Cai, Zhongang Qi, Xingyue Quan, Jianye Hao, Hang Xu, and Li Zhang. UniUGG: Unified 3D understanding and generation via geometric-semantic encoding. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=XbMp83tvK7.

Jianing Yang, Xuweiyi Chen, Nikhil Madaan, Madhavan Iyengar, Shengyi Qian, David F. Fouhey, and Joyce Chai. 3d-grand: A million-scale dataset for 3d-llms with better grounding and less hallucination, 2025a. URL https://arxiv.org/abs/2406.05132.

Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21924–21935. IEEE, 2025b.

Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025c.

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

Shusheng Yang, Jihan Yang, Pinzhi Huang, Ellis Brown, Zihao Yang, Yue Yu, Shengbang Tong, Zihan Zheng, Yifan Xu, Muhan Wang, et al. Cambrian-s: Towards spatial supersensing in video. In *International Conference on Learning Representations*, volume 2026, pp. 78185–78225, 2026.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12–22, 2023.

Baiqiao Yin, Qineng Wang, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, et al. Spatial mental modeling from limited views. In *Structural Priors for Vision Workshop at ICCV'25*, 2025.

Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Ranchi Zhao, et al. Minicpm-v 4.5: Cooking efficient mllms via architecture, data, and training recipe. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11704–11715, 2026.

Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. ViewCrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024.

Jiahui Zhang, Yurui Chen, Yanpeng Zhou, Yueming Xu, Ze Huang, Jilin Mei, Junhui Chen, Yu-Jie Yuan, Xinyue Cai, Guowei Huang, Xingyue Quan, Hang Xu, and Li Zhang. From flatland to space: Teaching vision-language models to perceive and reason in 3d. *arXiv preprint arXiv:2503.22976*, 2025a.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Forrester Cole, Deqing Sun, Ming-Hsuan Yang, et al. Monst3r: A simple approach for estimating geometry in the presence of motion. In *International Conference on Learning Representations*, volume 2025, pp. 82863–82886, 2025b.

Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21936–21947. IEEE, 2025c.

Yiming Zhang, ZeMing Gong, and Angel X Chang. Multi3drefer: Grounding text description to multiple 3d objects. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15179–15179. IEEE, 2023.

Shanshan Zhao, Xinjie Zhang, Jintao Guo, Jiakui Hu, Lunhao Duan, Minghao Fu, Yong Xien Chng, Guo-Hua Wang, Qing-Guo Chen, Zhao Xu, Weihua Luo, and Kaifu Zhang. Unified multimodal understanding and generation models: Advances, challenges, and opportunities. *arXiv preprint arXiv:2505.02567*, 2025.

Duo Zheng, Shijia Huang, and Liwei Wang. Video-3d llm: Learning position-aware video representation for 3d scene understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8995–9006. IEEE, 2025a.

Guangcong Zheng, Teng Li, Xianpan Zhou, and Xi Li. RealCam-Vid: High-resolution video dataset with dynamic scenes and metric-scale camera movements, 2025b. URL `https://arxiv.org/abs/2504.08212`.

Hongpei Zheng, Lintao Xiang, Lin Qian, Zhenghao Li, Qijun Yang, and Hujun Yin. Reg3d: Reconstructive geometry instruction tuning for 3d scene understanding. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1301–1305. IEEE, 2026.

Jensen Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12405–12414. IEEE, 2025.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics*, 37(4):65:1–65:12, 2018.

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming visual geometry transformer. In *International Conference on Learning Representations*, volume 2026, pp. 88055–88072, 2026.

A Qualitative Results

A.1 Qualitative Spatial Question Answering

BLINK: Spatial Relation



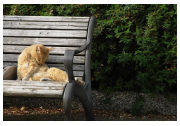
Question: Is the dog on the motorcycle? \n Select from the following choices.\n(A) yes\n(B) no\n Your answer can only include one of options A, B.\n\nAnswer with the option's letter from the given choices directly.
Prediction: B
Answer: (B)



Question: Is the person facing away from the horse?\nSelect from the following choices.\n(A) yes\n(B) no\nYour answer can only include one of options A, B.\n\nAnswer with the option's letter from the given choices directly.
Prediction: A
Answer: (A)



Question: Is the orange at the right side of the umbrella?\nSelect from the following choices.\n(A) yes\n(B) no\nYour answer can only include one of options A, B.\n\nAnswer with the option's letter from the given choices directly.
Prediction: B
Answer: (B)



Question: Is the bench at the left side of the cat?\nSelect from the following choices.\n(A) yes\n(B) no\nYour answer can only include one of options A, B.\n\nAnswer with the option's letter from the given choices directly.
Prediction: B
Answer: (B)

Figure 3: **Object-centric spatial reasoning on BLINK.** TinkerCube accurately recognizes relative positions, orientations, and contact relations between objects in diverse single-image scenes.

BLINK: Relative Depth & Multi-view Reasoning



Question: Two points are circled on the image, labeled by A and B beside each circle. Which point is closer to the camera?\nSelect from the following choices.\n(A) A is closer\n(B) B is closer\nYour answer can only include one of options A, B.\n\nAnswer with the option's letter from the given choices directly.
Prediction: A
Answer: (A)



Question: Two points are circled on the image, labeled by A and B beside each circle. Which point is closer to the camera?\nSelect from the following choices.\n(A) A is closer\n(B) B is closer\nYour answer can only include one of options A, B.\n\nAnswer with the option's letter from the given choices directly.
Prediction: A
Answer: (A)



Question: The images are frames from a video. The video is shooting a static scene. The camera is either moving clockwise (left) or counter-clockwise (right) around the object. The first image is from the beginning of the video and the second image is from the end. Is the camera moving left or right when shooting the video? Select from the following options.\n(A) left\n(B) right\nYour answer can only include one of options A, B.\n\nAnswer with the option's letter from the given choices directly.
Prediction: A
Answer: (A)

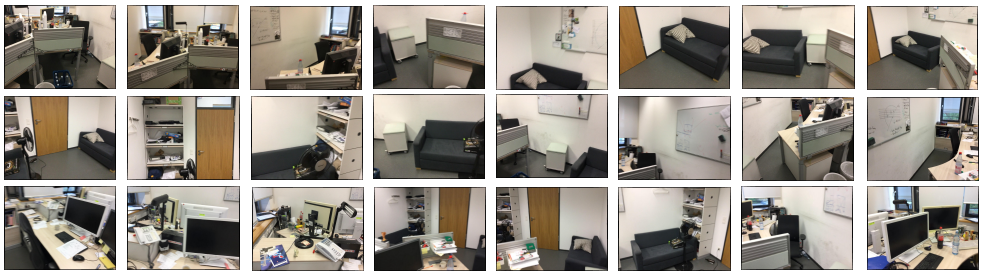


Question: The images are frames from a video. The video is shooting a static scene. The camera is either moving clockwise (left) or counter-clockwise (right) around the object. The first image is from the beginning of the video and the second image is from the end. Is the camera moving left or right when shooting the video? Select from the following options.\n(A) left\n(B) right\nYour answer can only include one of options A, B.\n\nAnswer with the option's letter from the given choices directly.
Prediction: A
Answer: (A)

Figure 4: **Depth and multi-view reasoning on BLINK.** TinkerCube resolves relative depth from individual images and infers camera-motion direction from paired observations.

As shown in Figs. 3–6, the examples span complementary forms of spatial reasoning, from single-image object relations and relative depth to cross-view camera motion, scene-level question answering, and egocentric reasoning. TinkerCube consistently identifies relevant objects, integrates observations across views, and interprets spatial relations from the queried perspective. Its predictions align with the reference answers, illustrating how 3D-aware representations complement visual semantics for robust spatial reasoning across diverse scenes and question types.

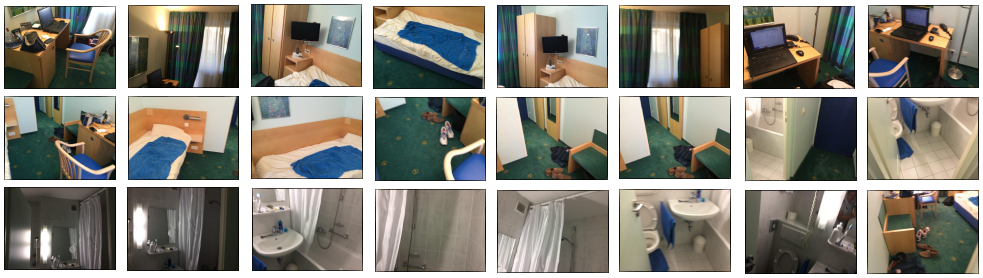
ScanQA



Question: On what color desk is the monitor closest to a whiteboard? **Prediction:** light brown desk **GT:** light brown
Question: How many monitors are on shorter curved table? Answer the question simply. **Prediction:** 2 **GT:** 2 monitors
Question: Where is the monitor sitting? Answer the question simply. **Prediction:** on desk **GT:** on desk
Question: What is the shelf next to? Answer the question simply. **Prediction:** door **GT:** next to door
Question: The keyboard closest to the door is located on the desk in front of what? **Prediction:** monitor **GT:** monitor
Question: What color sticky note is attached to the monitor? Answer the question simply. **Prediction:** yellow **GT:** yellow
Question: In what corner of a black couch is the gray pillow located? **Prediction:** left corner of it **GT:** left corner
Question: What color is the file cabinet? Answer the question simply. **Prediction:** white **GT:** white
Question: Where is the black office chair located? Answer the question simply. **Prediction:** in front of the desk **GT:** in front of desks
Question: What color is the door? Answer the question simply. **Prediction:** brown **GT:** brown
Question: What shape is the desk on the left side of the room? Answer the question simply. **Prediction:** curved **GT:** curved

Figure 5: **Multi-view question answering on ScanQA.** TinkerCube integrates observations across an indoor scene to reason about object attributes, counts, locations, and spatial relations.

SQA3D



Question: I am washing my face at the sink with a bathtub on my right. Is the toilet seat covered up or down to my left? Answer the question using a single word or phrase. **Prediction:** up **GT:** up
Question: I am standing by the toilet facing the bathtub. Do the mat and the towel on my left have different colors? Answer the question using a single word or phrase. **Prediction:** no **GT:** no
Question: I am standing picking up my pillow with a backpack in my six o'clock direction. Are the curtains on my 8 o'clock direction open or closed? Answer the question using a single word or phrase. **Prediction:** closed **GT:** closed
Question: I am lying on the bed with a table in my nine o'clock direction and facing the wardrobe. Is the TV on or off in front of me to the right? Answer the question using a single word or phrase. **Prediction:** off **GT:** off
Question: I am using the toilet. What is the blue object to the left of me not on the floor? Answer the question using a single word or phrase. **Prediction:** towel **GT:** towel

Figure 6: **Situated question answering on SQA3D.** TinkerCube interprets viewpoint-dependent descriptions and reasons about object states, directions, and relations from an embodied perspective.

A.2 Controllable Spatial Scene Exploration

As shown in Figs. 7 and 8, the two sequences illustrate a shared inference process across language control, geometry, and generation. Motion descriptions and captions specify the intended camera displacement and scene content, while the geometry expert predicts the corresponding pose and depth. These estimates support point-cloud reconstruction and produce target-view renderings that provide geometric scaffolds for image generation. Across wide viewpoint changes, TinkerCube preserves the principal scene layout and recovers plausible appearance where direct projection is incomplete. The aligned pose, depth, rendering, NVS, and point-cloud outputs demonstrate how semantic and geometric representations cooperate to support controlled exploration beyond the initial observation.

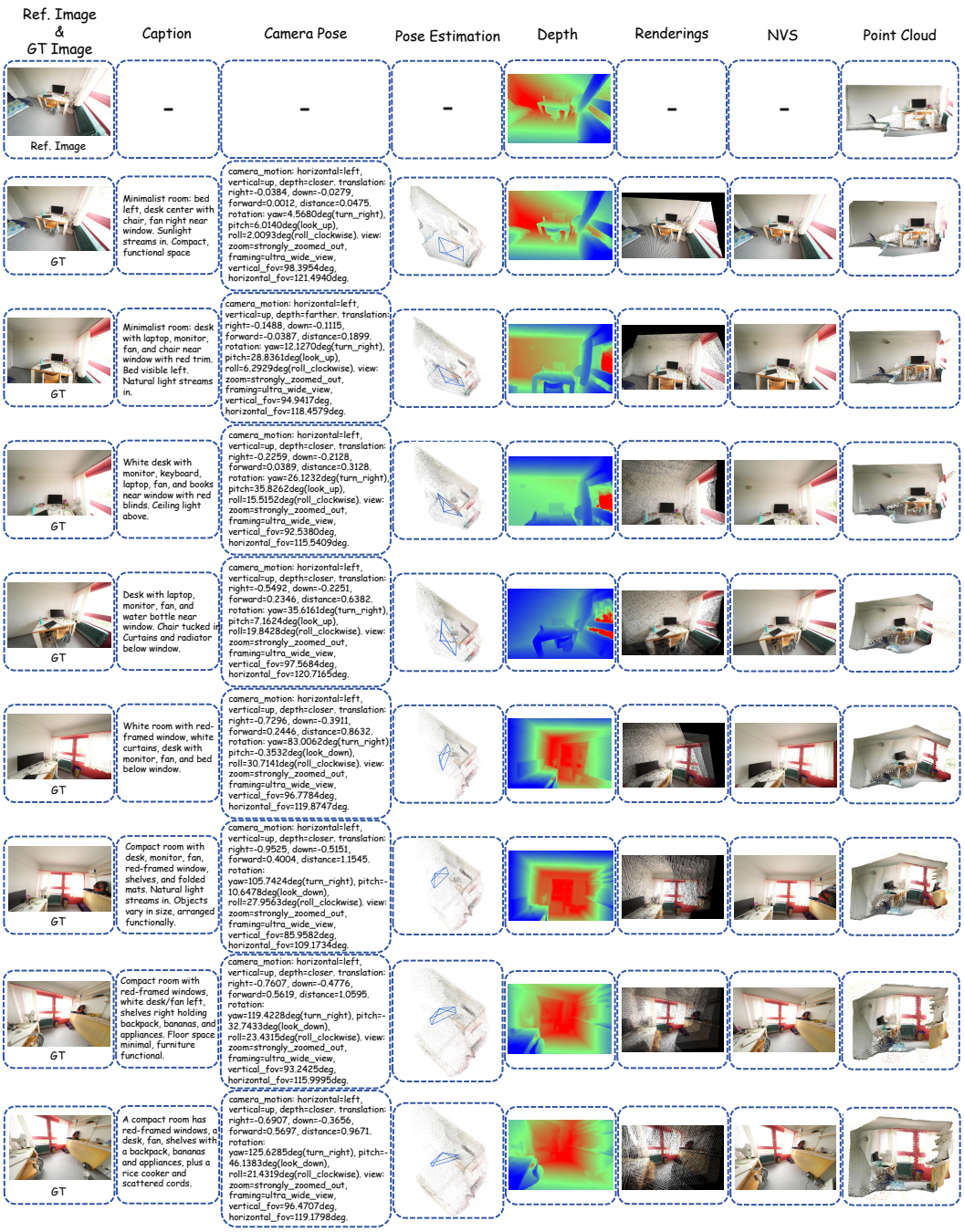


Figure 7: Qualitative results of TinkerCube for controllable spatial scene exploration. Starting from a single reference image, the model interprets captions and motion instructions, estimates target camera poses and depths, renders the recovered geometry at each requested viewpoint, synthesizes novel-view images, and constructs corresponding point clouds. Ground-truth images are shown only for comparison.

B TASK-SPECIFIC INFERENCE

TinkerCube exposes a shared set of representations rather than a single fixed output interface. At inference time, the requested task determines where the computation terminates and which functional modules are activated.



Figure 8: Qualitative results of TinkerCube for controllable spatial scene exploration. Starting from a single reference image, the model interprets captions and motion instructions, estimates target camera poses and depths, renders the recovered geometry at each requested viewpoint, synthesizes novel-view images, and constructs corresponding point clouds. Ground-truth images are shown only for comparison.

Spatial perception and reasoning. Given ordered observations $\mathcal{I}_r = \{I_i\}_{i=1}^{N}$ and a question x_q, GAR writes appearance, scene, camera, and register tokens into the common context. The understanding expert then autoregressively decodes

$$\hat{y} = \mathrm{ARDecode}(\Phi_{\mathrm{GAR}}(\mathcal{I}_r, x_q)) \,. \tag{16}$$

This path uses the complementary semantic detail of $\mathcal{F}_v$ and the cross-view structure of $\mathcal{F}_s$, but does not invoke target-view construction or image sampling.

Target-view geometry prediction. For a requested target observation, LCG consumes reference images, motion m_t, and an optional content description c_t. The spatial encoder extracts reference structure, while the understanding expert supplies the language conditions. Learnable target tokens integrate this evidence through layerwise modulation and residual decoding to predict

$$(\hat{\mathbf{p}}_t, \hat{D}_t, \hat{P}_t) = \Phi_{\mathrm{LCG}}(\mathcal{I}_r, m_t, c_t), \qquad \hat{P}_t(u) = \pi^{-1}_{\hat{\mathbf{p}}_t}(u, \hat{D}_t(u)). \tag{17}$$

Motion specifies the requested observation; reference structure and layerwise modulation refine its geometric realization. Multiple predicted point maps can be fused in the reference coordinate frame for explicit scene modeling.

Novel-view synthesis. LCG predicts the target camera $\hat{\mathbf{p}}_t$ from the motion instruction and reference context. RGG constructs its colored point cloud from available references, projects it under this camera, and performs scene-image rendering in latent space under the semantic, appearance, ray, and coverage conditions:

$$\hat{I}_t = \Phi_{\mathrm{RGG}}(\mathcal{I}_r, \hat{\mathbf{p}}_t, m_t, c_t). \tag{18}$$

Target RGB, target depth, and target point maps are never inserted into the reference point cloud. Consequently, the generated view remains grounded in observed scene evidence while language and camera conditions determine the requested observation.

TinkerCube supports progressive scene expansion by incorporating synthesized views into the reference set and updating the reconstructed point cloud. Here, scene dynamics refer to evolving spatial coverage rather than motion of objects within the scene. Repeated prediction and rendering thus extend the scene representation beyond the initial observations.

C Training Data and Sample Construction

C.1 Data Sources and Train–Evaluation Separation

Stage 1 contains approximately 11.02M instruction records, each comprising 1–32 ordered images or video frames, an instruction, and an answer. Representative sources include 3D-LLM (Hong et al., 2023), Grounded 3D-LLM (Chen et al., 2024b), HoliSpatial (Gao et al., 2026), SceneVerse (Jia et al., 2024), SpaceR (Ouyang et al., 2025), SpatialQA (Cai et al., 2025), and RealCam-Vid (Zheng et al., 2025b). They provide scene description, grounding, spatial relations, camera-aware reasoning, and temporally ordered observations.

Stages 2–3 draw from approximately 88.7K scene or trajectory entries across ScanNet, ScanNet++, DL3DV, RealEstate10K, and ARKitScenes. Sampling first selects one of the five sources with equal probability and then selects a scene, balancing geometry and rendering supervision across sources. All three training stages use explicit scene- or trajectory-level splits disjoint from the evaluation sets. Evaluation scenes and trajectories are excluded from training, so the reported results assess generalization to held-out scenes.

C.2 Stage 1 Instruction Samples

A Stage 1 sample is $\mathcal{X}^{(1)} = (\{I_i\}_{i=1}^{N}, x^{\mathrm{human}}, y^{\mathrm{assistant}})$, with $1 \le N \le 32$. Image, multi-image, and video records are converted to the same conversation format while preserving frame order. The mixture covers recognition and attributes, relative position, depth ordering, referring expressions, camera motion, cross-view correspondence, and compositional spatial reasoning. The answer alone is supervised, allowing the question and visual-spatial context to serve as causal conditioning information.

For a video with L available frames, we first choose N and divide the ordered index range into N intervals. One frame is sampled from each interval with a bounded random offset, yielding an ordered, approximately uniform subset $\mathcal{J} = (j_1 < \cdots < j_N)$. This uniform-with-jitter strategy retains the global temporal extent while preventing the spatial encoder from overfitting to a fixed

stride. Multi-image records preserve their annotated order. The 2D branch uses dynamic-resolution inputs, whereas the spatial branch receives 256×256 views, allowing semantic detail and multi-view structure to be packed under a shared token budget.

C.3 Stage 2 Target-View Samples

Stage 2 constructs a shared reference set and two distinct target candidates from each sampled scene. The two candidates are emitted as independent training examples, so every forward pass still contains 3–16 references and one target, as specified in the main paper. This construction increases target coverage without coupling the gradients of two target views in a single forward pass.

Reference sampling. We use local, uniform, and wide-baseline modes with probabilities $0.50/0.35/0.15$. Let s denote a source-specific temporal unit used to normalize index distances, let a be an anchor sampled uniformly from valid frames, and let $d_j = |j - a|$ be a candidate's distance from the anchor. Candidate frames are partitioned into

$$\begin{aligned} \mathcal{B}_{\text{near}} &: d \le \max(2s, N/2), \\ \mathcal{B}_{\text{mid}} &: \max(2s, N/2) < d \le \max(8s, 2N), \\ \mathcal{B}_{\text{far}} &: \max(8s, 2N) < d \le \max(20s, 4N). \end{aligned} \tag{19}$$

Local, uniform, and wide-baseline sampling draw from $(\mathcal{B}_{\text{near}}, \mathcal{B}_{\text{mid}}, \mathcal{B}_{\text{far}})$ with proportions $(0.75, 0.25, 0)$, $(0.40, 0.50, 0.10)$, and $(0.30, 0.45, 0.25)$, respectively. Thus, local samples emphasize dense correspondence, uniform samples cover the trajectory, and wide-baseline samples expose the model to larger viewpoint changes.

More precisely, after sampling the desired reference count N, mode $g \in \{\text{local}, \text{uniform}, \text{wide}\}$ determines a quota $n_b = \text{Round}(N w_{g,b})$ for each temporal bin b. Frames are drawn without replacement inside each bin. If a bin contains fewer than its quota, the deficit is redistributed to the nearest non-empty bin; any remaining slots are filled from the union of valid candidates according to increasing temporal distance for local mode and stratified trajectory position for the other two modes. The selected indices are finally sorted before encoding:

$$\mathcal{J}_r = \text{Sort}\left(\bigcup_{b \in \{\text{near},\text{mid},\text{far}\}} \text{SampleWithoutReplacement}(\mathcal{B}_b, n_b) \right). \tag{20}$$

Consequently, every sample has a deterministic view order even though its baseline composition changes stochastically.

Motion and content conditions. Motion instructions describe the intended viewpoint change relative to the first ordered reference. They may specify horizontal, vertical, and depth-wise translation; yaw, pitch, and roll; and zoom, framing, and horizontal or vertical field of view. These factors can be expressed categorically or at finer granularity, allowing the same interface to represent coarse directions, partial requests, and compositional camera motions.

We stochastically simplify m_t during training. With probability 0.50, only the `camera_motion` field is retained to describe the overall motion direction. In the remaining cases, each of the `translation`, `rotation`, `roll`, and `vertical_fov` fields is independently dropped with probability 0.25. This exposes LCG to complete, coarse, and partially specified requests, encouraging it to identify the intended motion and use reference-scene context to resolve unspecified viewing degrees of freedom. Target captions summarize visible object categories, relative layouts, and coarse physical-size relations, providing semantic conditions for content-controlled target prediction and generation.

Target difficulty is sampled with near/medium/far probabilities $0.20/0.55/0.25$. The second target is selected before, inside, or after the reference span with probabilities $0.30/0.40/0.30$ and must be separated from the first target by at least $\max(2, s)$ frames. With probability 0.30, candidates are ranked by normalized translation and rotation and a large-motion pair is preferred.

Validity filtering. Frames with missing or non-finite camera parameters, a degenerate quaternion, or an invalid field of view are excluded. A depth value is valid when it is finite and exceeds 10^{-4};

depth, normal, and point losses are evaluated only on the resulting mask. Samples without any valid target depth are resampled. Reprojection supervision further keeps only pixels that project inside a reference image and agree with its visible depth. These rules avoid turning missing measurements into geometric supervision while retaining all available surface evidence.

C.4 Stage 3 Overlap-Connected Samples

Stage 3 samples 1–32 references and target frames from the same scene. Rather than selecting isolated frames independently, overlap-connected sampling first chooses an anchor and grows a chain by randomly extending either temporal direction. Every selected reference is required to share a valid edge with at least one other selected reference; a one-reference sample is the only singleton exception. This produces a connected set from which a coherent reference point cloud can be formed.

The hard edge criterion is relative camera rotation. For ScanNet, ScanNet++, DL3DV, and RealEstate10K, the temporal search window is 96 frames and the rotation limit is 105°. For ARKitScenes, whose trajectories contain denser local turns, the corresponding values are 32 frames and 50°. Once a direction encounters the first frame beyond the rotation limit, that direction is closed and no fallback crosses the boundary. If both ends are closed, the sampler searches valid frames around the interior of the existing chain. Candidate probability is proportional to $\exp(4\theta/\theta_{\max})$, and the optional large-pair draw uses $\exp(6\theta/\theta_{\max})$, giving informative viewpoints greater weight while respecting the connectivity constraint.

Translation and image-space coverage are retained as diagnostics rather than additional hard edge rejection criteria. Translation is normalized by the median adjacent-camera baseline, with 10 used as the diagnostic upper range; the nominal field-of-view overlap factor is 0.90. This choice preserves valid forward-motion and scale-varying trajectories. Actual support is measured after point-cloud rendering by the binary coverage map M, which is explicitly provided to RGG. A sequence that cannot produce the requested connected reference count and distinct valid targets within 64 trials is resampled.

With probability 0.60, a target is sampled near the latest chain endpoint; otherwise it is sampled around the chain interior. Two target candidates are constructed for broader view coverage and optimized independently. Point-cloud rendering uses the raw projection mask: uncovered pixels remain black and no hole-filling iteration is applied. Coverage therefore separates genuine dark appearance from regions without projected support.

D Architectural and Optimization Details

D.1 Spatial Readout and Reference Memory

Scene token. The scene representation $\mathbf{g}_i^{\mathrm{sc}}$ used by GAR is obtained from the final camera and register states of $\mathcal{F}_s$. We normalize and flatten the $N(1+16)$ structural tokens, prepend a learnable scene readout, and apply four self-attention blocks:

$$\mathbf{g}^{\mathrm{sc}} = \mathrm{Norm}\left(\mathrm{Readout}([\mathbf{q}_{\mathrm{sc}}; \{\mathbf{g}_i^{\mathrm{cam}}, \mathbf{G}_i^{\mathrm{reg}}\}_{i=1}^{N}])\right). \tag{21}$$

The shared readout is placed in each view block as $\mathbf{g}_i^{\mathrm{sc}} = \mathbf{g}^{\mathrm{sc}}$ for $i = 1, \ldots, N$; the view index denotes its position in the sequence. The readout is projected and ℓ_2-normalized before Role-Aware Spatial Projection. It summarizes scene-wide structure without averaging away the distinct camera and register representations that are separately exposed to the language model.

Condition pooling. Pool_K in Equation 4 converts a variable-length language span $\mathbf{H} = (\mathbf{h}_1, \ldots, \mathbf{h}_L)$ to $K = 32$ ordered tokens by adaptive average pooling:

$$\mathbf{q}_k = \frac{1}{|\mathcal{A}_k|} \sum_{\ell \in \mathcal{A}_k} \mathbf{h}_\ell, \quad \mathcal{A}_k = \left\{ \ell : \left\lfloor \frac{(k-1)L}{K} \right\rfloor + 1 \le \ell \le \left\lceil \frac{kL}{K} \right\rceil \right\}. \tag{22}$$

Here $k = 1, \ldots, K$. This gives motion and caption conditions a fixed interface while retaining their coarse token order.

Reference memory. For each reference view, $\mathbf{M}_r$ contains its camera token, 16 register tokens, and a 4×4 adaptive pooling of patch tokens. After projection to the decoder dimension,

$$\mathbf{M}_r = \mathrm{Concat}_{i=1}^{N}[\mathbf{g}_i^{\mathrm{cam}}; \mathbf{G}_i^{\mathrm{reg}}; \mathrm{Pool}_{4\times4}(\mathbf{G}_i^{\mathrm{patch}})], \tag{23}$$

so the memory contains $33N$ tokens. The compact representation preserves camera state, cross-view summary, and coarse spatial layout while keeping target decoding efficient.

D.2 Layerwise View Modulation

LCG initializes the target with one camera token, 16 register tokens, and a 32×32 learnable patch grid. The target grid is bicubically interpolated when a different spatial resolution is required. View modulation is inserted after spatial-encoder layers $\{2, 6, 9, 14, 20, 23\}$. At each insertion, target camera/register tokens attend to the pose condition, while target patch tokens attend successively to caption semantics and pose tokens and receive the corresponding Plücker-ray embedding. Each block uses pre-normalized cross-attention and feed-forward residuals. For the six insertions, the outer update is written as

$$\mathbf{T}_t^{(\ell+1)} = \mathbf{T}_t^{(\ell)} + \alpha\, \Delta_\ell\Big(\mathbf{T}_t^{(\ell)}; \mathbf{Q}_c, \mathbf{Q}_p, \mathbf{Q}_\rho\Big), \qquad \alpha = \frac{1}{\sqrt{|\mathcal{L}_c|}} = \frac{1}{\sqrt{6}}, \tag{24}$$

where $\mathcal{L}_c = \{2, 6, 9, 14, 20, 23\}$ is the set of conditioned layers and Δ_ℓ is the pre-normalized attention–MLP residual predicted at layer ℓ. The scale α distributes conditioning across depth without overwhelming the spatial backbone states.

The main model uses full inter-view attention among observed tokens and the learnable target tokens. Because the target tokens contain no target RGB or target geometry, this attention pattern does not expose target observations. When reference states are cached, an asymmetric mask prevents reference queries from reading target keys while allowing target queries to retrieve the reference memory. The residual path always retains the unconditioned spatial state, making language and pose control additive refinements rather than a replacement for reference structure.

D.3 Camera Coordinates and Back-Projection

The first ordered reference defines the common frame. A world-to-camera pose maps a point $\mathbf{X}$ to $\mathbf{x}_c = \mathbf{R}\mathbf{X} + \mathbf{t}$, with camera center $\mathbf{o} = -\mathbf{R}^\top \mathbf{t}$. The bounded camera residuals in Equation 8 are applied as

$$\begin{aligned} \hat{\mathbf{o}}_t &= \mathbf{o}_t^0 + \Delta\mathbf{o}_t, \quad \hat{\mathbf{R}}_t = \exp([\Delta\boldsymbol{\omega}_t]_\times)\mathbf{R}_t^0, \\ \hat{\mathbf{t}}_t &= -\hat{\mathbf{R}}_t\hat{\mathbf{o}}_t, \qquad \hat{\mathbf{f}}_t = \exp(\log \mathbf{f}_t^0 + \Delta\mathbf{f}_t). \end{aligned} \tag{25}$$

Here $[\cdot]_\times$ is the skew-symmetric cross-product matrix, and $\Delta\mathbf{f}_t$ is the log-FoV residual. Thus camera-center refinement is expressed as world-to-camera translation for supervision and projection. Quaternions use (q_x, q_y, q_z, q_w) order and are normalized before conversion to rotations, with the identity quaternion used for near-zero norms. Sign-equivalent q and $-q$ represent the same orientation.

For homogeneous pixel coordinate $\bar{u}$ and camera intrinsics $\mathbf{K}_i$, the point map is

$$\hat{P}_i(u) = \hat{\mathbf{R}}_i^\top \left(\hat{D}_i(u)\hat{\mathbf{K}}_i^{-1}\bar{u} - \hat{\mathbf{t}}_i\right). \tag{26}$$

Depth is measured along the camera's optical axis. Intrinsics follow the crop, resize, and padding applied to the image, so depth, rays, and point maps share one pixel grid.

D.4 Depth, Point Clouds, and Rendering

Only finite positive depth values are unprojected. Surface normals require a valid 2×2 neighborhood, and reprojection uses in-frame, depth-consistent pixels. The main configuration uses this deterministic validity mask for optimization; predicted confidence is retained as a diagnostic signal rather than a source-dependent cutoff.

For RGG, each valid reference pixel is transformed into the shared world frame with its RGB value. At most 800K points are retained using a balanced per-reference quota and deterministic uniform

subsampling. We preserve multi-view observations instead of voxel-merging them prematurely. During rendering, points are projected to target pixels and a z-buffer keeps the nearest surface; colors from points within a relative depth tolerance of 10^{-4} are averaged. This resolves duplicate projections while preserving the appearance evidence supplied by multiple references.

D.5 Stage 2 Supervision

Stage 2 supervises language-conditioned target-camera prediction, dense geometry, and cross-view consistency. For target view t, the point-map label is obtained by $P_t^*(u) = \pi_{\mathbf{p}_t^*}^{-1}(u, D_t^*(u))$. The three loss terms in Equation 14 are defined as

$$\begin{aligned} \mathcal{L}_{\text{pose}} &= \ell_1(\hat{\mathbf{t}}_t, \mathbf{t}_t^*) + d_R(\hat{\mathbf{R}}_t, \mathbf{R}_t^*) + \ell_1(\hat{\mathbf{f}}_t, \mathbf{f}_t^*), \\ \mathcal{L}_{\text{struct}} &= \ell_1(\hat{D}_t, D_t^*) + \ell_1(\hat{P}_t, P_t^*) + \ell_{\text{normal}}(\hat{P}_t, P_t^*), \\ \mathcal{L}_{\text{view}} &= \sum_{i=1}^{N} \ell_{\text{proj}}(\hat{P}_t; D_i^*, \mathbf{p}_i^*). \end{aligned} \tag{27}$$

Here, $\hat{\mathbf{t}}_t$, $\hat{\mathbf{R}}_t$, and $\hat{\mathbf{f}}_t$ denote the predicted translation, rotation matrix, and fields of view; d_R is the rotation distance. D_i^* and $\mathbf{p}_i^*$ are the supervisory depth and camera annotations for reference view i. Dense terms are evaluated over valid pixels, while ℓ_{proj} retains valid in-frame, depth-consistent projections.

For a valid index set $\mathcal{V}$, the notation $\ell_1(A, B) = |\mathcal{V}|^{-1} \sum_{u \in \mathcal{V}} \|A(u) - B(u)\|_1$ denotes the mean valid-element residual. The rotational term uses the sign-invariant angular distance

$$d_R(\hat{\mathbf{R}}, \mathbf{R}^*) = \arccos\left(\text{clip}\left(\frac{\text{tr}(\hat{\mathbf{R}}\mathbf{R}^{*\top}) - 1}{2}, -1, 1 \right) \right). \tag{28}$$

Surface normals are obtained from the cross product of horizontal and vertical point-map differences on valid 2×2 neighborhoods and normalized to unit length. The normal term averages $1 - \langle \hat{\mathbf{n}}, \mathbf{n}^* \rangle$ over valid neighborhoods, using the same difference orientation for both point maps. For cross-view supervision, $\hat{P}_t(u)$ is transformed into reference camera i, projected using $\mathbf{K}_i^*$, and compared with the sampled reference depth D_i^*. Only valid, in-frame, depth-consistent projections contribute to ℓ_{proj}; the reference-view contributions are summed as in Equation 27.

D.6 Reference Render Cache and Generation Conditions

RGG caches at most 12 reference views. If more are available, up to six are first selected by target relevance, combining viewing-direction similarity, camera-center proximity, and target-frustum support with weights $0.45/0.30/0.25$. The remaining slots are filled by farthest-point sampling over normalized camera centers and viewing directions. This selection balances target relevance with baseline diversity. Each cache entry contains the real reference I_i, its point-cloud re-rendering $\widetilde{I}_i$, raw coverage M_i, and camera rays.

Images are encoded by a VAE with an eight-fold spatial reduction, and latent patch size two makes one generation token correspond to a 16×16 image region. Coverage is average-pooled to the VAE grid; each generation token stores the local 2×2 values in top-left, top-right, bottom-left, bottom-right order and maps them from $[0, 1]$ to $[-1, 1]$. A learned linear map $C_{\text{cov}} : \mathbb{R}^4 \to \mathbb{R}^d$ converts this code to the generation space. The map is initialized at zero when adapting a pre-coverage checkpoint, preserving the original generator at initialization while allowing coverage semantics to emerge during Stage 3.

For pixel ray direction $\mathbf{d}$ and camera center $\mathbf{o}$, the spatial ray is the Plücker coordinate $[\mathbf{d}; \mathbf{o} \times \mathbf{d}]$. A stride-eight ray encoder followed by latent-patch projection aligns ray tokens with generation tokens. Reference and target generation embeddings are

$$\begin{aligned} \mathbf{e}_i^{\text{ref}} &= \frac{1}{2} C_{2z}([\mathcal{E}(I_i), \mathcal{E}(\widetilde{I}_i)]) + \mathbf{e}_{\text{pos}} + \mathbf{e}_{\text{ray},i} + C_{\text{cov}}(M_i), \\ \mathbf{e}_t(\tau) &= \frac{1}{2} C_{2z}([\mathbf{z}_\tau, \mathcal{E}(\widetilde{I}_t)]) + \mathbf{e}_{\text{pos}} + \mathbf{e}_{\text{ray},t} + C_{\text{cov}}(M_t) + \mathbf{e}_\tau. \end{aligned} \tag{29}$$

All selected reference blocks precede the target block in the generation sequence. Reference and target coverage codes are projected in one batched call and then split by token range, which keeps the distributed module-call graph identical across ranks. The target coverage and ray conditions are reintroduced at every flow-integration step.

D.7 Optimization Settings

All stages use AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$, zero weight decay, gradient clipping at 1.0, bfloat16 computation, fully sharded data parallelism, and 16 A100 GPUs. Dynamic packing groups variable-frame samples by token count. Table 11 summarizes the stage-specific settings.

Table 11: Optimization and input settings. The stated frame counts follow the main training configuration.

Setting	Stage 1	Stage 2	Stage 3
Primary objective	Answer likelihood	Target camera and structure	Render-guided flow matching
Learning rate	10^{-5}	10^{-5}	10^{-5}
Schedule	700-step warm-up, then constant	Constant	Constant
Training steps	20,000	25,000	30,000
Number of input references	1–32	3–16	1–32
Maximum packed sample budget	16,384	25,000	30,000 tokens
Expected packed budget	32,768	27,000	32,768 tokens
Hard packed-batch limit	36,864	31,000	36,864 tokens
Trainable functional path	GAR and $\mathcal{U}$	LCG, $\mathcal{U}$, and interfaces	RGG and generation interfaces

In Stage 2, the first 2,000 steps adapt the learnable target tokens, understanding pathway, and language–geometry interfaces while keeping the spatial encoder and camera/dense heads fixed. The geometry pathway then joins optimization. The objective follows Equation 14, with $(\lambda_{\text{pose}}, \lambda_{\text{view}}, \lambda_{\text{struct}}) = (0.5, 0.3, 0.2)$ weighting camera, cross-view, and dense-structure supervision, respectively. In Stage 3, GAR/LCG and the VAE remain fixed, while the generation expert and its dual-latent, ray, output, and coverage interfaces are optimized at the same learning rate. We use at most 12 reference cache entries, at most 800K fused points, and no render-hole filling.

Model size and sampling. TinkerCube has approximately 15B parameters, comprising a 7B understanding expert, a 7B generation expert, and a 1B geometry expert. Each novel view is synthesized using 50 flow-integration steps.

E Training Procedures

For completeness, Algorithms 1– 3 summarize the three stages at the level of functional modules. They intentionally share the notation of Section 3.

Algorithm 1 Stage 1: spatial–language alignment.

Require: Ordered frames $\mathcal{I}$, question x, and answer y
1: Sample $\mathcal{J} = (j_1 < \cdots < j_N)$ by uniform-with-jitter sampling
2: $\mathbf{V}_i \leftarrow C_v(\mathcal{F}_v(I_{j_i}))$ for each selected view
3: $(\mathbf{g}_i^{\text{sc}}, \mathbf{g}_i^{\text{cam}}, \mathbf{G}_i^{\text{reg}}) \leftarrow \mathcal{F}_s(\mathcal{I}_{\mathcal{J}})$
4: **for** $i = 1, \ldots, N$ **do**
5: $\quad \mathbf{B}_i \leftarrow [e_{\text{soi}}; \mathbf{V}_i; e_{\text{eoi}}; e_{\text{s3d}}; C_{\text{sc}}(\mathbf{g}_i^{\text{sc}}); C_{\text{cam}}(\mathbf{g}_i^{\text{cam}}); C_{\text{reg}}(\mathbf{G}_i^{\text{reg}}); e_{\text{e3d}}]$
6: **end for**
7: $\mathbf{H}^u \leftarrow \mathcal{U}([\mathbf{B}_1; \cdots; \mathbf{B}_N; x])$
8: $\mathcal{L}_{\text{S1}} \leftarrow -\sum_{j \in \mathcal{Y}} \log p(y_j \mid y_{<j}, \mathbf{H}^u)$
9: Update GAR and $\mathcal{U}$ with $\nabla \mathcal{L}_{\text{S1}}$
Ensure: A reasoning space aligned with view-consistent spatial evidence

F Design Clarifications

Beyond a sequential composition of pretrained modules. The three functional modules exchange task-relevant representations at multiple levels. GAR converts camera and cross-view reg-

Algorithm 2 Stage 2: language-conditioned target-view structure learning.

Require: References $\mathcal{I}_r$, motion m_t, caption c_t, and labels $(\mathbf{p}_t^*, D_t^*)$
Sample references and target with $p_{\mathrm{ref}}(\mathrm{local}, \mathrm{uniform}, \mathrm{wide})$
$\mathbf{p}_t^* \leftarrow \mathrm{Pose9}(\mathbf{T}_t^{\mathrm{wc}} \mathbf{T}_{r_1}^{\mathrm{cw}})$
$m_t \leftarrow \mathrm{AugmentMotion}(m_t)$
$\mathbf{H}^u \leftarrow \Phi_{\mathrm{GAR}}(\mathcal{I}_r, [m_t; c_t])$
$\mathbf{Q}_m \leftarrow \mathrm{Pool}_{32}(\mathbf{H}^u[\mathcal{M}])$
$\mathbf{Q}_c \leftarrow C_c(\mathrm{Pool}_{32}(\mathbf{H}^u[\mathcal{C}]))$
$\mathbf{M}_r \leftarrow \mathrm{Mem}(\mathcal{F}_s(\mathcal{I}_r))$
$\mathbf{p}_t^0 \leftarrow \mathcal{P}_{\mathrm{anchor}}(C_m(\mathbf{Q}_m), \mathbf{M}_r)$
$(\mathbf{Q}_p, \mathbf{Q}_\rho) \leftarrow \mathrm{PoseRay}(\mathbf{p}_t^0)$
for $\ell \in \mathcal{L}_c$ **do**
    $\mathbf{T}_t^{(\ell+1)} \leftarrow \mathbf{T}_t^{(\ell)} + \frac{1}{\sqrt{6}} \Delta_\ell(\mathbf{T}_t^{(\ell)}; \mathbf{Q}_c, \mathbf{Q}_p, \mathbf{Q}_\rho)$
end for
$(\hat{\mathbf{p}}_t, \hat{D}_t) \leftarrow \mathrm{Decode}(\mathbf{T}_t, \mathbf{M}_r, \mathbf{Q}_c)$
$\hat{P}_t(u) \leftarrow \pi_{\hat{\mathbf{p}}_t}^{-1}(u, \hat{D}_t(u))$
$\mathcal{L}_{\mathrm{S2}} \leftarrow \lambda_{\mathrm{pose}} \mathcal{L}_{\mathrm{pose}} + \lambda_{\mathrm{struct}} \mathcal{L}_{\mathrm{struct}} + \lambda_{\mathrm{view}} \mathcal{L}_{\mathrm{view}}$ on valid masks
Update the active LCG and language-conditioning parameters according to Table 11
Ensure: A target camera and coordinate-consistent depth and point map

Algorithm 3 Stage 3: coverage-aware render-guided generation.

Require: Connected references $\mathcal{I}_r$, target I_t, motion m_t, and caption c_t
$(\hat{\mathbf{p}}_t, \hat{D}_t, \hat{P}_t) \leftarrow \Phi_{\mathrm{LCG}}(\mathcal{I}_r, m_t, c_t)$ with LCG fixed
Estimate $(\mathbf{p}_i, D_i) \leftarrow \mathcal{F}_s(\mathcal{I}_r)_i$ with GAR/LCG fixed
Fuse valid reference observations into $\mathcal{P}_r$
Select cache indices $\mathcal{S} \subseteq \mathcal{J}_r$ with $|\mathcal{S}| \leq 12$
for $i \in \mathcal{S}$ **do**
    $(\widetilde{I}_i, M_i) \leftarrow \mathcal{R}(\mathcal{P}_r, \mathbf{p}_i)$
    $\mathbf{e}_i^{\mathrm{ref}} \leftarrow \mathrm{DualLatent}(I_i, \widetilde{I}_i) + \mathbf{e}_{\mathrm{ray},i} + \mathbf{e}_{\mathrm{cov},i}$
end for
$(\widetilde{I}_t, M_t) \leftarrow \mathcal{R}(\mathcal{P}_r, \hat{\mathbf{p}}_t)$
Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ and $\tau \sim U(0, 1)$
$\mathbf{z}_0 \leftarrow \mathcal{E}(I_t)$, $\mathbf{z}_\tau \leftarrow (1-\tau)\mathbf{z}_0 + \tau\boldsymbol{\epsilon}$, $\mathbf{v}^* \leftarrow \boldsymbol{\epsilon} - \mathbf{z}_0$
$\mathbf{e}_t(\tau) \leftarrow \mathrm{DualLatent}(\mathbf{z}_\tau, \widetilde{I}_t) + \mathbf{e}_{\mathrm{ray},t} + \mathbf{e}_{\mathrm{cov},t} + \mathbf{e}_\tau$ using $\hat{\mathbf{p}}_t$ for rays and coverage
$\mathcal{C}_t \leftarrow [\mathcal{C}_{\mathrm{sem}}; \{\mathbf{e}_i^{\mathrm{ref}}\}_{i \in \mathcal{S}}; \mathbf{e}_t(\tau)]$
$\mathcal{L}_{\mathrm{S3}} \leftarrow \mathbb{E}[\|\mathbf{v}_{\mathcal{G}}(\mathbf{z}_\tau \mid \mathcal{C}_t, \tau) - \mathbf{v}^*\|_2^2]$
Update RGG with $\nabla \mathcal{L}_{\mathrm{S3}}$
Ensure: A render-guided generator respecting scene structure and viewpoint

ister states into the same contextual space that encodes motion and caption text. LCG then reads those contextual states to construct a target-view representation inside the spatial encoder, rather than predicting geometry from an isolated text encoder. RGG finally consumes the cameras and depths produced by the spatial path as an explicit point cloud, while reusing GAR semantics and LCG camera rays as generation conditions. Consequently, language affects the target camera and structure before rendering, and explicit structure constrains the subsequent image synthesis. The matched-data and composition comparisons in Section I.2 support the benefit of these interactions alongside progressive training and component specialization.

Task-aligned conditioning. Reference images supply spatial tokens, camera states, and dense scene structure. LCG combines this evidence with motion and optional content descriptions to predict target geometry. RGG then uses the predicted camera for rendering, rays, and coverage, maintaining the same conditioning flow during Stage 3 and NVS evaluation. Target images provide flow-matching supervision during training and image-scoring references during evaluation. Quantitative NVS omits content descriptions. Caption-conditioned visualizations demonstrate semantic content control; descriptions marked GT in the figures are target-derived conditions. These inter-

faces connect spatial reasoning, geometric prediction, and view synthesis through shared semantic and spatial evidence.

Coordinate-system consistency. All source cameras are converted to an OpenCV camera-to-world/world-to-camera pair before sample construction. The first ordered reference defines the canonical coordinate frame, and intrinsics are converted to horizontal and vertical FoV after image resizing. Depth, point maps, target rays, and point-cloud fusion therefore share one camera convention. Geometry-specific scoring follows Section G.

Sampling coverage and optimization stability. Local views provide dense correspondence, wide-baseline views teach extrapolation, and uniform views connect these regimes. The dual-target construction increases target diversity while independent forwards preserve a well-defined one-target objective and bounded memory use. Stage 3 further requires reference connectivity so that large motion is introduced without forming a point cloud from mutually disconnected observations. Invalid depth and camera filtering is applied before loss construction, whereas uncovered rendering regions are retained and represented by coverage tokens. The model therefore receives reliable geometric supervision and still learns to complete the regions where explicit projection is unavailable.

G Geometry Evaluation

Tables 3 and 5 evaluate LCG's target-view depth and point maps; the TinkerCube row in Table 4 reports the corresponding LCG target cameras. Reference images supply scene evidence, and language conditions describe the requested viewpoint and content following Equation 13. Geometric annotations supply the scoring reference. Table 4 summarizes camera accuracy. Table 12 summarizes LCG structural scoring and camera scoring, also used in the training ablations.

Table 12: Target-geometry scoring and conditioned-camera diagnostics.

Setting	Depth	Camera diagnostics	Point map
Datasets	Sintel, NYU-v2	Sintel, 7-Scenes	7-Scenes, ETH3D
Prediction	Target depth	Target cameras	Target point maps
Scene evidence	Shared reference observations and target-view requests		
Preprocessing	Aspect-ratio-preserving balanced 512, patch size 16		
Alignment	Per-image median scale	Sequence Sim(3) for ATE; relative poses for AUC	Scene-level correspondence Sim(3), then rigid ICP
Aggregation	Valid-pixel weighted	Pooled pairs for AUC	Equal weight per scene
Units	Dimensionless	AUC: percent; ATE: Sintel world units	Meters

Image grids and valid regions. Balanced 512 allocates approximately 1,024 patches while preserving aspect ratio. Images outside the aspect-ratio interval $[0.5, 2.0]$ are centrally cropped to this interval; resized views are symmetrically padded to a common sequence canvas. Depth predictions are restored to the scoring grid, while point-map evaluation maps annotations and intrinsics to the model grid. NYU-v2 uses finite positive depth in its validation split; Sintel uses final RGB and positive depth below 70. Point-map scoring uses shared valid regions and finite positive depths, retaining all valid points independently of predicted confidence.

Depth. Let $\widetilde{D}$ denote the scale-aligned prediction under Table 12. Sintel predictions are capped at 70. Relative error averages $|\widetilde{D} - D^*|/D^*$, and $\delta_{1.25}$ counts pixels satisfying $\max(\widetilde{D}/D^*, D^*/\widetilde{D}) < 1.25$. Scores pool valid pixels across the evaluation set.

Camera scoring. The TinkerCube row in Table 4 uses the following ATE and AUC definitions. The AUC scores in Tables 7, 8, and 15 compare training variants under the same motion and content conditions. They characterize target-camera consistency within the LCG task. Predicted cameras are expressed in the common reference frame. Relative-pose evaluation enumerates unordered target-view pairs and combines rotation angle e_R with translation direction error e_t through

$e = \max(e_R, e_t)$:

$$\mathrm{AUC@}\tau = \frac{100}{\tau} \int_0^\tau F_e(v)\, dv, \tag{30}$$

where F_e is the pooled error distribution and angles are in degrees. Translation direction uses the absolute cosine; zero-translation pairs receive $90°$. ATE measures camera-center deviation after the sequence-level alignment in Table 12, in Sintel world units; the similarity fit uses Umeyama's method.

Point maps. Predicted depth and camera parameters yield points through Equation 26. Scene registration follows Table 12, using point-to-point ICP with a 0.1 m correspondence threshold and at most two million fixed-sampled correspondences for the initial similarity fit. For aligned predictions $\mathcal{P}$ and reference points $\mathcal{Q}$,

$$\mathrm{Acc} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \min_{q \in \mathcal{Q}} \|p - q\|_2, \qquad \mathrm{Comp} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \min_{p \in \mathcal{P}} \|q - p\|_2. \tag{31}$$

Both distances are reported in meters and averaged equally across scenes. Together, they quantify surface precision and coverage under a common geometric alignment.

H Novel-View Synthesis Evaluation Protocol

H.1 Clip Construction and Reference Updates

We use a unified trajectory-based protocol for DL3DV and RealEstate10K. Each clip draws 25 frames from one scene: frame 0 is the sole real reference, and frames 1–24 define the requested target viewpoints. We use a fixed random seed to select 300 valid clips per dataset; all methods share the same scene list, sampled frame indices, and requested viewpoints. A clip is valid only if all 25 RGB frames and camera poses are readable.

For each target, motion text specifies the requested translation, orientation, and field-of-view change. LCG predicts $\hat{\mathbf{p}}_t$ from this instruction and the reference context; RGG uses the prediction for geometric projection, camera rays, and view synthesis. Content descriptions are omitted in quantitative NVS, and target images provide the image-scoring reference. All methods share the requested trajectory and reference-update schedule, starting from one real reference and refreshing generated references every ten targets.

H.2 Image Alignment

The generation resolution is selected from the fixed bucket set {(320,704), (352,640), (384,608), (416,560), (448,512), (480,480), (512,448), (560,416), (608,384), (640,352), (704,320)}. The bucket closest to the reference aspect ratio is used for the entire clip. Reference and target images undergo the same bicubic resize-to-cover followed by center crop; camera fields of view are adjusted for this transform. A generated image and its target are therefore compared at identical resolution and pixel support, without post-hoc stretching.

I Additional Quantitative Analysis

I.1 Cross-Task Performance

Spatial reasoning. Tables 1 and 2 show leading SPAR, ViewSpatial overall, and VSI-Bench results alongside strong open-ended scene QA. Camera-centered relations benefit from viewpoint-aware evidence, while competitive BLINK performance reflects retained visual semantics. ScanQA exact match and CIDEr, together with SQA3D exact match, extend the advantage to free-form answers. These patterns support GAR's joint use of appearance, camera, and cross-view representations.

Target geometry and control. Tables 3 and 5 characterize target structure conditioned on reference observations and target requests. Leading Sintel depth results and low NYU relative error support accurate relative structure; NYU threshold accuracy matches G^{2}VLM. ETH3D results combine surface precision and completeness. Table 4 summarizes camera accuracy across geometry systems and TinkerCube. LCG's trajectory and relative-pose scores characterize the camera interface used to organize its target depth and point maps. Together, target-token interaction and residual decoding connect language-guided motion to the geometry used by RGG's target-view rendering.

View synthesis. Table 6 shows strong image fidelity across both datasets. TinkerCube leads PSNR on DL3DV and RealEstate10K and achieves the highest SSIM on RealEstate10K. The gains over BAGEL support conditioning the generator on target projection, reference appearance, rays, and coverage. Their complementary roles are further evaluated through matched-data and retrained component comparisons below.

I.2 Matched Training and Component Comparisons

All comparisons use matched evaluation inputs, preprocessing, and scoring. The two Stage 1 adaptations share initialization, instruction data, sampling, optimizer, and 20k training steps. Stage 2 variants start from the same Stage 1 checkpoint and share data, 25k steps, and the initial adaptation period, varying the specified trainable pathway. GAR and RGG component variants are retrained under matched initialization, data, and optimization settings with the indicated component removed. For the full model and one-shot fusion, pretrained components, three-stage data, and training budgets are matched. These controls connect the following results to the corresponding representational and optimization choices.

Adaptation of the multimodal backbone. Table 13 compares the original BAGEL backbone with the complete model. Improvements span image reasoning, video understanding, and open-ended answers, showing broad benefits from spatial adaptation. Joint gains in ScanQA EM, BLEU-4, and CIDEr reflect improved answer correctness and agreement with reference descriptions, while BLINK demonstrates compatibility with general visual semantics.

Table 13: Original backbone and TinkerCube across spatial reasoning tasks. VS denotes ViewSpatial; VSI denotes VSI-Bench.

Model	BLINK	SPAR	VS	VSI	ScanQA EM	ScanQA BLEU-4	ScanQA CIDEr	SQA3D EM
BAGEL	62.5	39.3	42.0	26.7	4.8	1.2	30.3	46.8
TinkerCube	70.8	65.6	56.2	61.4	31.5	19.5	113.4	61.5

Geometry-aware reasoning with matched instructions. Table 14 separates instruction adaptation from the additional contribution of geometry-aware inputs. Training BAGEL on the same Stage 1 data improves spatial performance, and GAR provides further gains on every reported metric. ViewSpatial and VSI-Bench support the value of viewpoint and cross-frame structure, while ScanQA and SQA3D extend the benefit to scene-level language decoding. Together with improved BLINK, the results support complementary semantic and geometric representations.

Table 14: Matched Stage 1 training. Checks indicate spatial instruction adaptation and geometry tokens; all metrics are percentages.

Model	S1 data	Geom. tokens	BLINK	SPAR	VS	VSI	ScanQA EM	SQA3D EM
BAGEL	×	×	62.5	39.3	42.0	26.7	4.8	46.8
BAGEL + S1	✓	×	64.7	55.8	48.9	51.6	24.3	50.8
TinkerCube, S1	✓	✓	66.9	60.4	52.7	55.2	27.1	55.6

Joint semantic–geometric optimization. Table 15 shows how Stage 2 couples geometry learning with the understanding pathway. Updating geometry with the understanding expert fixed improves camera prediction; updating the understanding pathway brings additional reasoning gains. Joint optimization achieves the strongest results in both task families, supporting coordinated adaptation of semantic states and geometric representations. Table 7 further shows that Stage 3 translates these capabilities into improved NVS while retaining the preceding reasoning and camera performance.

Table 15: Stage 2 optimization from the same Stage 1 checkpoint. UND and GEO indicate updates to the understanding expert and geometry pathway.

Configuration	S2	UND	GEO	SPAR	VS	Sintel AUC@30
S1 checkpoint	×	×	×	60.4	52.7	8.0
Frozen understanding	✓	×	✓	60.4	52.7	66.5
Frozen geometry	✓	✓	×	63.7	55.8	49.7
Full S2	✓	✓	✓	65.6	56.2	75.4

GAR representations. The retrained variants in Table 9 show complementary contributions from all token groups. Appearance tokens provide the largest benefit, grounding object identity and visual detail. Among spatial tokens, register states contribute most consistently to cross-task performance, matching their role in cross-view structure aggregation. Camera states provide dedicated viewpoint information, and scene tokens add a compact global summary. Their combined advantage supports role-aware interaction within the language context.

RGG conditions. Table 10 shows that target rendering provides the strongest generation constraint across image and camera metrics. Reference-cache conditioning further improves appearance and viewpoint consistency. Rays specify observation direction, and coverage describes the spatial support of the projection. The complete configuration performs best across the reported ablation metrics, supporting the joint use of geometry, appearance, direction, and visibility. The Stage 3 interface comparison in Table 8 additionally supports adapting these conditions to the generation expert.

I.3 Language-Conditioned Camera Control

We evaluate partial motion instructions on held-out requests without content captions or target RGB. The instructions retain directional or compositional constraints while omitting continuous camera values; paraphrases vary wording while preserving the intended motion. Because these requests leave some viewing choices unspecified, we evaluate whether the realized camera satisfies the stated constraints. We compare the initial LCG anchor with the camera obtained after target-token interaction and residual refinement under identical reference observations and motion instructions.

Table 16: Camera control with partial, compositional, and paraphrased instructions. Scores report constraint satisfaction (%).

Method	Trans. ↑	Rot. ↑	Combined ↑	Paraphrase ↑	Valid ↑
LCG anchor	89.7	88.8	84.6	85.9	100.0
Full LCG	94.4	93.6	90.2	91.8	100.0

Table 16 evaluates satisfaction of partial, compositional, and paraphrased motion constraints. Translation and rotation scores assess the corresponding directional requirements, the combined score requires all specified constraints to hold, and the paraphrase score measures control consistency under alternative wording. Full LCG consistently improves over its initial anchor across all four categories. These gains show that reference-conditioned target-token interaction and residual decoding progressively refine the intended viewpoint while preserving valid camera outputs.

I.4 Module Composition and Matched-Data Fusion

The frozen composition connects BAGEL to the outputs of frozen VGGT-Ω through geometry-derived context for reasoning and point-cloud renderings for generation. The matched-data one-shot

fusion uses the same pretrained components and three-stage training budget as TinkerCube, with shared geometry projection, initial target-token conditioning, and a unified generation-conditioning interface. TinkerCube instead uses role-aware projection and repeated target-view modulation. Evaluation inputs and protocols are matched for each task.

Table 17: NVS under the shared trajectory and reference-update protocol.

Dataset	System	PSNR↑	LPIPS↓	SSIM↑
DL3DV	BAGEL	12.488	0.430	0.298
	Frozen composition	14.723	0.348	0.374
	One-shot fusion	15.681	0.289	0.462
	TinkerCube	16.218	0.263	0.503
RealEstate10K	BAGEL	14.378	0.380	0.498
	Frozen composition	16.583	0.291	0.547
	One-shot fusion	18.793	0.228	0.642
	TinkerCube	19.419	0.204	0.682

Table 17 shows consistent gains on both datasets. Geometric rendering improves the frozen generator, and matched-data fusion strengthens image quality. TinkerCube further improves the listed fidelity metrics over these composition baselines, supporting coordinated conditioning on reference appearance, projected structure, rays, and coverage for faithful view synthesis.

Table 18: Spatial reasoning under pretrained composition and matched-data fusion. VS and VSI denote ViewSpatial and VSI-Bench.

System	BLINK	SPAR	VS	VSI	SQA3D EM
BAGEL	62.5	39.3	42.0	26.7	46.8
Frozen composition	63.4	47.8	46.9	38.2	50.4
One-shot fusion	68.3	63.1	54.8	58.7	59.1
TinkerCube	70.8	65.6	56.2	61.4	61.5

Table 18 shows progressively stronger spatial reasoning from frozen composition to trained fusion and TinkerCube. Frozen composition exposes geometric context to the semantic backbone, while joint adaptation improves the interaction of language and spatial representations. TinkerCube further improves all listed reasoning metrics under the matched training budget, supporting its coordinated interfaces beyond one-shot fusion across image reasoning, video understanding, and open-ended scene questions.