

OmniCam: Omni-Camera Trajectory Generation via Geometry-Grounded Pose Token Learning

Zhenyang Liu^{1,2,3} Chenjie Cao³ Yisu Zhang^{4,3} Xuhui Zuo³
 Xiangyang Xue^{1,†} Yanwei Fu^{1,2,‡} Tengfei Wang^{3,†} Chunchao Guo³

¹Fudan University ²Shanghai Innovation Institute ³Tencent Hunyuan ⁴Zhejiang University

Project Page: ZhenyangLiu.github.io/OmniCam-Website

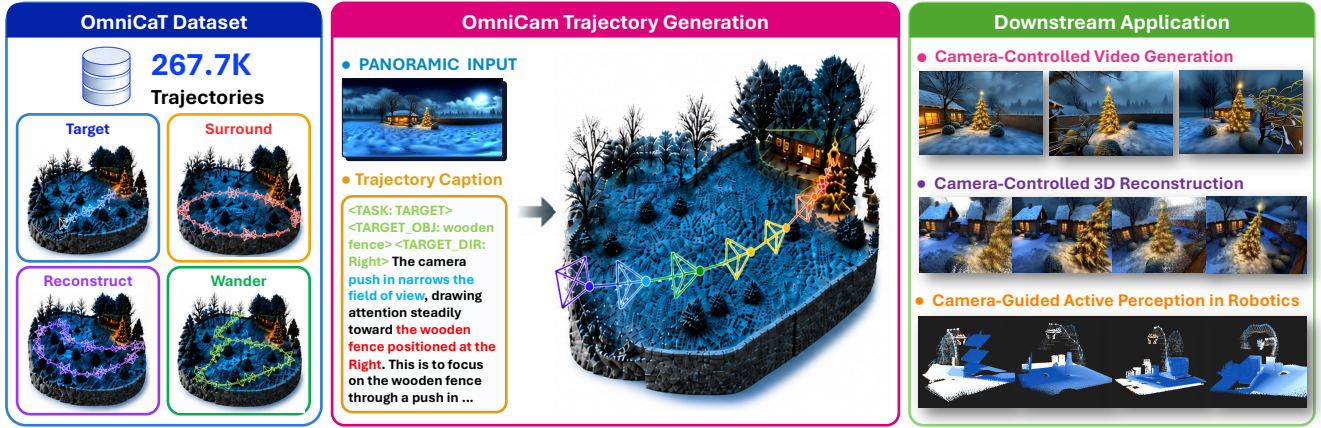


Figure 1. **Overview of OmniCam.** Given a single panorama and textual trajectory descriptions, OmniCam generates spatially aware camera trajectories (SE(3) pose sequences). OmniCam supports four camera behaviors that enable various downstream applications such as camera-controlled video generation, 3D reconstruction, and active perception in robotics.

Abstract

Camera trajectories control viewpoint changes in video generation, scene reconstruction, and robotic perception. Generating them from language requires both scene geometry and target-aware framing. We introduce **OmniCam**, an autoregressive model that generates camera pose sequences from a single panorama and textual trajectory descriptions. Its geometry-grounded pose token learning combines three components: a panoramic point-cloud encoder for omnidirectional geometric context; hybrid absolute-rotation and relative-translation tokenization with temporally consistent quaternion signs; and separate geometric and semantic conditioning streams with an explicit 3D target anchor. We also construct **OmniCaT**, containing 267,700 trajectories across four camera behaviors. On the reported OmniCaT evaluation, OmniCam reduces trajectory errors by 28–47% and collision rate by 65.8% relative to GenDoP retrained on OmniCaT. Against the best baseline for each metric, the

ATE and collision reductions are 43.0% and 62.3%, respectively. Component ablations support the use of geometric and target-aware conditioning, while downstream experiments examine camera-controlled video generation and robotic active perception.

1. Introduction

Camera trajectories serve as a control interface for spatial-intelligence systems, including world models [1], video generation [13, 31], and robotics [2, 16, 21]. Specifying a trajectory requires deciding *where to move*, *how to orient*, and *what to attend to*. A prescribed orbit or linear sweep may fail when it intersects scene geometry or loses the intended target. We study generating scene-aware camera trajectories from a panoramic observation and a textual instruction (Figure 1).

Recent trajectory generators [7, 18, 23, 34] motivate three design questions. (i) *Scene coverage*: perspective RGBD inputs [34] provide limited angular coverage, while geometry-based planners require explicit spatial representa-

[†]Project Lead. [‡]Corresponding authors.

tions [29, 30]. A panorama provides observations in all directions, although occluded surfaces remain unknown. (ii) *Pose serialization*: quantization trades precision against token vocabulary size. Quaternion sign ambiguity ($q \equiv -q$), coordinate scale, and the accumulation of relative increments can affect the resulting learning problem. (iii) *Conditioning*: geometric free-space information and language-referenced targets serve different roles in planning. Providing dedicated pathways for these cues is a useful architectural choice whose value should be tested empirically. We use *geometry-grounded pose token learning* to describe a design that connects geometric perception, pose serialization, and target-aware conditioning.

OmniCam implements this design with a panoramic point-cloud encoder, hybrid pose tokenization, and a dual-branch conditioning architecture. The point cloud, reconstructed from estimated panoramic depth, provides omnidirectional but partial geometry. Rotations are encoded absolutely with sequential quaternion sign alignment; translations are encoded as increments in a scene-normalized coordinate system. Sign alignment selects consistent representatives of the quaternion double-cover, while quantization remains lossy. Absolute rotations avoid composition of rotation-quantization residuals; relative translations trade a smaller per-step range against accumulated displacement error. We analyze these distinctions in Section I and compare the encodings empirically.

Separate geometric and semantic pathways supply the decoder with structural and target-related features. An attention-derived 3D target centroid is injected through gated cross-attention to provide a spatial anchor for object-directed trajectories. The geometry pathway does not directly receive text; this routing does not imply statistical independence or guarantee the absence of interference after fusion. At inference, a format constraint restricts termination to complete pose-token groups.

Training requires trajectories paired with scene observations and instructions. Existing collections [24, 33, 34, 37] cover recorded object, scene, and cinematic motions. We construct **OmniCaT** by recovering geometry and semantic targets from panoramas and synthesizing four behaviors: Target, Surrounding, Reconstruction, and Wandering. The dataset contains 267,700 trajectories with hierarchical descriptions of task, target, and motion. Because supervision is generated on estimated geometry, the learned model may inherit reconstruction errors and planner preferences.

Our contributions are:

- **Panorama-conditioned trajectory generation.** OmniCam combines explicit point-cloud features, dedicated geometric and semantic pathways, and a 3D target anchor to generate camera trajectories from panoramic imagery and language.
- **Hybrid pose serialization.** We combine absolute rota-

tions, relative translations, and consistent quaternion signs; provide qualified error and representation analyses; and compare the choices through ablations.

- **OmniCaT and empirical evaluation.** We introduce a 267,700-trajectory dataset spanning four behaviors. The reported OmniCaT results improve on retrained baselines, with component ablations and downstream video and robotic-perception evaluations.

2. Related Work

Camera Trajectory Generation. Early approaches rely on optimization-based planning [3, 11, 25, 28] or learning-based control [4, 10, 15, 17]. Recent methods integrate camera movement with scene dynamics: CCD [18] employs a diffusion model with text and keyframe conditioning in character-centric coordinates; E.T. [7] incorporates character trajectories in global coordinates; Director3D [23] adopts a DiT-based framework on multi-view data; Hou et al. [14] predict next-frame motion autoregressively for aerial videography; GenDoP [34] combines text with RGBD inputs in an autoregressive framework. These methods use different task-specific inputs and trajectory representations. OmniCam studies panoramic point-cloud conditioning and explicit target anchoring, together with the effect of absolute and relative pose serialization in an autoregressive generator.

Camera Trajectory Datasets. Existing trajectory datasets remain limited in spatial diversity. MVIImgNet [33], RealEstate10K [37], and DL3DV-10K [24] provide calibrated trajectories but focus on simple motions around stationary objects or within static environments. CCD [18] and E.T. [7] offer character-centric tracking data that does not generalize to unconstrained spatial navigation. DataDoP [34] pairs trajectories with depth maps and text for cinematic applications, yet relies on perspective-view depth covering only a limited field of view. None of these resources simultaneously provides omnidirectional scene geometry, diverse spatial behaviors, and hierarchical textual guidance. OmniCaT fills this gap by grounding trajectory generation in panoramic point-cloud geometry and pairing each trajectory with structured language annotations across four distinct navigation behaviors.

Pose Representation Learning. Rotation representation [19, 22] has a well-studied impact on learning stability. Zhou et al. [38] show that quaternions and Euler angles suffer from discontinuities under the topology of $SO(3)$ and propose continuous 6D representations for single-frame rotation regression. Subsequent work considers sequential settings such as human motion prediction [5, 8] and robotic manipulation [21, 26]. OmniCam uses sequential quaternion sign alignment to reduce inconsistent discrete labels for equivalent rotations, combined with relative translation. This convention does not provide a globally continuous rotation parameterization or an injective finite quantizer.

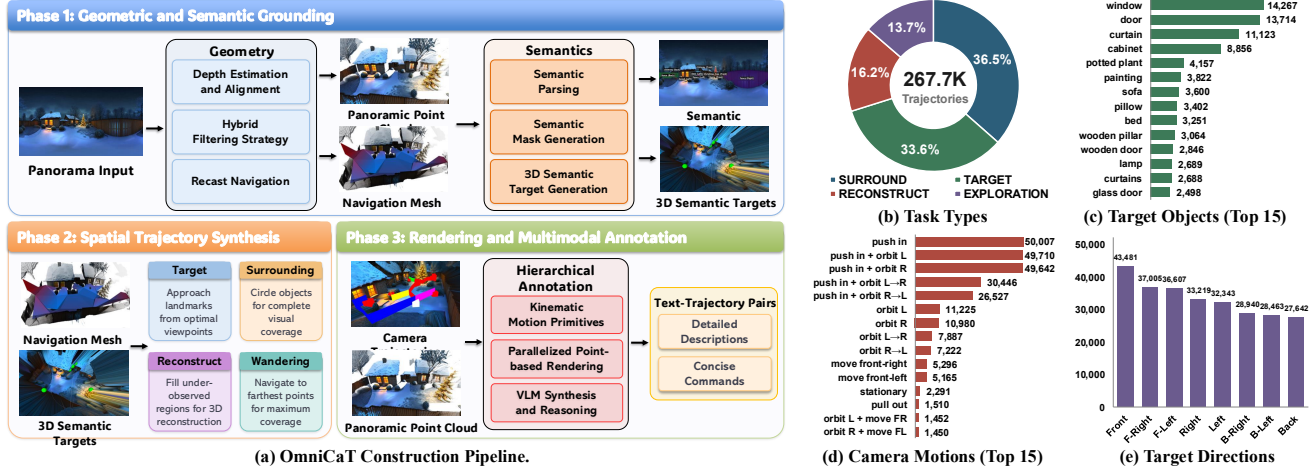


Figure 2. **OmniCaT Dataset Overview.** *Left:* (a) Construction pipeline comprising (1) Geometric and Semantic Grounding to extract nav-meshes and 3D targets, (2) Spatial Trajectory Synthesis to generate diverse camera behaviors, and (3) Rendering and Multimodal Annotation to produce text-trajectory pairs. *Right:* Dataset statistics showing (b) task distribution, (c) top object categories, (d) motion primitive diversity, and (e) target direction distribution.

3. The OmniCaT Dataset

Collecting camera trajectories with controlled behavior and target coverage is costly. **OmniCaT** synthesizes trajectories on geometry reconstructed from panoramic images, enabling controlled variation of spatial behaviors. Feasibility is assessed against the reconstructed geometry and does not by itself guarantee collision avoidance in the physical scene.

3.1. Dataset Construction

As illustrated in Figure 2, the pipeline consists of three phases. (1) *Geometric and semantic grounding*: panoramic depth maps are aligned into a global point cloud P^{pan} [32], filtered for outliers, and augmented with semantic landmarks and 3D volumetric masks using language-assisted annotation and segmentation models [6, 20, 27]; a navigation mesh is then constructed to define the traversable space. (2) *Spatial trajectory synthesis*: four categories of camera behaviors (Target, Surrounding, Reconstruction, and Wandering) are generated on the navigation mesh, with waypoints and orientations refined via spline and spherical linear interpolation. (3) *Rendering and multimodal annotation*: a point-based renderer produces video sequences from the planned poses, and a hierarchical annotation scheme derives kinematic motion primitives from camera extrinsics before a vision-language model synthesizes motion-aware textual instructions. Full implementation details, including depth alignment optimization, NavMesh engineering, and trajectory planning heuristics, are provided in Section A.

3.2. Dataset Statistics

Scale and Annotations. As summarized in Figure 2, OmniCaT comprises 267,700 trajectories across 9,998 scenes (21.7M rendered frames), categorized into four spatial behaviors: Surrounding (97,702), Target (90,065), Reconstruction (43,316), and Wandering (36,617). Each trajectory is annotated with 3D semantic targets, directional bearings, and hierarchical textual descriptions, enabling joint geometric-semantic conditioning. Full annotation details are provided in Section A.

Scope of Synthetic Supervision. The planner uses reconstructed geometry, navigation meshes, and semantic targets to generate feasible examples. OmniCam learns to predict trajectories from a panorama and text without constructing a navigation mesh or running graph search at inference. Downstream evaluation tests the usefulness of this learned mapping. A gap to the planner does not establish freedom from planner bias; evaluation on independently measured geometry and independently specified tasks is needed to assess that limitation.

4. Method

4.1. Hybrid Pose Tokenization

We serialize continuous camera parameters using scene-aware calibration and a hybrid absolute-relative representation. Quaternion sign alignment addresses equivalent positive and negative representations; finite binning still introduces quantization error.

Scene-Aware Calibration. Panoramic depth is unprojected into a filtered 3D point cloud. Scene statistics, including a

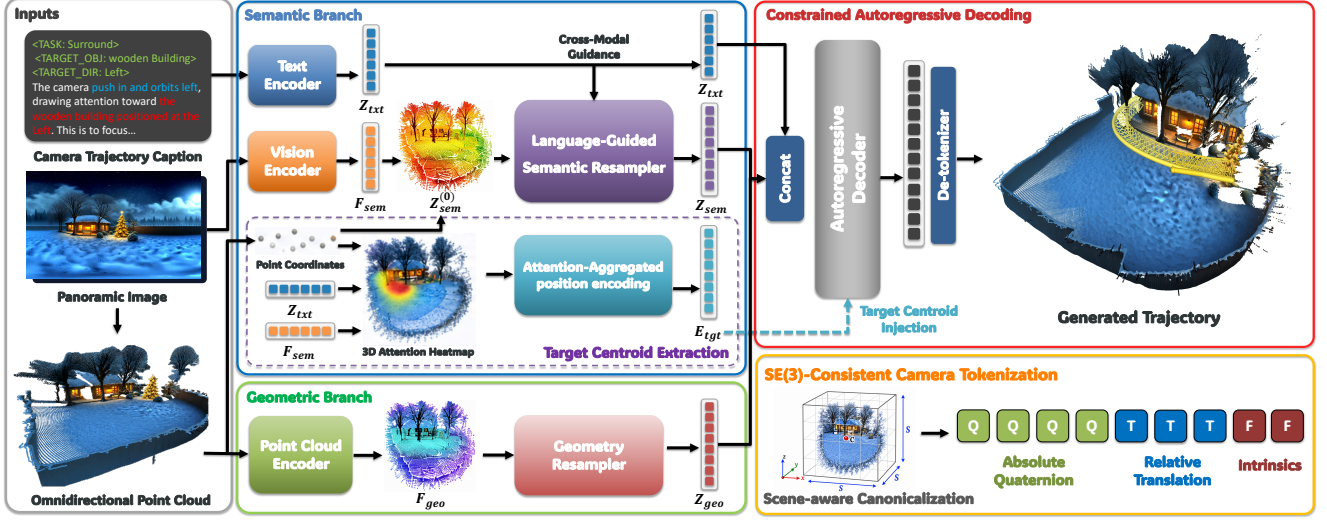


Figure 3. **OmniCam Architecture.** Three parallel branches encode the caption, panoramic image, and point cloud. A Language-Guided Resampler produces target-aware semantic tokens while a Geometry Resampler yields geometric descriptors; an attention-aggregated 3D target centroid provides an explicit spatial anchor for the decoder. The autoregressive decoder generates pose tokens from the concatenated condition, which are mapped to continuous SE(3) trajectories by a de-tokenizer. Right bottom: SE(3) tokenization with 9 tokens for each pose frame.

centroid $\mathbf{c}$ and a scale s , calibrate camera translations and the initial frame. Normalization reduces variation in coordinate magnitude across scenes but does not eliminate errors in estimated metric scale.

Hybrid Absolute-Relative Pose Encoding. Rotations are encoded as absolute unit quaternions with sequential sign alignment: each frame is aligned with the preceding aligned representative. The deterministic convention and its assumptions are stated in Proposition 2; implementation-specific tie handling must agree with that convention. Translations after the initial frame use relative increments, reducing the per-step dynamic range while allowing quantization residuals to accumulate. These components, together with normalized intrinsics, are quantized via uniform binning into per-frame token groups of 9 tokens: 4 rotation, 3 translation, 2 intrinsics (Figure 3, right bottom). The token sequence $\mathbf{S}$ is augmented with boundary tokens (**BOS**, **EOS**, **PAD**) and projected into a latent space through a learnable codebook.

Theoretical Motivation. The hybrid scheme balances different reconstruction-error tradeoffs (Section I). If each decoded rotation has angular quantization error at most ϵ , absolute encoding retains this per-frame bound, whereas integrating relative rotations permits a worst-case bound of $\min\{\pi, (T-1)\epsilon\}$ when the initial rotation is exact (Proposition 1). The value of ϵ depends on the component quantizer and decoder normalization. Relative translations can use a smaller displacement range and hence finer local resolution, but their errors can also accumulate. Proposition 2 shows that a specified initial sign rule followed by alignment to the

already aligned preceding quaternion removes dependence on the arbitrary quaternion sign when adjacent rotations differ by less than π . This resolves double-cover ambiguity in the supervision; finite quantization remains lossy and many-to-one.

4.2. Decoupled Dual-Branch Conditioning

OmniCam encodes estimated panoramic geometry and language-referenced appearance in separate pathways. The point cloud supplies omnidirectional, partial scene structure; the semantic pathway supplies target-related features. This architecture separates input routing, without assuming that the learned representations are statistically independent.

Semantic Branch (Text & Vision Encoders). The semantic branch jointly processes the trajectory caption and the panoramic image to produce target-aware features. A pre-trained text encoder maps the caption T_{txt} into textual features $\mathbf{Z}_{txt} \in \mathbb{R}^{N_{txt} \times C_{txt}}$, which also serve as cross-modal guidance for the vision pathway. In parallel, a vision encoder produces a visual feature map $\mathbf{F}_{sem} \in \mathbb{R}^{H \times W \times C_{sem}}$ from $\mathbf{I}_{pano}$. Text and the target descriptor T_{obj} guide the subsequent semantic resampling stage. Structured caption dropout varies instruction composition during training (Section H).

Geometric Branch (Point Cloud Encoder). The geometric branch receives no direct text input. At inference time, a monocular depth estimator [32] reconstructs panoramic depth from the input RGB panorama; the estimated depth is then unprojected into a 3D point cloud, voxelized, and processed by a sparse-convolution encoder combined with a Transformer encoder to yield geometric descriptors $\mathbf{F}_{geo} \in$

$\mathbb{R}^{M \times C_{geo}}$. Its features provide structural context for trajectory generation. Architectural details for both branches are provided in Section C.

4.3. Spatial Grounding via Resampling and Target Centroid Injection

This section introduces two parallel resamplers that compress heterogeneous encoder outputs (text, vision, and point cloud features) into compact prefix tokens, and an attention-aggregated 3D centroid that serves as an explicit spatial anchor.

Language-Guided Semantic Resampler. Given the visual features from the vision encoder, we first project them into the 3D point-cloud space to obtain 3D semantic features $\mathbf{Z}_{sem}^{(0)}$. The semantic resampler combines self-attention, spatial attention, and textual attention. The interaction is summarized schematically at layer l :

$$\begin{aligned} \mathbf{Z}_{sem}^{(l)} = & \text{SelfAttn}(\underbrace{\mathbf{Z}_{sem}^{(l-1)}}_{Q,K,V}) \\ & + \text{CrossAttn}(\underbrace{\mathbf{Z}_{sem}^{(l-1)}}_Q, \underbrace{[\mathbf{P}; \mathbf{F}_{sem}]}_{K,V}) \\ & + \text{CrossAttn}(\underbrace{\mathbf{Z}_{sem}^{(l-1)}}_Q, \underbrace{\mathbf{Z}_{txt}}_{K,V}). \end{aligned} \quad (1)$$

where 3D semantic features serve as the common query; self-attention captures inter-query dependencies; spatial cross-attention grounds queries in physical scene geometry aggregated by 3D point coordinates $\mathbf{P}$ concatenated with visual features $\mathbf{F}_{sem}$, and text-guided cross-attention aggregates textual guidance from $\mathbf{Z}_{txt}$. The resulting $\mathbf{Z}_{sem} \in \mathbb{R}^{M_{sem} \times C}$ serves as the semantic latent code.

Geometry Resampler. A text-agnostic geometry resampler compresses dense per-point descriptors $\mathbf{F}_{geo}$ into a fixed-length latent sequence $\mathbf{Z}_{geo} \in \mathbb{R}^{M_{geo} \times C}$ through a stack of Transformer decoder layers, providing a geometric condition without a direct text input.

Target Centroid Injection. A target centroid $\mathbf{c}_{tgt}$ supplies an explicit spatial anchor. Let a_j denote the target-attention logit associated with point $\mathbf{p}_j$ and let $\tau > 0$ be the temperature. The described soft-argmax aggregation has the form

$$\alpha_j = \frac{\exp(a_j/\tau)}{\sum_{k=1}^N \exp(a_k/\tau)}, \quad \mathbf{c}_{tgt} = \sum_{j=1}^N \alpha_j \mathbf{p}_j. \quad (2)$$

This requires point-indexed weights; attention over text tokens alone does not directly define them. The centroid is positionally encoded into an auxiliary target memory $\mathbf{E}_{tgt} \in \mathbb{R}^{M_{tgt} \times C}$. When caption dropout removes the target object tag, these tokens are masked.

The final multimodal condition $\mathbf{Z}$ is formed by concatenating the textual features with the resampled panoramic

latents:

$$\mathbf{Z} = [\mathbf{Z}_{txt}; \mathbf{Z}_{sem}; \mathbf{Z}_{geo}]. \quad (3)$$

The target token bank $\mathbf{E}_{tgt}$ serves as auxiliary memory accessed by the decoder through a gated cross-attention module, as detailed in Section 4.4.

4.4. Format-Constrained Autoregressive Decoding

An autoregressive Transformer decoder $\mathcal{D}$ generates discrete trajectory tokens conditioned on the multimodal prefix $\mathbf{Z}$ and the target token bank $\mathbf{E}_{tgt}$.

Decoder Architecture. The condition $\mathbf{Z}$ is prepended to the input sequence before the **BOS** token. Each decoder block consists of causal self-attention, gated cross-attention over $\mathbf{E}_{tgt}$, and a feed-forward network. The gate controls the contribution of the target memory. Its effect on target-directed trajectories is evaluated through the grounding ablations; per-behavior gate strength is not inferred from the aggregate results.

Format-Constrained Decoding. During inference, the **EOS** token is masked within each group of 9 tokens (rotation, translation, and intrinsics), so termination occurs only at a frame boundary. The initial frame is constrained by the calibrated origin. This is a sequence-format constraint: valid rotations require an appropriate quaternion decoding rule, and collision avoidance is assessed empirically rather than guaranteed by token masking. Top- k sampling controls the sampling support.

Training Objective. The model is optimized with a weighted cross-entropy loss $\mathcal{L}_{ce}$ over pose tokens, where rotation and translation channels receive higher weights to prioritize geometric accuracy. An auxiliary smooth ℓ_1 localization loss $\mathcal{L}_{loc}$ supervises the estimated centroid $\mathbf{c}_{tgt}$ against the annotated 3D target position: $\mathcal{L} = \mathcal{L}_{ce} + \lambda \cdot \mathcal{L}_{loc}$.

5. Experiments

5.1. Experimental Setup

Datasets. OmniCaT experiments are reported on disjoint training and evaluation splits. Exact scene-level counts and split manifests require verification. DataDoP [34] results are reported for OmniCam trained on OmniCaT without DataDoP fine-tuning. The construction of OmniCam inputs and the evaluation geometry on DataDoP must be specified before interpreting these as a controlled zero-shot comparison.

Baselines. Director3D [23] and GenDoP [34] are retrained on OmniCaT. CCD [18] and E.T. [7] use released checkpoints as out-of-domain references. Their character-centric conditioning differs from the available OmniCaT inputs, so these rows do not isolate model quality under matched conditioning.

Metrics. We report trajectory errors (ATE, FDE, and RPE-R/T, using the EVO toolkit [12] where applicable), collision against reconstructed navigation meshes, target visibility,

Table 1. Comparison with representative trajectory generation baselines on OmniCaT and the public DataDoP benchmark.

Method	Dataset	ATE ↓	FDE ↓	RPE-R ↓	RPE-T ↓	Coll. (%) ↓	Vis. (%) ↑	Angle (°) ↓
<i>Evaluation on the OmniCaT benchmark</i>								
CCD [18]	Pre-trained	2.483	3.396	3.112	0.082	36.9	12.3	32.5
E.T. [7]	Pre-trained	1.983	2.368	2.151	0.069	46.2	21.3	40.2
Director3D [23]	Pre-trained	1.828	2.942	2.309	0.052	28.4	28.1	27.2
Director3D	OmniCaT	1.522	2.648	2.187	0.049	27.6	29.3	26.5
GenDoP [34]	Pre-trained	1.865	2.951	1.681	0.044	32.6	31.5	29.1
GenDoP	OmniCaT	1.624	2.483	1.577	0.038	30.4	33.6	27.6
OmniCam (Ours)	OmniCaT	0.868	1.462	1.135	0.027	10.4	52.0	10.9
<i>Evaluation on the public DataDoP benchmark [34]</i>								
CCD [18]	Pre-trained	2.697	3.338	3.218	0.093	44.2	18.7	34.2
E.T. [7]	Pre-trained	2.568	3.142	3.086	0.084	45.9	22.5	33.8
Director3D [23]	Pre-trained	1.944	2.887	2.968	0.077	39.6	30.8	34.3
Director3D	DataDoP	1.688	2.762	2.819	0.062	41.7	32.2	29.5
GenDoP [34]	DataDoP	1.437	2.255	1.642	0.058	28.5	36.6	29.8
OmniCam (Ours)	OmniCaT	0.925	1.834	1.118	0.046	8.9	64.2	12.7

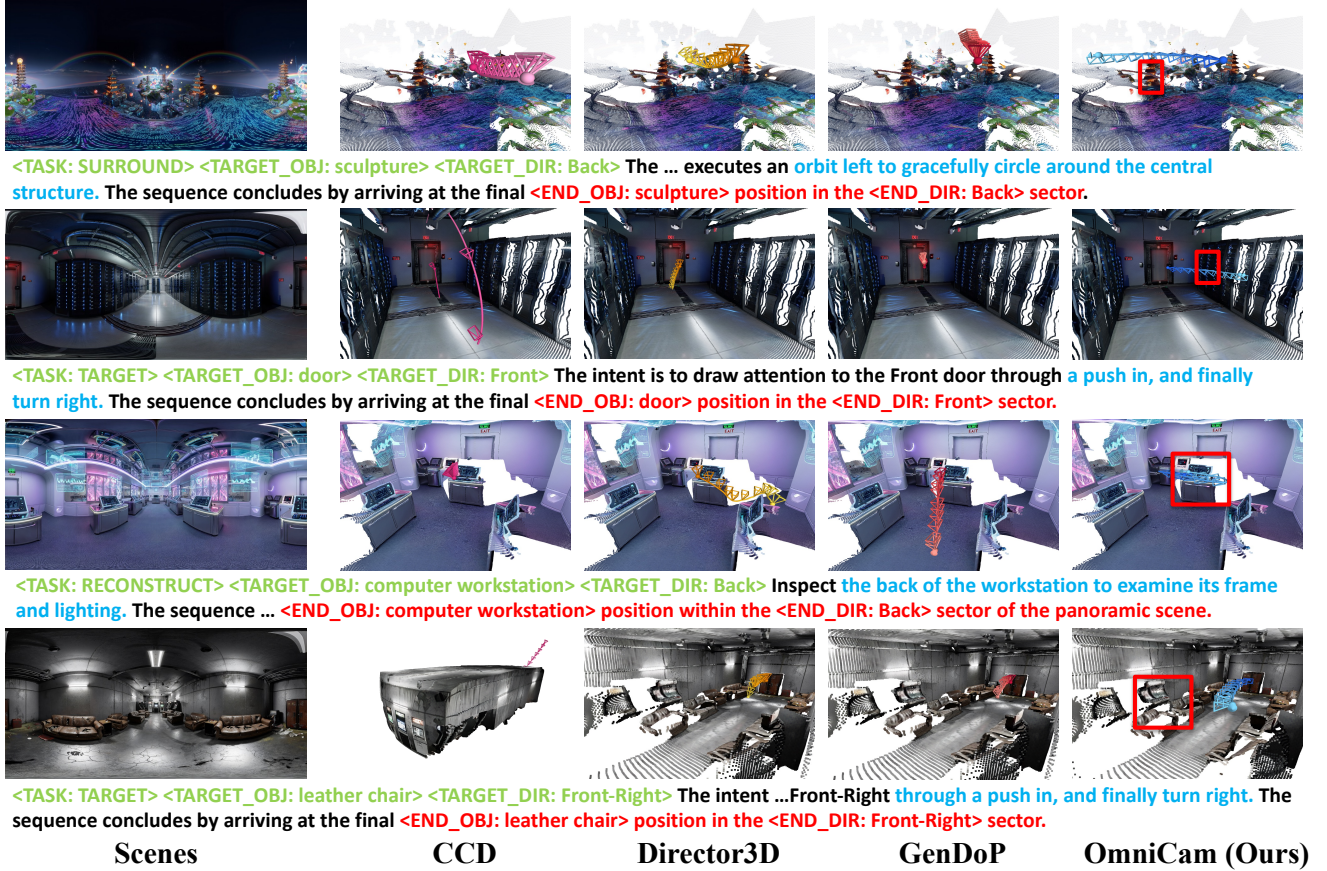


Figure 4. Qualitative comparison on representative panoramic scenes. Each column shows the trajectory produced by a specific method. The target object is indicated by a red bounding box.



Figure 5. **Qualitative comparison on panoramic scenes.** Each row displays an input panorama, generated video sequences, and 3D reconstruction results. The two rightmost columns compare reconstruction outputs without and with the OmniCam trajectory; no quantitative reconstruction metric is reported here.

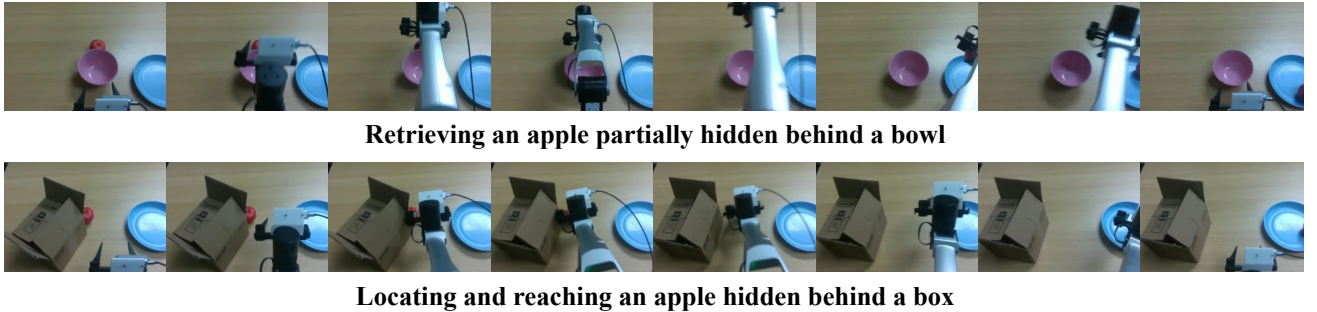


Figure 6. **Real-robot demonstration of OmniCam-guided active perception.** OmniCam plans a camera trajectory that actively repositions the observation camera to resolve target occlusion.

and angular deviation. These reference-trajectory errors measure agreement with the generated supervision; multiple distinct trajectories can satisfy the same instruction.

Implementation. OmniCam uses an OPT-style [35] decoder with FlashAttention-2 [9], trained in bf16 on $8 \times \text{H20}$ GPUs for ~ 100 epochs. The base configuration (hidden dim 1024, 8 heads, 12 layers; $\sim 500\text{M}$ trainable parameters) is used for the main comparisons; a larger variant ($\sim 1\text{B}$) is analyzed in Section E.2. Hyperparameters are provided in Section B.

5.2. Main Results

Quantitative Comparison. On OmniCaT (Table 1), OmniCam reduces ATE, FDE, RPE-R, and RPE-T by 46.6%, 41.1%, 28.0%, and 28.9% relative to retrained GenDoP, respectively. Collision decreases from 30.4% to 10.4% (65.8% relative reduction), and visibility rises from 33.6% to 52.0% (54.8% relative increase). Director3D is the best baseline for ATE and collision; relative to it, the reductions are 43.0% and 62.3%. The scene-bootstrap intervals for OmniCam and GenDoP are reported in Section M; component-specific effects are examined separately below.

Zero-Shot Cross-Dataset Transfer. The reported DataDoP

Table 2. **Ablation studies on OmniCaT.** The dual-branch panoramic encoder, target injection components, and pose tokenization schemes are investigated.

	Variant	ATE ↓	FDE ↓	RPE-R ↓	RPE-T ↓	Coll. (%) ↓	Vis. (%) ↑	Angle (°) ↓
<i>Encoder</i>	w/o semantic branch	2.186	3.247	2.694	0.072	13.6	11.5	64.7
	w/o geometric branch	1.774	3.084	2.881	0.049	45.8	33.6	19.7
<i>Geo. Input</i>	Panorama RGB only	1.536	2.687	2.143	0.044	38.7	36.2	18.4
	Panorama RGB-D	1.247	2.118	1.685	0.036	22.6	42.8	14.7
<i>Grounding</i>	w/o target cross-attention	1.673	2.259	2.774	0.042	15.2	25.8	35.6
	w/o target localization loss	1.582	2.148	2.661	0.038	14.8	32.4	28.7
	w/o target cross-attn. & loss	1.924	2.576	2.903	0.051	18.9	21.3	40.2
<i>Tokenization</i>	All absolute (rot. + trans.)	1.278	1.886	1.625	0.058	14.3	41.6	15.3
	All relative (rot. + trans.)	1.449	1.927	1.984	0.062	16.6	43.2	16.4
	Hybrid w/o hemisphere consistency	1.094	1.756	1.847	0.033	12.2	46.1	13.8
Full	OmniCam	0.868	1.462	1.135	0.027	10.4	52.0	10.9

results favor OmniCam over GenDoP trained on DataDoP: ATE 0.925 vs. 1.437, collision 8.9% vs. 28.5%, visibility 64.2% vs. 36.6%, and angle 12.7° vs. 29.8°. OmniCam is reported to use no DataDoP fine-tuning. Establishing the extent of transfer requires the input adaptation, data-overlap checks, and metric protocol identified above; these aggregate values alone do not rule out planner bias or differences in available scene information.

Comparison with the Heuristic Planner. With access to reconstructed geometry, navigation meshes, and semantic annotations, the planner has 5.2% collision and 58.7% visibility, compared with OmniCam’s 10.4% and 52.0%. This is a privileged-information reference, not a mathematical performance bound. The reported runtimes are 0.84s and approximately 16.5s (Section J); matched timing conditions require verification.

Per-Behavior Analysis. Section D reports lower ATE and FDE for OmniCam in all four behavior categories. These results describe coverage of the evaluated behaviors without identifying which component causes each gain.

Qualitative Comparison. Figure 4 compares trajectories across representative scenes. OmniCam trajectories remain aligned with navigable regions, follow the intended view-point of the referenced object, and exhibit smoother transitions in both position and orientation compared to all baselines. Additional qualitative examples are provided in Section K.

5.3. Ablation Studies

Dual-Branch Encoder. Removing the semantic branch reduces visibility from 52.0% to 11.5%, while removing the geometric branch increases collision from 10.4% to 45.8%. This supports the usefulness of both pathways. Because branch removal also changes capacity and available information, a capacity-matched single-branch comparison is needed to isolate the effect of routing.

Geometric Input Modality. RGB-only, RGB-D, and point-cloud variants yield collision rates of 38.7%, 22.6%, and 10.4%, respectively. The results favor explicit 3D processing in these configurations. Since the point cloud is unprojected from depth, this comparison concerns representation and processing, rather than additional geometric information created by unprojection.

Target Centroid Injection. Removing target cross-attention changes visibility from 52.0% to 25.8% and angle from 10.9° to 35.6°. Removing the localization loss yields 32.4% and 28.7°; removing both yields 21.3% and 40.2°, the worst values among these grounding variants. The comparison supports explicit target anchoring in the tested architecture (Section G).

Hybrid Pose Encoding. Hybrid encoding obtains RPE-R 1.135 and RPE-T 0.027, compared with 1.625/0.058 for all-absolute and 1.984/0.062 for all-relative encoding. Removing sign alignment increases RPE-R to 1.847. The representation analysis in Section I concerns quantization and sign selection; it does not guarantee the accuracy of autoregressive predictions.

5.4. Downstream Application Validation

With WorldStereo [36], OmniCam trajectories improve the reported CLIP score by 14.1% and consecutive-frame SSIM by 15.4% relative to GenDoP. Figure 5 illustrates video and reconstruction outputs. Robotic active-perception results and a qualitative real-robot demonstration are presented in Section L and Fig. 6. The robot success-rate aggregation needs verification; the demonstration is evidence of feasibility rather than a systematic sim-to-real evaluation.

6. Conclusion

OmniCam generates camera trajectories from panoramic imagery and language using point-cloud conditioning, hybrid pose serialization, and explicit 3D target anchoring.

OmniCaT supplies 267,700 examples across four camera behaviors. The reported comparisons and ablations support these design choices within the evaluated settings, and downstream studies examine their practical use. The representation analysis distinguishes quaternion sign consistency from lossy quantization and identifies the error-accumulation tradeoff of relative encoding. Remaining limitations include estimated and incomplete geometry, planner-derived supervision, static-scene observations, and the need for independently verified evaluation protocols. Future work includes dynamic scenes and closed-loop planning.

References

- [1] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in Atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.
- [2] Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15791–15801, 2025.
- [3] Jim Blinn. Where am i? what am i looking at?(cinematography). *IEEE Computer Graphics and Applications*, 8(4):76–81, 2002.
- [4] Rogerio Bonatti, Wenshan Wang, Cherie Ho, Aayush Ahuja, Mirko Gschwindt, Efe Camci, Erdal Kayacan, Sanjiban Choudhury, and Sebastian Scherer. Autonomous aerial cinematography in unstructured environments with learned artistic decision-making. *Journal of Field Robotics*, 37(4):606–641, 2020.
- [5] Zhe Cao, Hang Gao, Karttikeya Mangalam, Qi-Zhi Cai, Minh Vo, and Jitendra Malik. Long-term human motion prediction with scene context. In *European Conference on Computer Vision*, pages 387–404. Springer, 2020.
- [6] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryal, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. SAM 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.
- [7] Robin Courant, Nicolas Dufour, Xi Wang, Marc Christie, and Vicky Kalogeiton. ET the exceptional trajectories: Text-to-camera-trajectory generation with character awareness. In *European Conference on Computer Vision*, pages 464–480. Springer, 2024.
- [8] Qiongjie Cui, Huaijiang Sun, and Fei Yang. Learning dynamic relationships for 3D human motion prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6519–6527, 2020.
- [9] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.
- [10] Steven M Drucker, Tinsley A Galyean, and David Zeltzer. Cinema: A system for procedural camera movements. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 67–70, 1992.
- [11] Quentin Galvane, Marc Christie, Chrstophe Lino, and Rémi Ronfard. Camera-on-rails: automated computation of constrained camera paths. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, pages 151–157, 2015.
- [12] Michael Grupp. evo: Python package for the evaluation of odometry and SLAM. <https://github.com/MichaelGrupp/evo>, 2017.
- [13] 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.
- [14] Yunzhong Hou, Liang Zheng, and Philip Torr. Learning camera movement control from real-world drone videos. *arXiv preprint arXiv:2412.09620*, 2024.
- [15] Chong Huang, Chuan-En Lin, Zhenyu Yang, Yan Kong, Peng Chen, Xin Yang, and Kwang-Ting Cheng. Learning to film from professional human motion videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4244–4253, 2019.
- [16] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.
- [17] Hongda Jiang, Bin Wang, Xi Wang, Marc Christie, and Baoquan Chen. Example-driven virtual cinematography by learning camera behaviors. *ACM Trans. Graph.*, 39(4):45, 2020.
- [18] Hongda Jiang, Xi Wang, Marc Christie, Libin Liu, and Baoquan Chen. Cinematographic camera diffusion model. *Computer Graphics Forum*, 43(2):e15055, 2024.
- [19] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Rotation invariant spherical harmonic representation of 3 d shape descriptors. In *Symposium on geometry processing*, pages 156–164, 2003.
- [20] Beomyoung Kim, Chanyong Shin, Joonhyun Jeong, Hyungsik Jung, Se-Yun Lee, Sewhan Chun, Dong-Hyun Hwang, and Joonsang Yu. ZIM: Zero-shot image matting for anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23828–23838, 2025.
- [21] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.
- [22] Soohwan Kim and Minkyung Kim. Rotation representations and their conversions. *Ieee Access*, 11:6682–6699, 2023.
- [23] Xinyang Li, Zhangyu Lai, Linning Xu, Yansong Qu, Liujuan Cao, Shengchuan Zhang, Bo Dai, and Rongrong Ji. Director3D: Real-world camera trajectory and 3D scene generation from text. *Advances in neural information processing systems*, 37:75125–75151, 2024.
- [24] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. DL3DV-10K: A large-scale scene dataset for deep learning-based 3D vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024.

- [25] Christophe Lino and Marc Christie. Intuitive and efficient camera control with the toric space. *ACM Transactions on Graphics (TOG)*, 34(4):1–12, 2015.
- [26] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.
- [27] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024.
- [28] Xinyi Liu, Tianyi Zhang, Matthew Johnson-Roberson, and Weiming Zhi. SplaTraj: Camera trajectory generation with semantic gaussian splatting. *arXiv preprint arXiv:2410.06014*, 2024.
- [29] Sebastian Pütz, Thomas Wiemann, Jochen Sprickerhof, and Joachim Hertzberg. 3D navigation mesh generation for path planning in uneven terrain. *Ifac-Papersonline*, 49(15):212–217, 2016.
- [30] Fabio Ruetz, Emili Hernández, Mark Pfeiffer, Helen Oleynikova, Mark Cox, Thomas Lowe, and Paulo Borges. OVPC mesh: 3D free-space representation for local ground vehicle navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8648–8654. IEEE, 2019.
- [31] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwei Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.
- [32] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. MoGe-2: Accurate monocular geometry with metric scale and sharp details. *arXiv preprint arXiv:2507.02546*, 2025.
- [33] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, et al. MVImgNet: A large-scale dataset of multi-view images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9150–9161, 2023.
- [34] Mengchen Zhang, Tong Wu, Jing Tan, Ziwei Liu, Gordon Wetzstein, and Dahua Lin. GenDoP: Auto-regressive camera trajectory generation as a director of photography. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18229–18239, 2025.
- [35] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.
- [36] Yisu Zhang, Chenjie Cao, Tengfei Wang, Xuhui Zuo, Junta Wu, Jianke Zhu, and Chunchao Guo. WorldStereo: Bridging camera-guided video generation and scene reconstruction via 3D geometric memories. *arXiv preprint arXiv:2603.02049*, 2026.
- [37] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.
- [38] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019.

A. Implementation Details of OmniCaT

A.1. Geometric Initialization and Refinement

Depth Alignment Optimization. During geometry initialization, the ERP space is subdivided into 42 perspective views (an increase from the default 12) to ensure dense coverage. Depth maps are aligned via a GPU-accelerated Least-Squares Minimal Residual (LSMR) solver. To mitigate artifacts, a grounding pipeline is employed to mask sky regions and to remove depth discontinuities (edge floaters).

NavMesh Engineering. The raw NavMesh generated by the Recast algorithm is configured with cell size 0.1 m, cell height 0.1 m, agent height 0.1 m, agent radius 0.05 m, maximum climb 0.05 m, and maximum slope 30° . Post-processing consists of: (1) height filtering to remove polygons above a roof threshold; (2) connectivity filtering to retain only the largest connected component; (3) surface snapping through dense ray-casting to correct misaligned vertices that lie below the actual mesh surface; and (4) boundary erosion using KD-Tree accelerated searches to maintain a safety buffer from mesh edges. The resulting navigation graph is constructed by sampling points on the NavMesh surface at 0.05 m spacing (up to 600K nodes), connecting neighbors within $5\times$ spacing with a height coherence constraint ($|\Delta y| \leq 0.5$ m), and computing shortest paths via Dijkstra from the camera origin.

A.2. Trajectory Planning Heuristics

Scoring Function for Target Paths. For target trajectories, 72 candidate nodes are sampled on a circular boundary at radius $r = \min(1.5 \times s_{obj}, d_{obj})$, where s_{obj} is the object scale and d_{obj} is the distance from the camera origin to the target. The optimal node is selected by maximizing a scoring function: $S = 5000 \cdot R_{los} - 100 \cdot \Delta\theta$, where R_{los} is the Line-of-Sight ratio calculated via a 50-point projection test between the candidate position and the target center, and $\Delta\theta$ is the angular deviation from the ideal approach direction. Paths are computed via Dijkstra on the NavMesh graph, pruned of backtracking segments, and smoothed with B-spline interpolation (smoothing factor 0.5). All trajectories are resampled to 81 keyframes.

Surround Behavior. For surround trajectories, the orbit radius is computed as $r = \min(d_{obj}, 2 \times s_{obj})$, clamped by a scene-dependent threshold ($4\times$ global median depth). 72 points are sampled on the orbit circle; only those reachable via the NavMesh graph are connected to form continuous surround paths.

Exploration Behavior. For wander trajectories, 8 radial directions are explored from the origin along the NavMesh, each extending up to $4\times$ global median depth in path length. The top 10 trajectories are selected via Farthest Point Sampling on endpoint positions, with a minimum angular separation of 40° and maximum overlap ratio of 0.5.

Reconstruction Behavior. Reconstruction targets are selected via the top-10 semantic objects ranked by segment area and viewpoint diversity (FPS in 3D space). For each target, multi-view paths are planned with a reduced smoothing factor (0.2) to preserve sharp viewpoint transitions optimal for 3D reconstruction coverage.

Path Smoothing and Orientation. All discrete waypoints are smoothed using B-splines and Gaussian filtering ($\sigma = 8.0$). Camera orientations are managed via Spherical Linear Interpolation (SLERP). For “Wander” and “Target” modes, orientations follow the path tangent and target centroid, respectively.

B. Training and Inference Details

The reported tokenizer uses $B = 256$ bins and a normalized displacement limit $\Delta t_{\max} = 0.025$. These settings determine precision and clipping tradeoffs; sign alignment does not remove quantization boundaries.

OmniCam is trained with the Accelerate framework under bf16 mixed precision. The autoregressive backbone is an OPT-style decoder equipped with FlashAttention-2. The optimizer is AdamW with $\beta = (0.9, 0.95)$, a weight decay of 0.01, a peak learning rate of 3×10^{-5} , a linear warm-up over the first one percent of the schedule, and cosine annealing for the remaining steps. The gradient is clipped to a global norm of 1.0 and gradient checkpointing is enabled throughout the decoder. The discrete bin size is $B = 256$, and the maximum translation delta is $\Delta t_{\max} = 0.025$ in the normalized coordinate system. The hidden dimension, number of heads, and number of layers are configured as the base variant (hidden dim 1024, 8 heads, 12 layers). Training is performed for approximately one hundred effective epochs on a single node with eight NVIDIA H20 GPUs, with a per-GPU batch size of thirty-two. The localization loss weight is set to $\lambda = 0.5$. Inference adopts the *sample* strategy with top- k truncation at $k = 10$ and with the format-constrained decoding procedure described in Section 4.4.

Weighted Cross-Entropy Loss. The per-token cross-entropy loss is weighted by three multiplicative factors to emphasize the most critical pose channels and learning phases: (1) *channel importance weights*: rotation channels receive weight 2.0, translation channels 1.5, and intrinsic channels 0.5, reflecting the relative importance of orientation accuracy over positional precision; (2) *temporal decay*: a linear schedule assigns weight 2.0 to the first frame and decreases to 1.0 at the final frame, prioritizing early trajectory establishment; and (3) *curriculum learning mask*: during the initial training phase, only early frames contribute to the loss, with later frames gradually enabled as training progresses, preventing the model from attempting to learn long-horizon dynamics before mastering short-horizon patterns.

Caption Dropout Probabilities. The five-variant caption dropout strategy applies the following distribution: full cap-

tion (35%), prefix tags only (25%), body text only (25%), motion tag only (5%), and partial prefix (10%). This is the distribution used in the reported experiments.

Computational Cost. Table 3 reports training and inference costs for GenDoP and two OmniCam configurations. The original timing description includes depth estimation, point-cloud construction, encoding, and autoregressive decoding. The prefix condition is reused across decoding steps. These reported timings do not establish real-time operation under an unspecified deployment budget.

C. Encoder Architecture Details

The dual-branch encoder routes appearance and text through a semantic pathway and point-cloud features through a geometry pathway. The outputs are fused by the decoder.

Semantic Branch. The panoramic image I_{pano} is processed by a frozen SAM 3 vision encoder with 1024-dimensional features. The target descriptor T_{obj} and caption guide the Language-Guided Resampler after visual feature extraction. This pathway provides target-related information, but it is not excluded from contributing to geometric decisions after decoder fusion.

Geometric Branch. Panoramic geometric priors are unprojected into a 3D point cloud, which is then structured into a voxel grid (0.02m resolution) to enable efficient feature extraction. A LitePT encoder uses positional and surface-normal features to extract per-voxel descriptors via sparse convolution, which are re-projected to yield $F_{geo} \in \mathbb{R}^{M \times C_{geo}}$. This branch has no direct text input. Removing it increases collision rate from 10.4% to 45.8% in Table 2; this supports the usefulness of the geometric pathway but does not isolate routing from capacity or input-information changes.

D. Per-Behavior Decomposition

Table 4 decomposes the OmniCaT results. Compared with retrained GenDoP, OmniCam has lower ATE for Wander (1.150 vs. 2.421), Target (0.677 vs. 1.088), Surround (1.258 vs. 1.932), and Reconstruct (0.385 vs. 1.055). The corresponding Target and Reconstruct reductions are 37.8% and 63.5%. This comparison supports performance across the four evaluated behaviors; it does not isolate caption dropout, loss weighting, or geometry as the cause of individual gains.

E. Scaling Analysis

We examine the reported dependence on training-data volume and model size. These finite comparisons show trends within the evaluated range; they do not establish a scaling law or the absence of representational bottlenecks.

E.1. Data Scaling

Table 5 reports the base model at four nominal data scales, labeled 27K, 53K, 107K, and 267K in the original experiment record. Metrics improve across these settings: ATE decreases from 1.483 to 0.868, collision from 19.8% to 10.4%, and visibility increases from 34.6% to 52.0%. Four data points do not determine a power law or establish gains beyond the observed range.

E.2. Model Scaling

Three model sizes are compared: Small (hidden dimension 512, 8 heads, 8 layers; approximately 120M trainable parameters), Base (1024, 8, 12; approximately 500M), and Large (1536, 16, 24; approximately 1B). The reported setup holds other hyperparameters constant while changing architecture dimensions. ATE decreases from 1.127 to 0.743, collision from 13.8% to 8.7%, and visibility increases from 44.6% to 56.3% (Table 6). These observations support gains at the tested sizes, without establishing which representational choices explain the trend.

F. Failure Case Analysis

Understanding when and why the framework fails is essential for guiding future improvements. Five categories of failures are identified, each linked to a specific component or assumption of the pipeline:

- **Noisy panoramic geometry / thin structures** (geometric branch limitation): in large open environments or near thin structures (fences, poles, railings), the monocular depth estimator [32] produces noisy or incomplete reconstructions with floating artifacts. These propagate into the LitePT voxel features, causing the geometric branch to either miss collision boundaries or hallucinate obstacles, resulting in physically implausible paths.
- **Ambiguous target phrases** (centroid injection limitation): when multiple semantically similar objects coexist (e.g., “the wooden door” in a hallway with several doors, or “the pillar” among clustered columns), the attention-aggregated centroid may ground to an incorrect instance. Spatial descriptors in the textual guidance partially mitigate this, but the failure persists when such cues are absent or when objects are visually indistinguishable from the panoramic viewpoint.
- **Dynamic and interactive scenes** (static-scene assumption): the current framework assumes a static panoramic observation. In environments with moving objects (people, vehicles, opening doors), the generated trajectory cannot react to temporal changes, potentially producing paths that collide with dynamic obstacles or fail to track moving targets. Extending OmniCam to video-conditioned or closed-loop settings is an important direction for future work.

Table 3. Reported computational costs for GenDoP and OmniCam Base/Large.

Method	Training Time	Inference Latency	#Params
GenDoP	~18h (8×H20)	0.42s / trajectory	~380M
OmniCam (Base)	~24h (8×H20)	0.84s / trajectory	~500M
OmniCam (Large)	~40h (8×H20)	1.20s / trajectory	~1B

Table 4. Per-behavior decomposition of the main result on OmniCaT. Metrics are reported separately for Wander, Target, Surround, and Reconstruct trajectories.

Method	Dataset	Wander		Target		Surround		Reconstruct	
		ATE ↓	FDE ↓	ATE ↓	FDE ↓	ATE ↓	FDE ↓	ATE ↓	FDE ↓
CCD [18]	Pre-trained	3.718	4.907	1.485	2.285	2.524	3.419	2.206	2.972
E.T. [7]	Pre-trained	3.151	4.083	1.735	1.945	1.580	1.640	1.467	1.804
Director3D [23]	Pre-trained	2.969	4.779	1.444	2.207	1.620	2.688	1.279	2.094
Director3D	OmniCaT	2.562	4.250	1.130	2.089	1.351	2.304	1.045	1.950
GenDoP [34]	Pre-trained	2.667	4.408	1.362	2.131	2.176	3.248	1.255	2.016
GenDoP	OmniCaT	2.421	3.652	1.088	1.933	1.932	2.773	1.055	1.574
OmniCam (Ours)	OmniCaT	1.150	2.301	0.677	0.962	1.258	1.836	0.385	0.749

- **Reflective/transparent surfaces** (semantic branch limitation): glass walls and mirrors produce ambiguous visual features in the vision encoder, leading to unreliable target-aware conditioning and erratic trajectory orientation near such surfaces.
- **Occluded targets** (centroid injection limitation): when the target object is heavily occluded, the cross-attention weights spread to visible but irrelevant regions, causing the estimated centroid to shift away from the true target location.

These cases identify limitations in perception and target grounding. Decoder errors, quantization, and exposure to planner-generated supervision can also contribute; the examples do not isolate their relative importance.

Broader Impacts. Scene-aware camera planning may assist content creation and robotic perception. Camera deployment can also raise privacy or surveillance concerns, and generated video may be used misleadingly. In robotics, errors in inferred geometry or open-loop trajectories can produce unsafe camera motions; the reported collision metrics are not deployment guarantees. Dataset release requires source-specific permission and license review, and physical deployment requires validation against the actual environment.

G. Attention Visualization and Centroid Analysis

The target centroid supplies an explicit spatial anchor (Section 4.3). The following visualizations illustrate the attention patterns used to derive this anchor.

Figure 7 shows the input panorama, reconstructed point

cloud, and attention projected onto scene points. The examples show concentration near referenced regions. These visualizations complement the grounding ablation, but do not quantify localization accuracy or establish that attention alone causes improved visibility. Point-level aggregation is specified conceptually in Equation (2); its implementation details require verification.

H. Caption Dropout Ablation

We compare no caption dropout, single-variant dropout (dropping the entire caption with probability 0.1), and the five-variant distribution in Section B. The reported values are means over three training seeds, with ATE standard deviation below 0.02. The five-variant scheme has higher visibility than no dropout (52.0% vs. 38.7%) and lower angular deviation (10.9° vs. 18.4°), while no dropout has slightly lower ATE (0.856 vs. 0.868) and collision (10.1% vs. 10.4%). This is a tradeoff between the reported metrics. Attribution to instruction-format overfitting requires separate evaluation on complete, partial, and rephrased instructions.

I. Theoretical Analysis of Hybrid Pose Tokenization

We analyze reconstruction of a known trajectory after quantization and dequantization. These results concern representation error; they do not bound autoregressive prediction error, depth-estimation error, or errors in scene-scale calibration. Let d denote the rotation-angle geodesic distance on $SO(3)$, measured in radians, with range $[0, \pi]$. This distance

Table 5. Reported effect of nominal data scale using the base architecture. Exact training-subset counts and the fixed evaluation split require verification.

#Trajectories	ATE ↓	FDE ↓	RPE-R ↓	RPE-T ↓	Coll. (%) ↓	Vis. (%) ↑
27K (10%)	1.483	2.586	1.836	0.054	19.8	34.6
53K (20%)	1.206	2.080	1.528	0.042	15.8	42.6
107K (40%)	1.003	1.709	1.297	0.033	12.7	48.2
267K (100%)	0.868	1.462	1.135	0.027	10.4	52.0

Table 6. Reported effect of model scale. The common training subset and parameter-count scope require verification against training records.

Size	#Params	ATE ↓	FDE ↓	RPE-R ↓	RPE-T ↓	Coll. (%) ↓	Vis. (%) ↑
Small	~120M	1.127	1.893	1.428	0.036	13.8	44.6
Base	~500M	0.868	1.462	1.135	0.027	10.4	52.0
Large	~1B	0.743	1.248	0.982	0.022	8.7	56.3

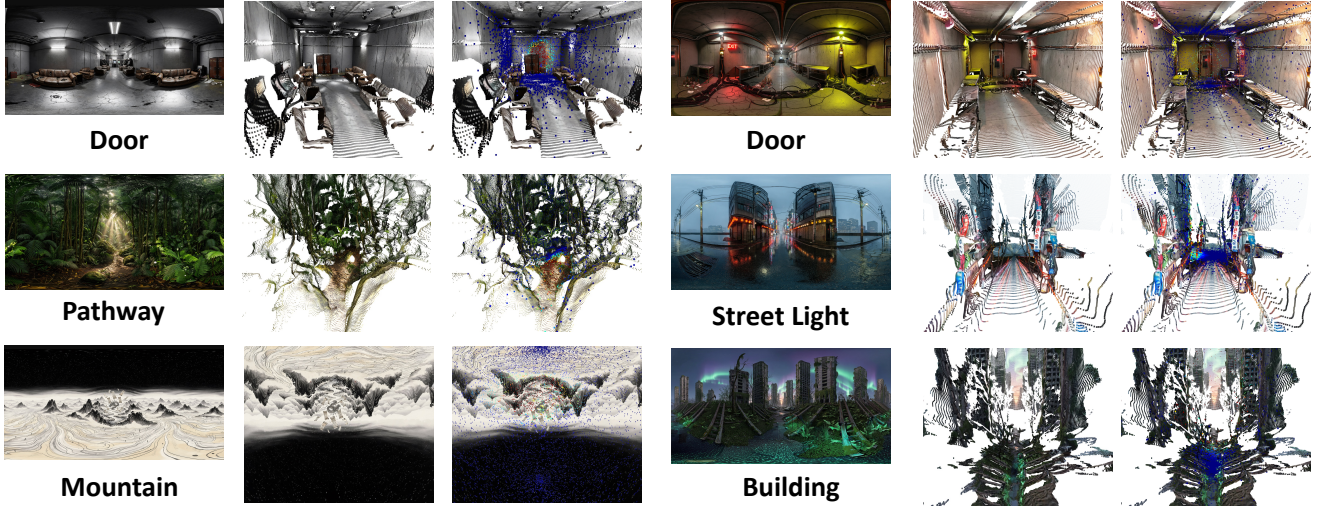


Figure 7. **Attention visualization of the Language-Guided Semantic Resampler.** Each group shows (from left to right) the input panorama, the reconstructed point cloud, and the 3D attention map. The maps visualize target-related attention on reconstructed geometry for the displayed examples; they are qualitative illustrations rather than a quantitative localization evaluation.

Table 7. Caption-dropout ablation. The five-variant scheme favors visibility and angle, while no dropout has slightly lower geometric errors.

Dropout Strategy	ATE ↓	FDE ↓	RPE-R ↓	RPE-T ↓	Coll. (%) ↓	Vis. (%) ↑	Angle (°) ↓
No dropout	0.856	1.438	1.118	0.026	10.1	38.7	18.4
Single-variant	0.862	1.451	1.126	0.027	10.3	45.8	14.2
Full (5-variant)	0.868	1.462	1.135	0.027	10.4	52.0	10.9

is invariant under left and right multiplication by a rotation.

Proposition 1 (Deterministic bounds on rotation reconstruction error). *Let $R_1, \dots, R_T \in \text{SO}(3)$ and $\Delta R_t = R_{t-1}^{-1} R_t$ for $t \geq 2$. Suppose decoded relative rotations satisfy $d(\widehat{\Delta R}_t, \Delta R_t) \leq \epsilon_t$, and the decoded initial rotation satisfies $d(\widehat{R}_1, R_1) \leq \epsilon_1$. Reconstruct relative orientations by $\widehat{R}_t = \widehat{R}_{t-1} \widehat{\Delta R}_t$. Then*

$$d(\widehat{R}_t, R_t) \leq \min \left\{ \pi, \epsilon_1 + \sum_{j=2}^t \epsilon_j \right\}. \quad (4)$$

If $\widehat{R}_1 = R_1$ and every $\epsilon_j \leq \epsilon$, this bound becomes $\min\{\pi, (t-1)\epsilon\}$. In contrast, if absolute orientations are decoded with per-frame reconstruction error at most ϵ , then $d(\widehat{R}_t^{\text{abs}}, R_t) \leq \epsilon$ at every frame, without integrating earlier quantization residuals.

Proof. The triangle inequality and bi-invariance give

$$d(\widehat{R}_t, R_t) \leq d(\widehat{R}_{t-1} \widehat{\Delta R}_t, \widehat{R}_{t-1} \Delta R_t) + d(\widehat{R}_{t-1} \Delta R_t, R_{t-1} \Delta R_t) \quad (5)$$

$$= d(\widehat{\Delta R}_t, \Delta R_t) + d(\widehat{R}_{t-1}, R_{t-1}). \quad (6)$$

Iterating this inequality proves the sum bound. Every rotation-angle distance is at most π , giving the stated minimum. The absolute-encoding bound follows directly from its assumed per-frame reconstruction guarantee. $\square$

This is a worst-case upper bound, not a claim that relative errors necessarily grow at each frame or obey a particular expected growth rate. Residuals can cancel. Absolute encoding avoids integration of rotation quantization residuals, but predicted absolute tokens can still depend on previous generated tokens and therefore need not have independent prediction errors.

Connecting component quantization to angular error.

The bound ϵ depends on the quantizer and the rotation decoder; B component bins do not by themselves imply $\epsilon = \pi/B$. For a conditional example, let $q \in \mathbb{S}^3$ be a unit quaternion, and suppose component dequantization produces v with $\|v - q\|_2 \leq \eta < 1$. If the decoder normalizes v to $\widehat{q} = v/\|v\|_2$, then

$$d(R(\widehat{q}), R(q)) \leq 2 \arcsin \eta. \quad (7)$$

Indeed, $q^\top v > 0$, so the angle ϕ between q and v lies in $[0, \pi/2)$. Decomposing the error along $\widehat{q}$ gives $\|v - q\|_2^2 = (\|v\|_2 - \cos \phi)^2 + \sin^2 \phi$, hence $\phi \leq \arcsin \eta$. The corresponding rotation angle is 2ϕ . For uniform bins of width h , center decoding without clipping gives at most $h/2$ error per quaternion component and thus $\eta \leq h$. In particular, $B > 2$ equal bins on $[-1, 1]$ give the conditional bound $2 \arcsin(2/B)$. Applying such a numerical bound requires these decoder and range assumptions to hold.

Proposition 2 (Representative-independent quaternion sign selection). *Let $R_1, \dots, R_T$ satisfy $d(R_{t-1}, R_t) < \pi$ for every $t \geq 2$, and choose arbitrary unit-quaternion representatives q_t of R_t . For $q = (w, x, y, z)$, let $c(q)$ select the sign for which the first nonzero component in the order (w, x, y, z) is positive. Define*

$$\begin{aligned} \widetilde{q}_1 &= c(q_1), \\ \widetilde{q}_t &= \begin{cases} q_t, & \langle \widetilde{q}_{t-1}, q_t \rangle > 0, \\ -q_t, & \langle \widetilde{q}_{t-1}, q_t \rangle < 0, \end{cases} \quad t \geq 2. \end{aligned} \quad (8)$$

This produces a unique aligned quaternion sequence determined by the rotation sequence, independently of the initial choices between q_t and $-q_t$. Consequently, any deterministic component quantizer with fixed bin-boundary rules produces the same token sequence from all such choices of input representatives. The tokenization remains many-to-one.

Proof. A unit quaternion has a nonzero component, and $c(q) = c(-q)$, so the first representative is uniquely fixed, including when $w_1 = 0$. Suppose the aligned representative at $t-1$ is fixed. The quaternion formula for rotation distance gives

$$|\langle \widetilde{q}_{t-1}, q_t \rangle| = \cos \left(\frac{d(R_{t-1}, R_t)}{2} \right) > 0.$$

Exactly one of q_t and $-q_t$ therefore has positive inner product with the already aligned previous quaternion. Replacing the input q_t by $-q_t$ reverses the inner-product sign and leaves the selected representative unchanged. Induction proves representative independence of the entire sequence, and deterministic quantization preserves this property. It does not preserve injectivity: a finite token vocabulary cannot distinguish every continuous rotation sequence. $\square$

The proposition specifies sufficient sign and boundary conventions; a concrete implementation must document these choices. At an exactly π inter-frame rotation, both inner products are zero and an additional deterministic tie rule is required, such as selecting $c(q_t)$. No shortest-arc preference exists at that tie. This sequence-level sign selection removes double-cover ambiguity under the stated conventions; it does not construct a globally continuous quaternion parameterization of $\text{SO}(3)$, remove quantization boundaries, or make lossy quantization invertible.

Translation: local resolution versus accumulated error.

For additive displacements in a common coordinate frame, relative translation trades a smaller encoding range for integration error. If normalized absolute coordinates lie in $[-L, L]$ and displacements lie in $[-D, D]$ per component, uniform center decoding with B bins and no clipping bounds

Table 8. **Privileged-information planner comparison.** The planner receives reconstructed geometry, navigation meshes, and semantic annotations. OmniCam estimates geometry from the input panorama and text.

Method	NavMesh/annotation access	Runtime	Coll. (%) ↓	Vis. (%) ↑	Angle (°) ↓
Heuristic Planner	Yes (privileged)	~16.5s	5.2	58.7	9.4
GenDoP [34]	No	0.42s	30.4	33.6	27.6
OmniCam (Ours)	No	0.84s	10.4	52.0	10.9

the per-frame absolute position error by $\sqrt{3}L/B$ and each displacement error by $\sqrt{3}D/B$. Integrating the latter gives

$$\|\hat{\mathbf{p}}_t - \mathbf{p}_t\|_2 \leq \|\hat{\mathbf{p}}_1 - \mathbf{p}_1\|_2 + (t-1) \frac{\sqrt{3}D}{B}. \quad (9)$$

Thus $D \ll L$ improves local displacement resolution, while long sequences can accumulate translation error. If displacements are expressed in rotating local frames, orientation errors must also be included. The hybrid representation is therefore a design tradeoff whose trajectory-level benefit requires empirical evaluation, rather than a guarantee of universally minimal pose error.

J. Heuristic Planner Oracle Comparison

The heuristic planner is evaluated with reconstructed point clouds, navigation meshes, semantic masks, and target annotations. This is a privileged-information comparison on estimated geometry, rather than an oracle with complete physical ground truth. It attains 5.2% collision and 58.7% visibility, versus OmniCam’s 10.4% and 52.0%. OmniCam estimates geometry internally from the panorama and does not run navmesh construction or graph search. The reported runtimes (approximately 16.5s vs. 0.84s) differ by a factor of 19.6; the timing assumptions must be matched before attributing the difference to a particular component.

K. More Qualitative Results

Figure 8 provides additional qualitative trajectory comparisons. Each row shows one scene, and the columns show the outputs of different methods. These selected examples illustrate differences in trajectory placement and orientation; they do not establish dataset-wide collision or smoothness guarantees.

Figure 9 presents additional downstream examples from indoor, outdoor, and stylized environments. Each row contains a panorama, selected generated video frames, and reconstruction results with and without the OmniCam trajectory. The figure illustrates qualitative differences; reconstruction accuracy and completeness require a separate quantitative evaluation.

L. Downstream Application Validation

Camera-Controlled Video Generation. OmniCam and GenDoP trajectories serve as camera control signals for WorldStereo [36]. The reported evaluation uses 484 trajectories across indoor, outdoor, and stylized scenes. For each prompt, the resulting poses drive point-cloud rendering and subsequent video synthesis.

Generated videos are evaluated with five reported measures. *CLIP-Score* averages the cosine similarity between CLIP ViT-B/32 image and scene-description embeddings over 10 sampled frames. *Consecutive-frame SSIM* measures adjacent-frame similarity and is sensitive to motion magnitude; it is not a motion-compensated temporal-consistency metric. *Aesthetic proxy* combines CLIP embedding magnitude and inter-frame embedding coherence on a reported 0–10 scale. Its formula and normalization require verification. *Known-region Fidelity* compares generated frames with point-cloud renderings in observed regions using SSIM, so it uses a rendering reference. *Generation Coverage* is the mean unknown-region mask ratio, a descriptor of synthesis difficulty rather than an accuracy metric.

Relative to GenDoP, the reported OmniCam results improve CLIP score by 14.1%, consecutive-frame SSIM by 15.4%, the aesthetic proxy by 18.0%, and known-region fidelity by 9.2%. The unknown-region mask ratio is 38.5% vs. 35.8%. These results describe outputs of the trajectory-plus-video pipeline; they do not isolate framing, path length, speed, or geometric consistency as the cause of improvement. Quantitative reconstruction accuracy and completeness are not evaluated in this table.

Robotic Manipulation (Active Perception). OmniCam is evaluated as an active-perception module for $\pi_{0.5}$ [16], generating camera trajectories to improve target observations. A trial succeeds when the designated object is grasped within the time limit. The original record describes 200 episodes across 50 tabletop scenes, with reported success rates of 72.5% for OmniCam, 48.0% for GenDoP, and 56.3% for a fixed camera. The fixed-camera percentage is incompatible with a single pooled denominator of 200 binary trials; its aggregation remains unresolved.

Figure 6 presents qualitative physical-robot grasping examples with initially occluded targets. The demonstration illustrates feasibility in the shown cases. A systematic sim-

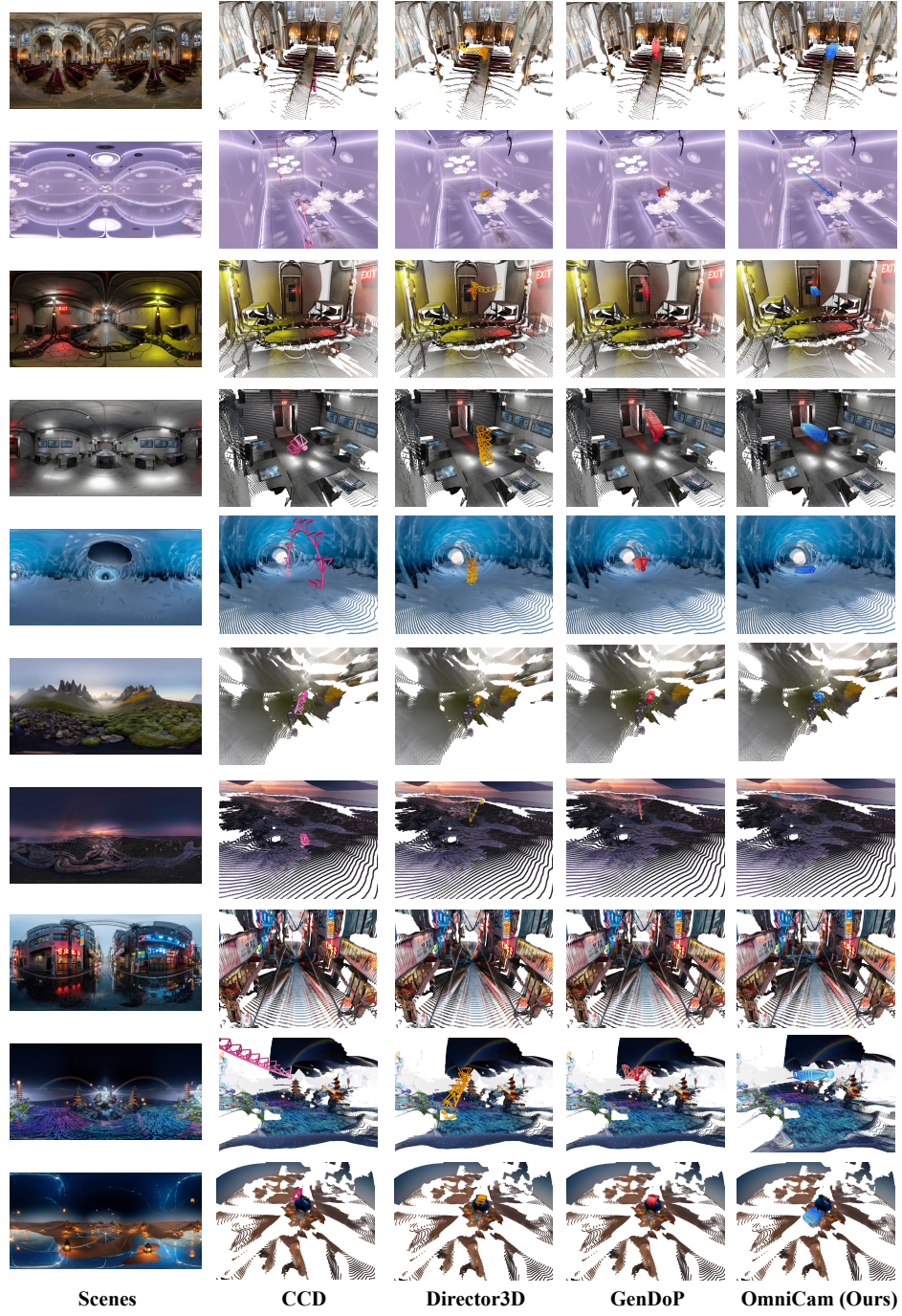


Figure 8. **Additional qualitative comparison on panoramic scenes.** Extending the comparison in Figure 4, each row shows a different scene spanning indoor, outdoor, and stylized environments, and each column shows the trajectory produced by a specific method. The examples illustrate differences in target-directed trajectories across scene categories. The trajectory captions used for each scene are listed in Table 9.

Table 9. **Trajectory captions for Figure 8.** Each row corresponds to a scene in the qualitative comparison figure, listing the scene type, target object, and the hierarchical textual instruction provided to OmniCam.

Row	Type	Target	Caption
1	Target	door	<TASK: TARGET> <TARGET_OBJ: door> <TARGET_DIR: Back> Approach the back door to bring it into focus. The sequence concludes by arriving at the final <END_OBJ: door> position within the <END_DIR: Back> sector of the panoramic scene.
2	Target	light fixture	<TASK: TARGET> <TARGET_OBJ: light fixture> <TARGET_DIR: Front> Approach the front light fixture to bring it into focus. The sequence concludes by arriving at the final <END_OBJ: light fixture> position within the <END_DIR: Front> sector of the panoramic scene.
3	Wander	free space	<TASK: WANDER> <TARGET_OBJ: free space> <TARGET_DIR: Front> Approach the front workstation to explore the layout of the industrial corridor. The sequence concludes by arriving at the final <END_OBJ: free space> position within the <END_DIR: Front> sector of the panoramic scene.
4	Target	door	<TASK: TARGET> <TARGET_OBJ: door> <TARGET_DIR: Back> Approach the back door to bring it into focus. The sequence concludes by arriving at the final <END_OBJ: door> position within the <END_DIR: Back> sector of the panoramic scene.
5	Target	ice tunnel	<TASK: TARGET> <TARGET_OBJ: ice tunnel> <TARGET_DIR: Front> Approach the ice tunnel to bring it into focus. The sequence concludes by arriving at the final <END_OBJ: ice tunnel> position within the <END_DIR: Front> sector of the panoramic scene.
6	Wander	free space	<TASK: WANDER> <TARGET_OBJ: free space> <TARGET_DIR: Front> Approach the distant peaks to explore the mountain valley. The sequence concludes by arriving at the final <END_OBJ: free space> position within the <END_DIR: Front> sector of the panoramic scene.
7	Surround	clouds	<TASK: SURROUND> <TARGET_OBJ: clouds> <TARGET_DIR: Front> Capture the clouds from multiple angles to highlight their form and color against the twilight sky. The sequence concludes by arriving at the final <END_OBJ: clouds> position within the <END_DIR: Front> sector of the panoramic scene.
8	Surround	neon sign	<TASK: SURROUND> <TARGET_OBJ: neon sign> <TARGET_DIR: Right> Capture the right neon sign from multiple angles to highlight its glowing details and reflections. The sequence concludes by arriving at the final <END_OBJ: neon sign> position within the <END_DIR: Right> sector of the panoramic scene.
9	Wander	free space	<TASK: WANDER> <TARGET_OBJ: free space> <TARGET_DIR: Right> Approach the right to explore the surreal landscape beyond the pagoda. The sequence concludes by arriving at the final <END_OBJ: free space> position within the <END_DIR: Right> sector of the panoramic scene.
10	Surround	sand dune	<TASK: SURROUND> <TARGET_OBJ: sand dune> <TARGET_DIR: Front> Capture the sand dune from multiple angles to reveal its textured ridges and surrounding lanterns. The sequence concludes by arriving at the final <END_OBJ: sand dune> position within the <END_DIR: Front> sector of the panoramic scene.

Table 10. **Downstream evaluation: camera-controlled video generation with WorldStereo [36].** Different trajectory sources are compared using the same video generation model.

Method	CLIP-S $\uparrow$	Adj.-SSIM $\uparrow$	Aest. proxy $\uparrow$	Fidelity $\uparrow$	Coverage
GenDoP [34] + WorldStereo	0.2644	0.4248	7.647	0.6924	35.8%
OmniCam (Ours) + WorldStereo	0.3018	0.4904	9.021	0.7560	38.5%

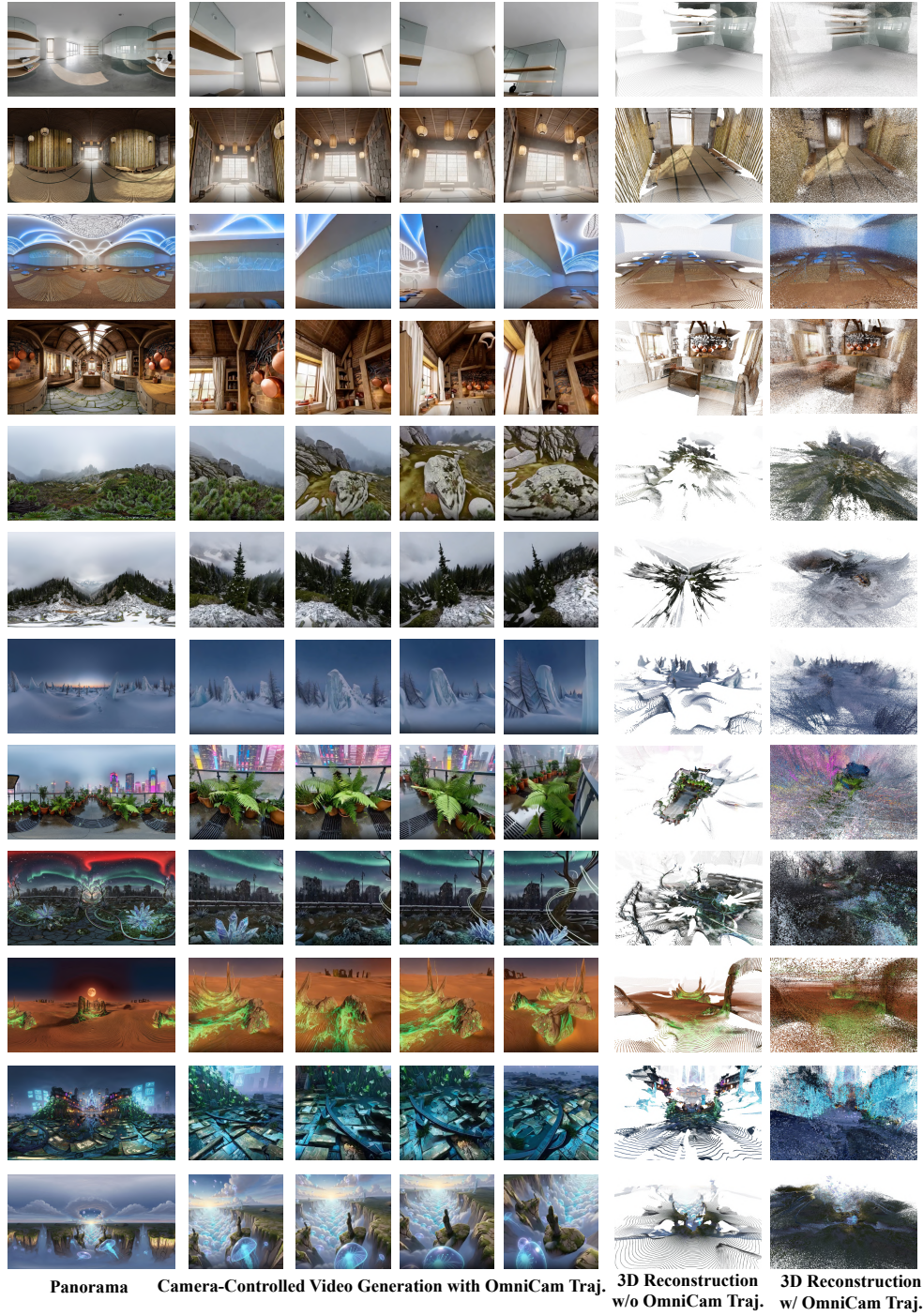


Figure 9. **Qualitative downstream results across diverse panoramic scenes.** Each row shows the input panorama (left), four selected frames from camera-controlled video generation using an OmniCam trajectory (middle), and 3D reconstruction results without and with the OmniCam trajectory (two rightmost columns). The examples include indoor, outdoor, and stylized environments. These visual comparisons illustrate the pipeline outputs; they do not quantify reconstruction accuracy or guarantee collision-free motion.

Table 11. **Downstream evaluation: robotic manipulation with $\pi_{0.5}$.** Reported grasp success (SR) and execution visibility (Vis.). The stated 200-episode setup and fixed-camera SR aggregation require reconciliation with the episode records.

Trajectory Source	Grasp SR (%) $\uparrow$	Exec. Vis. (%) $\uparrow$
$\pi_{0.5}$ [16] (fixed camera)	56.3	47.2
GenDoP [34]	48.0	52.6
OmniCam (Ours)	72.5	71.8

to-real claim would require repeated physical trials, matched fixed-camera controls, and documented camera motions and failure counts.

M. Statistical Confidence Analysis

Table 12 reports 95% confidence intervals from 1,000 bootstrap resamples of OmniCaT evaluation scenes for OmniCam and retrained GenDoP. The listed intervals do not overlap. These estimates characterize variation across evaluation scenes, not variation across training seeds, and do not cover all other experiments. GenDoP is the comparison in this table, but is not the best baseline for every main-table metric.

Table 12. **Bootstrap 95% confidence intervals on OmniCaT evaluation scenes** (1,000 resamples). Intervals describe evaluation-scene variability for the two listed methods.

Method	ATE ↓	FDE ↓	RPE-R ↓	RPE-T ↓	Coll. (%) ↓	Vis. (%) ↑	Angle (°) ↓
GenDoP (OmniCaT)	1.624±0.087	2.483±0.124	1.577±0.068	0.038±0.003	30.4±1.8	33.6±2.1	27.6±1.4
OmniCam (Ours)	0.868±0.042	1.462±0.071	1.135±0.053	0.027±0.002	10.4±1.1	52.0±1.9	10.9±0.8