VISPAC: VISUAL SPATIAL POSE CODEC FOR CAMERA-CONTROLLED VIDEO GENERATION

Anonymous authors
Paper under double-blind review

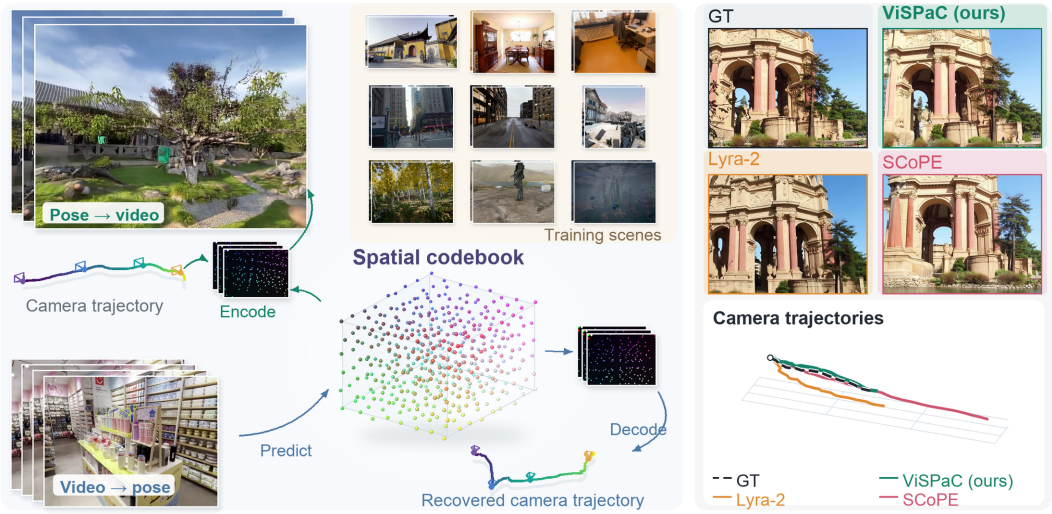

Figure 1: ViSPaC represents camera motion as evolving projections of colored landmarks in a synthetic 3D reference structure. Left: a shared model supports camera-controlled video generation and camera trajectory recovery from video. Center: the spatial codebook applies across diverse scenes without scene reconstruction; example training scenes are shown above. Right: generated views and estimated trajectories illustrate closer agreement with GT than Lyra-2 (Shen et al., 2026) and SCoPE (Yin et al., 2026).

ABSTRACT

Camera encodings and scene geometry have enabled substantial progress in camera-controlled video generation. We view camera control as a representation problem of relating geometric motion to visual dynamics, and propose ViSPaC, a Visual Spatial Pose Codec for joint modeling of camera motion and video. ViSPaC represents camera trajectories as evolving projections of colored landmarks in a synthetic 3D reference structure. These projections expose the visual effects of camera motion, while landmark identities and image positions establish correspondences for geometric recovery, without requiring reconstruction of the underlying scene. We identify two coupled constraints on this representation: visual compression limits the number of reliably distinguishable landmark identities, while their spatial distribution determines coverage across camera views. ViSPaC separates color identity from spatial placement and uses a spatial codebook to adapt the reference structure's scale and layout within a limited identity budget. A shared video diffusion transformer models video and visual pose, supporting both camera-controlled generation and video-conditioned pose recovery. Controlled comparisons show that ViSPaC improves camera control and pose recovery over alternative representations, with further gains from its identity budget and spatial adaptation. On DL3DV and Tanks and Temples, ViSPaC outperforms both implicit camera-conditioning and explicit scene-geometry methods, reducing relative translation and rotation errors by an average of 24.7% and 31.7%, respectively, over the previous state of the art.

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.

1 INTRODUCTION

Camera motion determines how a scene is observed over time, shaping viewpoint changes, image motion, and the visibility of scene content. Controlling camera motion in video generation enables users to prescribe how generated environments are explored and presented. Existing approaches incorporate camera geometry through several routes. Camera encodings supply pose information through ray features or geometry-aware attention (He et al., 2024; Zheng et al., 2024; Yin et al., 2026). Approaches based on explicit scene geometry reconstruct observed content to provide target-view renderings or cross-view correspondences (Ren et al., 2025; Shen et al., 2026). More recently, Rays as Pixels (Jang et al., 2026) represents cameras as three-channel ray images, or raxels, and models them jointly with video in a shared visual latent space.

An accurate camera trajectory specifies how the observer moves, while a video model must realize this motion through changes in image content. The form of the camera representation determines how this geometric information is presented to the model. We view camera control as a representation problem of relating geometric motion to visual dynamics. Projecting a known 3D reference structure along a trajectory exposes changes in the positions and visibility of identifiable image elements, expressing camera motion in a form that can be modeled alongside video. The reference structure supplies these projection dynamics without requiring reconstruction of the depicted scene.

We introduce ViSPaC, a *Visual Spatial Pose Codec* that implements this view using colored 3D landmarks (Figure 1). As the camera moves, the landmarks retain their color identities while their projections move across the image, expressing camera motion through the spatial patterns and temporal changes that video models are trained to process. Geometric readout connects this visual motion back to a camera trajectory: landmark colors identify known 3D points, and their image positions establish 2D–3D correspondences for pose recovery with known intrinsics. These correspondences provide an explicit route from a predicted visual-pose sequence to the geometry it represents.

A shared three-channel format establishes compatibility with a video model, but the organization of the visual signal determines how it expresses motion and preserves geometric information. For landmark-based visual pose encoding, we identify two coupled constraints. First, compression by a video VAE limits the number of reliably distinguishable color identities: adding landmarks narrows the color intervals separating them. Second, the spatial distribution of these landmarks determines their coverage across camera views. The identity constraint prevents coverage from being improved simply by adding ever more points, making the allocation of a limited set of landmarks a central design choice. ViSPaC therefore separates color identity from spatial placement. A spatial codebook adapts the reference structure's scale and layout to the trajectory within a limited identity budget, while preserving the mapping needed for geometric recovery.

With this codec, camera motion serves as a visual condition for video generation; predicting the same representation also enables geometric recovery. We encode video and pose with a shared video VAE and model their latent streams jointly in a diffusion transformer (Peebles & Xie, 2023). Our controlled comparisons show that ViSPaC improves control and pose recovery over alternative pose encodings, and that its limited identity budget and spatial adaptation contribute to these gains. On DL3DV and Tanks and Temples, the resulting model achieves state-of-the-art camera-control accuracy compared with both implicit camera-conditioning methods and approaches based on explicit scene geometry.

Our contributions are threefold:

- We introduce ViSPaC, a visual pose codec that expresses camera motion through identifiable projections of a synthetic 3D reference structure, linking visual dynamics to explicit geometric correspondences without scene reconstruction.
- We identify the coupled constraints of landmark identity distinguishability and spatial coverage, and design a spatial codebook that separates color identity from spatial placement to accommodate different trajectories within a limited identity budget.
- ViSPaC achieves state-of-the-art camera-control accuracy against implicit conditioning and explicit scene-geometry approaches, with the same model also supporting video-conditioned pose recovery.

2 Related Work

Camera conditioning in video diffusion. Camera control methods differ in their motion representations and integration into the generator. MotionCtrl (Wang et al., 2024b) and CameraCtrl (He et al., 2024) encode camera matrices and per-pixel Plücker rays, respectively, through conditioning modules. Geometry-aware attention incorporates camera relationships more directly: CamI2V (Zheng et al., 2024) uses epipolar attention, PRoPE (Li et al., 2026) encodes relative projective relationships in multi-view transformers, and SCoPE (Yin et al., 2026) introduces camera-ray coordinates into video-transformer attention. A complementary direction expresses camera geometry as visual sequences. Rays as Pixels (Jang et al., 2026) encodes cameras as three-channel ray images and models them jointly with video in a shared visual latent space. ViSPaC jointly models video and geometrically decodable landmark projections, addressing both identity distinguishability and spatial coverage.

Scene geometry for camera control. Reconstructed scene geometry provides target-view guidance by projecting observed content along a prescribed camera trajectory. ViewCrafter (Yu et al., 2024), Gen3C (Ren et al., 2025), and Uni3C (Cao et al., 2025) use point-cloud renderings derived from input views or generated frames to guide video synthesis. Geometry also serves other roles in generation: Lyra-2 (Shen et al., 2026) uses per-frame geometry to retrieve relevant history and establish target-view correspondences, while RealCam-I2V (Li et al., 2025) uses metric scene reconstruction for trajectory specification and scene-constrained noise shaping. ViSPaC instead expresses camera motion through a synthetic reference structure without reconstructing scene content.

Geometry from visual and generative models. Visual geometry models such as VGGT (Wang et al., 2025a) and Depth Anything 3 (Lin et al., 2025) estimate cameras and scene geometry from images. Related efforts connect geometric inference with video generation: JOG3R (Huang et al., 2025a) jointly trains video generation and point-map reconstruction, while Geometry Forcing (Wu et al., 2026) aligns video-diffusion features with a pretrained geometric model. ViSPaC focuses on camera control through a geometrically decodable visual representation, which also supports trajectory recovery.

3 Method

3.1 Overview

Given an initial image I_0 and a prescribed camera trajectory G, our primary task is *pose-to-video* (P2V): generating an RGB video V that follows the trajectory. The same model also supports *video-to-pose* (V2P), recovering a camera trajectory $\hat{G}$ from an observed video V. We assume known camera intrinsics K in both tasks.

For a video $V = \{I_t\}_{t=0}^{T-1}$, let $P_t \in \mathrm{SE}(3)$ denote its camera-to-world poses and $K = \{K_t\}_{t=0}^{T-1}$ its intrinsics. We express the trajectory $G = \{G_t\}_{t=0}^{T-1}$ in the coordinate system of the first camera:

$$G_t = P_0^{-1} P_t, \qquad G_0 = \mathbf{I}_4. \tag{1}$$

Each G_t, with rotation R_t and translation $\mathbf{p}_t$, maps points from camera t to the first camera's coordinate system, preserving the input trajectory's translation units.

Figure 2 summarizes the framework. ViSPaC projects a synthetic 3D reference structure along G to produce a three-channel visual-pose sequence $S = \{S_t\}_{t=0}^{T-1}$, matching the video's spatial resolution and temporal sampling (Figure 2(a)). A shared frozen video VAE encodes video and visual pose into separate latent streams, which a diffusion transformer models jointly (Figure 2(b)). Visual pose conditions P2V generation; for V2P, a geometric decoder converts predicted $\hat{S}$ into $\hat{G}$ using landmark correspondences and K (Figure 2(c)).

3.2 Visual Pose Representation

For a given spatial configuration Λ, ViSPaC pairs each landmark's RGB identity with a known 3D position and projects it along the camera trajectory.

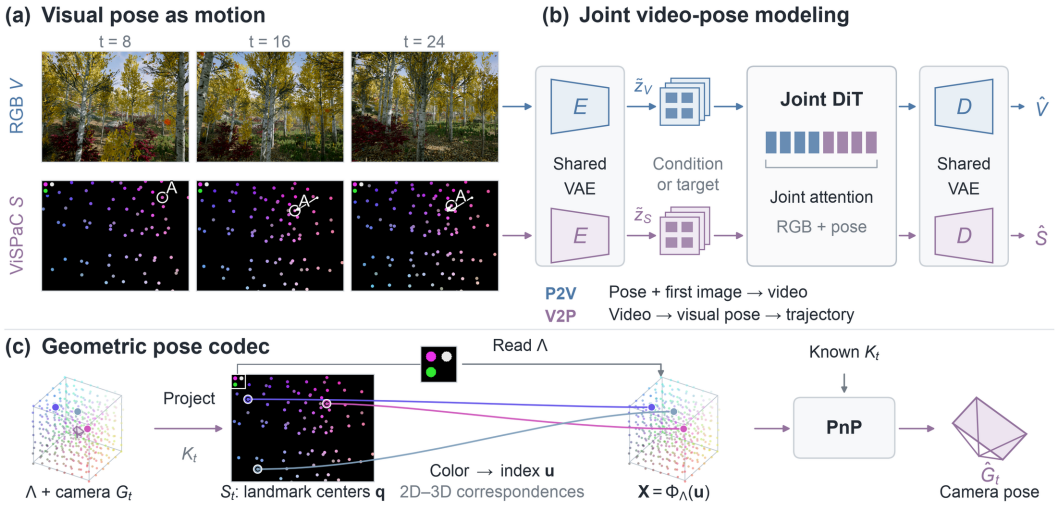

Figure 2: Overview of ViSPaC. (a) RGB video and visual pose follow the same camera trajectory; landmark A illustrates persistent identity across views. (b) A shared frozen Wan VAE (Wan et al., 2025) encodes both streams for joint diffusion modeling, supporting camera-controlled generation and video-conditioned pose recovery. (c) Projecting the reference structure encodes a trajectory. For decoding, configuration markers specify the reference geometry, while landmark colors and projected centers establish 2D–3D correspondences for PnP with known intrinsics. Trails and connecting lines are illustrative overlays.

Landmark identities. We define a discrete index grid with dimensions $\mathbf{n} = (n_x, n_y, n_z)$. Each landmark has an index $\mathbf{u} = (u_x, u_y, u_z)$, where $0 \le u_a < n_a$ and $n_a \ge 2$ for $a \in \{x, y, z\}$. Its RGB identity is

$$\mathbf{c}_\mathbf{u} = c_{\min}\mathbf{1} + (c_{\max} - c_{\min})\frac{\mathbf{u}}{\mathbf{n} - \mathbf{1}}, \tag{2}$$

where division is componentwise and $[c_{\min}, c_{\max}]$ is the color-coding range. Each channel encodes one component of the discrete index. The spatial configuration assigns that index a 3D position $\mathbf{X}_\mathbf{u} = \Phi_\Lambda(\mathbf{u})$ in the first-camera coordinate system (Figure 3(a)). Color thus encodes identity, while Λ determines its spatial location. Both positions and colors remain unchanged within a clip.

Projected visual motion. Using the camera convention in Section 3.1, we transform each landmark into the camera coordinate system at time t and project it to the image:

$$\mathbf{x}_{t,\mathbf{u}} = R_t^\mathsf{T}(\mathbf{X}_\mathbf{u} - \mathbf{p}_t), \qquad \mathbf{q}_{t,\mathbf{u}} = \pi(K_t\mathbf{x}_{t,\mathbf{u}}), \tag{3}$$

where π divides by the third coordinate to obtain the 2D image position $\mathbf{q}_{t,\mathbf{u}}$. As the camera moves, landmark colors preserve identity while projected positions and visibility change, expressing the trajectory as visual motion (Figure 2(a)).

We rasterize visible landmarks as uniformly colored disks centered at their projections. Appendix A.3 details visibility, disk sizing, background, and configuration markers. The construction yields a visual-pose sequence $S = \mathcal{R}(G; K, \Lambda)$ with the video's spatial resolution and temporal sampling.

Identity distinguishability. The index grid provides $N = n_x n_y n_z$ identities. Within the fixed coding range, adjacent color levels in channel a are separated by

$$\Delta c_a = \frac{c_{\max} - c_{\min}}{n_a - 1}. \tag{4}$$

Increasing the number of levels supplies more landmarks but narrows the intervals that distinguish their identities. Perturbations introduced by visual reconstruction can then change the decoded index. A coarser grid gives greater color separation, at the cost of fewer potential geometric correspondences.

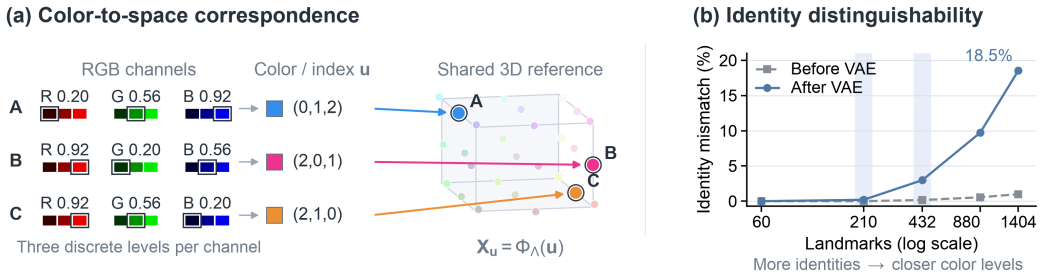

Figure 3: Landmark identity encoding and distinguishability. (a) Discrete RGB levels encode an index $\mathbf{u}$, which the spatial configuration Λ maps to a 3D landmark. The schematic uses three levels per channel, with colors displayed in $[0, 1]$. (b) Increasing the identity count narrows color intervals and increases identity mismatch after frozen Wan VAE reconstruction (Wan et al., 2025). Shading marks the grids used by ViSPaC.

Figure 3(b) measures this trade-off across five grid resolutions, before and after reconstruction by the frozen video VAE. After reconstruction, the aggregated identity mismatch rises from 0.2% at 210 landmarks to 3.0% at 432 and 18.5% at 1,404. These measurements support limiting the identity count for our encoding and VAE; ViSPaC uses grids with 210 and 432 landmarks. Appendix C.1 specifies the probe, metric aggregation, and complete results.

3.3 Spatial Adaptation and Geometric Recovery

Within the identity budget imposed by visual compression, expanding the reference structure increases landmark spacing. We therefore adapt its scale and layout to the camera path and projected coverage while preserving landmark identities.

Spatial configurations. We specify a configuration by $\Lambda = (s, \mathbf{n}, q)$, where s sets the scale and q indexes a layout with normalized origin $\mathbf{b}_q$ and span $\mathbf{h}_q$. The spatial mapping introduced in Section 3.2 is

$$\Phi_\Lambda(\mathbf{u}) = s\mathbf{b}_q + \mathbf{u} \odot \mathbf{d} + \boldsymbol{\delta}_{\Lambda,\mathbf{u}}, \qquad \mathbf{d} = \frac{s\mathbf{h}_q}{\mathbf{n} - \mathbf{1}}, \tag{5}$$

where $\mathbf{d}$ gives the grid spacing and $\odot$ denotes componentwise multiplication. A deterministic perturbation $\boldsymbol{\delta}_{\Lambda,\mathbf{u}}$ breaks regular grid alignments and can be reproduced during decoding. At fixed $\mathbf{n}$, varying s and q redistributes the 3D landmarks without changing their RGB identities. Our codebook provides standard, symmetric, and directional layouts (Figure 4(a)); Appendix A.1 specifies their geometry and perturbations.

Trajectory-based selection. For each layout, we select a scale from the camera path and check coverage in sampled views. We prefer the standard layout when containment and coverage requirements hold. Selection uses only the trajectory and intrinsics, with a single configuration per clip. Appendix A.2 gives the criteria and fallback rule.

Along the forward-moving trajectory in Figure 4(b), adapting scale and layout increases the minimum number of landmark centers in the view frustum from 4 to 152 with the same 432 identities. Appendix C.2 provides the coverage analysis protocol.

Geometric readout. An identity's 3D position depends on the configuration, so geometric recovery requires both landmark identities and their spatial configuration. ViSPaC encodes the scale, grid configuration, and layout in three colored markers within the visual-pose frames (Figure 2(c)), allowing recovery of $\hat{\Lambda}$ from the signal itself. For each accepted landmark, its color determines a discrete index, its image position provides a 2D observation, and $\Phi_{\hat{\Lambda}}$ maps the index to the corresponding 3D point. These correspondences and the known intrinsics K_t enable robust PnP estimation (Fischler & Bolles, 1981; Terzakis & Lourakis, 2020). Inverting the estimated world-to-camera transform yields $\hat{G}_t$ in the coordinate frame and scale of the declared reference structure, following

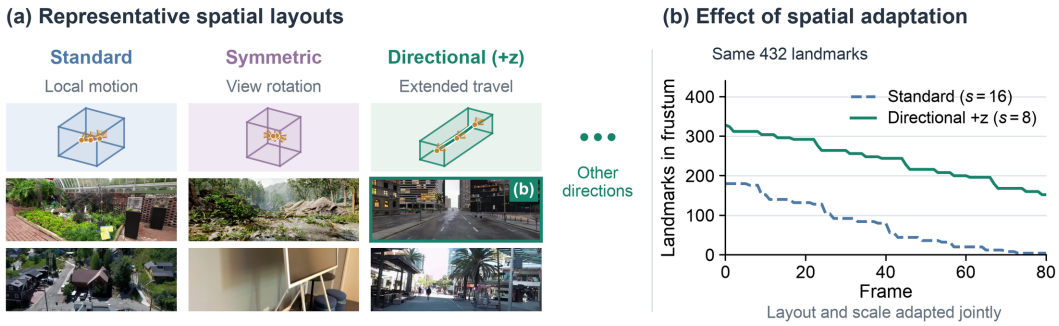

Figure 4: Spatial configurations and coverage. (a) Representative layouts with schematic camera paths and example scenes. (b) For the marked street example in (a), adapting layout and scale retains more landmark centers in the view frustum with the same 432 identities. Counts precede rasterization and occlusion.

the convention in Section 3.1. Appendix A.4 specifies correspondence filtering and pose-estimation settings.

3.4 Joint Modeling of Video and Pose

Joint transformer. Following latent diffusion (Rombach et al., 2022), the shared VAE yields $z_V = \mathcal{E}(V)$ and $z_S = \mathcal{E}(S)$ on matching spatiotemporal grids. Separate input and output projections connect these streams to a pretrained video transformer (Figure 2(b)). Each block attends jointly over their tokens:

$$[H_V; H_S] = \mathrm{Attn}([Q_V; Q_S], [K_V; K_S], [V_V; V_S]), \tag{6}$$

where $[\cdot;\cdot]$ denotes token concatenation. Modality embeddings and spatiotemporal rotary embeddings (Su et al., 2024) distinguish streams and positions. Both streams receive text cross-attention and separate noise conditioning.

Training objective. P2V predicts RGB latents from visual pose; V2P reverses these roles. For target latents z with d entries, visual conditions c, text y, and Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, flow matching (Lipman et al., 2022) trains v_θ to predict velocity $\epsilon - z$. With $\sigma \in [0, 1]$, we average squared error over tasks, noise levels, and samples:

$$\begin{aligned} z^\sigma &= (1-\sigma)z + \sigma\epsilon, \\ \mathcal{L} &= \mathbb{E}_{(z,c,y),\sigma,\epsilon}\left[\frac{1}{d}\left\|v_\theta(z^\sigma, c; \sigma, y) - (\epsilon - z)\right\|_2^2\right]. \end{aligned} \tag{7}$$

4 Experiments

4.1 Experimental Setup

Datasets. We train ViSPaC on approximately 121K clips from DL3DV (Ling et al., 2024), RealEstate10K (Zhou et al., 2018), ScanNet++ (Yeshwanth et al., 2023), MatrixCity (Li et al., 2023), TartanAir v1 and v2 (Wang et al., 2020), TartanGround (Patel et al., 2025), PointOdyssey (Zheng et al., 2023), and Stereo4D (Jin et al., 2025). We evaluate camera control on 110 held-out DL3DV clips for in-domain evaluation and 63 clips from Tanks and Temples (TNT) (Knapitsch et al., 2017) for out-of-domain evaluation. Training trajectories use an approximately metric scale; Appendix B.1 details data preparation and evaluation splits.

Implementation details. We initialize the joint DiT from Wan2.2-TI2V-5B (Wan et al., 2025) and use its frozen video VAE for both RGB and visual pose. We train for 30K steps, sampling P2V and V2P with equal probability, using the flow-matching objective with a noise-schedule shift factor of 5. Inference uses 50 UniPC sampling steps (Zhao et al., 2023). Appendix B.2 provides further training and inference details.

Table 1: Camera control on DL3DV (Ling et al., 2024). Geometry denotes scene reconstruction at inference. PSNR is in dB, ATE and RPE-t in meters, and RPE-r in degrees. Best results are bold and second-best underlined within each protocol; – indicates an unevaluated setting.

		Standard clip (81 frames)				Short clip (16 frames)			
Method	Geometry	PSNR↑	ATE↓	RPE-t↓	RPE-r↓	PSNR↑	ATE↓	RPE-t↓	RPE-r↓
Gen3C	Explicit	14.41	0.02214	0.01348	0.19338	18.35	0.01114	0.01396	0.19853
Uni3C	Explicit	13.82	0.03172	0.01044	0.16471	17.30	0.01027	0.00894	0.16463
Lyra-2	Explicit	<u>14.67</u>	<u>0.00986</u>	<u>0.00561</u>	<u>0.06589</u>	<u>18.88</u>	<u>0.00458</u>	<u>0.00558</u>	<u>0.06773</u>
RealCam-I2V	Implicit	12.75	0.08686	0.01692	0.20743	16.14	0.01687	0.01484	0.19170
CamI2V	Implicit	–	–	–	–	16.23	0.00910	0.00793	0.15903
Geometry Forcing	Implicit	11.28	0.03620	0.00837	0.12027	14.28	0.00793	0.00766	0.12411
SCoPE	Implicit	11.62	0.09934	0.01643	0.31264	14.22	0.01502	0.01176	0.30439
ViSPaC (ours)	Implicit	**15.90**	**0.00926**	**0.00361**	**0.04290**	**20.43**	**0.00340**	**0.00358**	**0.04753**

Table 2: Out-of-domain camera control on TNT (Knapitsch et al., 2017). Units, geometry categories, and ranking conventions follow Table 1.

		Standard clip (81 frames)				Short clip (16 frames)			
Method	Geometry	PSNR↑	ATE↓	RPE-t↓	RPE-r↓	PSNR↑	ATE↓	RPE-t↓	RPE-r↓
Gen3C	Explicit	14.58	0.10222	0.02954	0.20067	18.94	0.01806	0.02342	0.19193
Uni3C	Explicit	12.65	0.12173	0.03647	0.29874	16.07	0.01710	0.01638	0.19476
Lyra-2	Explicit	<u>15.45</u>	<u>0.04791</u>	<u>0.01669</u>	<u>0.11433</u>	<u>19.92</u>	<u>0.01130</u>	<u>0.01322</u>	<u>0.09226</u>
RealCam-I2V	Implicit	13.20	0.15488	0.03872	0.17459	17.20	0.02968	0.03152	0.17068
CamI2V	Implicit	–	–	–	–	15.33	0.02366	0.02029	0.38893
Geometry Forcing	Implicit	10.80	0.17827	0.03891	0.26669	13.78	0.01834	0.02348	0.21313
SCoPE	Implicit	11.74	0.18648	0.05537	0.35884	14.72	0.03054	0.02957	0.40124
ViSPaC (ours)	Implicit	**15.76**	**0.03653**	**0.01440**	**0.08170**	**20.47**	**0.01032**	**0.01185**	**0.07470**

Evaluation metrics. We report PSNR against target frames, excluding the conditioning frame. For camera following, frozen Depth Anything 3 (DA3) (Lin et al., 2025) and ViPE (Huang et al., 2025b) estimate trajectories from generated RGB. After Umeyama similarity alignment to the target (Umeyama, 1991), we compute absolute trajectory error (ATE) and consecutive-frame translation and rotation relative pose errors (RPE-t/RPE-r). We average RMSE over estimators and clips, with translation in meters and rotation in degrees. All methods share the evaluation clips and protocol. Appendix B.3 details the metrics and aggregation.

4.2 Camera Control

We compare with methods that use scene reconstruction at inference, including Gen3C (Ren et al., 2025), Uni3C (Cao et al., 2025), and Lyra-2 (Shen et al., 2026), and methods evaluated without scene reconstruction, including RealCam-I2V (Li et al., 2025), Geometry Forcing (Wu et al., 2026), and SCoPE (Yin et al., 2026). CamI2V (Zheng et al., 2024) is additionally evaluated on short clips. Standard clips contain 81 frames; short-clip evaluation uses CamI2V's 16-frame outputs and the first 16 frames of the other outputs, with trajectories re-estimated over this window. Appendices B.4 and B.3 detail baseline configurations and the short-clip protocol.

Quantitative comparison. Tables 1 and 2 show that ViSPaC achieves the lowest camera errors and highest PSNR across both datasets and clip lengths. Compared with Lyra-2, the strongest baseline in these comparisons, ViSPaC reduces standard-clip translation and rotation RPE on DL3DV by 35.7% and 34.9%, respectively, while improving PSNR by 1.23 dB. On out-of-domain TNT, it reduces standard-clip ATE by 23.8% and rotation RPE by 28.5%, showing that the control advantage extends beyond the training domain. The short-clip results, which also include CamI2V, show that the advantage is present over a shorter evaluation window. Together, these results demonstrate accurate camera following and improved pixel fidelity using a synthetic reference structure, without inference-time scene reconstruction. Appendix D.1 reports results by pose estimator.

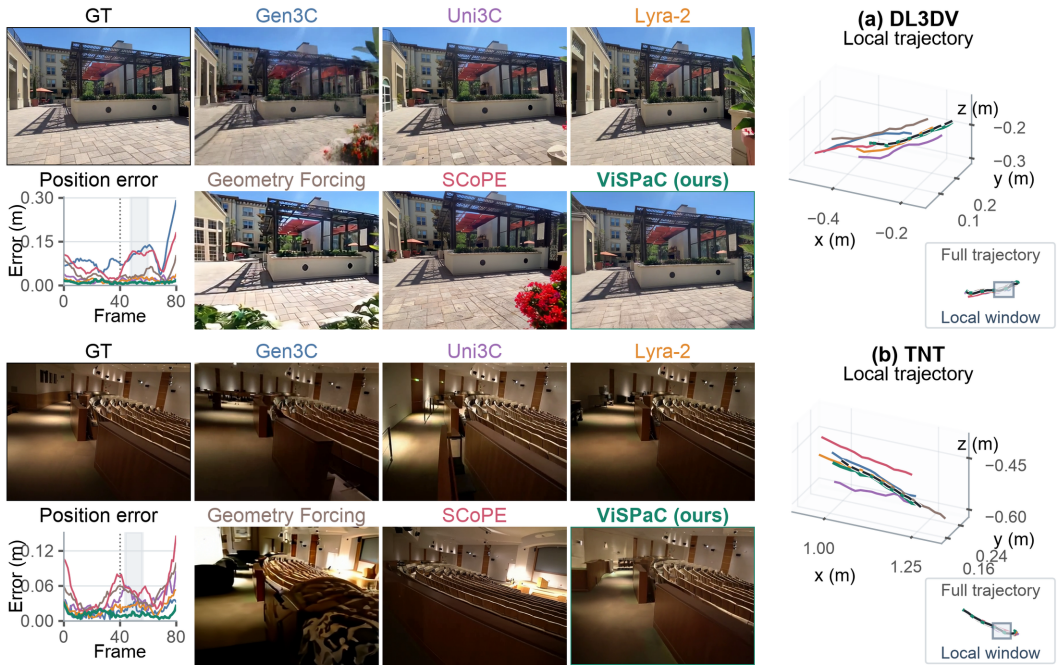

Figure 5: Camera control on DL3DV (top) and TNT (bottom) against prior methods (Ren et al., 2025; Cao et al., 2025; Shen et al., 2026; Wu et al., 2026; Yin et al., 2026). Frames at $t = 40$ accompany full-sequence position errors and enlarged camera paths, using DA3 (Lin et al., 2025) trajectories estimated from generated RGB. Dotted lines mark the displayed frame, shaded intervals select the enlarged segments, and dashed black paths denote GT.

Qualitative comparison. Figure 5 shows how camera-following differences appear in the generated views. In the courtyard, methods differ in the projected positions of the canopy and planter; in the auditorium, differences are visible in the aisle and seating layout. ViSPaC more closely reproduces the target viewpoints in both examples, consistent with its low position errors over the sequence and close agreement with the target in the enlarged trajectory segments. These comparisons illustrate why recognizable scene content alone does not establish accurate camera control: the spatial arrangement of that content must also follow the prescribed viewpoint.

4.3 Pose Representation and Codec Design Ablations

All configurations train for 10K steps on the same 10K-clip subset of the 121K training mixture and are evaluated on DL3DV (Table 3). Generation follows Section 4.1; pose readout uses joint relative-pose AUC at 15° and 30°, following VGGT (Wang et al., 2025a). Appendix B.5 details the protocol.

Pose representation. Within our joint modeling framework, we compare ViSPaC with raw numerical pose tokens and the three-channel Raxel representation (Jang et al., 2026). Raw pose provides a numerical encoding baseline, while Raxel tests an alternative geometry encoding in a visual format. ViSPaC achieves the best generation and readout scores. Compared with Raxel, rotation RPE decreases from 0.399° to 0.070°, and AUC@30 increases from 52.86 to 66.44. Raxel reduces camera-control errors relative to raw pose but does not improve readout AUC, showing that a three-channel format alone does not ensure better performance in both directions. These gains support identifiable landmark projections for control and readout.

Figure 6 shows closer target-view agreement for ViSPaC in both scenes; its estimated camera paths also follow GT more closely than those of Raxel and raw pose.

Table 3: Pose representation and ViSPaC design ablations on DL3DV (Ling et al., 2024). All configurations share the joint framework and training protocol. ViSPaC variants increase the identity count or remove spatial adaptation. Best results are bold.

	Generation				Readout	
Configuration	PSNR↑	ATE↓	RPE-t↓	RPE-r↓	AUC@15↑	AUC@30↑
Pose representations						
Raw pose	12.40	0.1161	0.0201	0.465	30.60	53.01
Raxel	12.35	0.0922	0.0168	0.399	26.18	52.86
ViSPaC variants						
1,404 identities	13.58	0.0398	0.0087	0.121	36.40	58.20
w/o spatial adaptation	13.89	0.0285	0.0067	0.086	43.10	63.80
ViSPaC (full)	**14.10**	**0.0227**	**0.0056**	**0.070**	**46.55**	**66.44**



Figure 6: Pose representation comparison on Showroom (top) and Pavilion (bottom) at $t = 48$, with Raxel (Jang et al., 2026) and raw pose. DA3 (Lin et al., 2025) paths from generated RGB share a projection, scale, and starting origin per scene. Dashed black denotes GT; markers locate the displayed frame.

Codec design. We increase the grid to 1,404 identities or fix the spatial scale and layout while retaining the other design choices. The larger grid tests whether additional landmarks compensate for reduced color separation. Fixing scale and layout tests the value of redistributing a limited identity set across camera views. Both variants underperform full ViSPaC in generation and readout, supporting the contributions of identity budget and spatial allocation. Without spatial adaptation, ATE rises from 0.0227 to 0.0285. These comparisons highlight two distinct aspects of codec design: how many landmark identities the visual representation can distinguish, and where those landmarks are placed in space. Increasing the identity count does not necessarily improve the representation, while retaining the same identities does not eliminate the need for spatial adaptation. The two choices therefore need to be considered together when constructing visual pose signals for camera control.

5 Conclusion

We introduced ViSPaC, a visual pose codec that expresses camera motion through identifiable projections of a synthetic 3D reference structure. By separating color identity from spatial allocation, it addresses coupled constraints on identity distinguishability under visual compression and coverage across views. ViSPaC achieves state-of-the-art camera control on in-domain and out-of-domain benchmarks, while also enabling video-conditioned pose recovery. Controlled comparisons support the representation and its choices of identity budget and spatial allocation. These findings highlight the importance of organizing visual pose to connect camera motion with video dynamics.

REPRODUCIBILITY STATEMENT

Section 3 describes the ViSPaC representation and joint modeling framework, with detailed codec construction and decoding procedures in Appendix A. Appendix B.1 documents the training data composition, preprocessing, scale alignment, and evaluation splits. Appendix B.2 provides the model configuration and training and inference settings, while Appendices B.3–B.5 specify the evaluation metrics, baseline configurations, and ablation protocols. Appendix C describes the analyses of landmark identity distinguishability under visual compression and projected spatial coverage.

ETHICS STATEMENT

This work studies camera representation for controllable video generation. Generated videos can be used to create misleading depictions of events or places. Outputs should be identified as synthetic when presented outside research evaluation.

AI USE STATEMENT

AI tools were used for writing assistance and literature/name checks under human direction. All technical claims, experiments, and final text are the authors' responsibility.

REFERENCES

Alexey Bochkovskiy, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. In *International Conference on Learning Representations*, volume 2025, pp. 75602–75637, 2025.

Chenjie Cao, Jingkai Zhou, Shikai Li, Jingyun Liang, Chaohui Yu, Fan Wang, Xiangyang Xue, and Yanwei Fu. Uni3c: Unifying precisely 3d-enhanced camera and human motion controls for video generation. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–12, 2025.

Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981.

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.

Chun-Hao Paul Huang, Niloy Mitra, Hyeonho Jeong, Jae Shin Yoon, and Duygu Ceylan. Jog3r: Towards 3d-consistent video generators. *arXiv preprint arXiv:2501.01409*, 2025a.

Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, et al. Vipe: Video pose engine for 3d geometric perception. *arXiv preprint arXiv:2508.10934*, 2025b.

Wonbong Jang, Shikun Liu, Soubhik Sanyal, Juan Camilo Perez, Kam Woh Ng, Sanskar Agrawal, Juan-Manuel Perez-Rua, Yiannis Douratsos, and Tao Xiang. Rays as pixels: Learning a joint distribution of video and camera trajectories. In *Forty-third International Conference on Machine Learning*, 2026.

Linyi Jin, Richard Tucker, Zhengqi Li, David Fouhey, Noah Snavely, and Aleksander Holynski. Stereo4d: Learning how things move in 3d from internet stereo videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10497–10509, 2025.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. *Advances in Neural Information Processing Systems*, 38:15984–16009, 2026.

Teng Li, Guangcong Zheng, Rui Jiang, Shuigen Zhan, Tao Wu, Yehao Lu, Yining Lin, Chuanyun Deng, Yepan Xiong, Min Chen, et al. Realcam-i2v: Real-world image-to-video generation with interactive complex camera control. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28785–28796. IEEE, 2025.

Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. Matrixcity: A large-scale city dataset for city-scale neural rendering and beyond. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3182–3192. IEEE, 2023.

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

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22160–22169. IEEE, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Manthan Patel, Fan Yang, Yuheng Qiu, Cesar Cadena, Sebastian Scherer, Marco Hutter, and Wenshan Wang. Tartanground: A large-scale dataset for ground robot perception and navigation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 20524–20531. IEEE, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6121–6132. IEEE, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Tianchang Shen, Sherwin Bahmani, Kai He, Sangeetha Grama Srinivasan, Tianshi Cao, Jiawei Ren, Ruilong Li, Zian Wang, Nicholas Sharp, Zan Gojcic, et al. Lyra 2.0: Explorable generative 3d worlds. *arXiv preprint arXiv:2604.13036*, 2026.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

George Terzakis and Manolis Lourakis. A consistently fast and globally optimal solution to the perspective-n-point problem. In *European Conference on Computer Vision*, pp. 478–494. Springer, 2020.

Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 13(4):376–380, 1991.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

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

Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5261–5271. IEEE, 2025b.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709. IEEE, 2024a.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4909–4916. IEEE, 2020.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

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.

Haoyu Wu, Diankun Wu, Tianyu He, Junliang Guo, Yang Ye, Yueqi Duan, and Jiang Bian. Geometry forcing: Marrying video diffusion and 3d representation for consistent world modeling. In *International Conference on Learning Representations*, volume 2026, pp. 119186–119209, 2026.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in neural information processing systems*, 37:21875–21911, 2024.

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

Minghao Yin, Jiahao Lu, Wenbo Hu, Wang Zhao, Ying Shan, and Kai Han. Scope: Sightline-coordinate positional encoding for video diffusion transformers. 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.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:49842–49869, 2023.

Guangcong Zheng, Teng Li, Rui Jiang, Yehao Lu, Tao Wu, and Xi Li. Cami2v: Camera-controlled image-to-video diffusion model. *arXiv preprint arXiv:2410.15957*, 2024.

Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19798–19808. IEEE, 2023.

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

APPENDIX: SUPPLEMENTARY MATERIAL

A CODEC CONSTRUCTION AND DECODING

We use the first-camera coordinate system and camera-to-world poses $G_t = (R_t, \mathbf{p}_t)$ defined in Section 3.1. Pixel coordinates follow the OpenCV convention, with the camera axes pointing right, down, and forward.

A.1 LANDMARK IDENTITIES AND SPATIAL CODEBOOK

Identity grid. We use $[c_{\min}, c_{\max}] = [-0.6, 0.84]$ within the VAE input range $[-1, 1]$. The sparse grid has dimensions $(6, 5, 7)$ and 210 identities; the dense grid has dimensions $(8, 6, 9)$ and 432 identities. Grid selection is specified in Appendix A.2.

Spatial codebook. Table 4 lists the normalized origin $\mathbf{b}_q$ and span $\mathbf{h}_q$ of the eight layouts. The scale is selected from $\mathcal{S} = \{0.5, 1, 2, 4, 8, 16\}$ and multiplies both quantities. All coordinates use the input trajectory's translation units.

We perturb each grid point by a deterministic offset bounded by 15% of the spacing along each axis to reduce regular grid alignments. The offsets are reproduced during decoding, ensuring identical reference geometry at both ends.

A.2 TRAJECTORY-DEPENDENT CONFIGURATION SELECTION

During encoding, we use the trajectory G and intrinsics K to select the grid density and disk size, followed by the spatial scale and layout. This yields a configuration $\Lambda = (s, \mathbf{n}, q)$ and a disk-size parameter ρ for the clip.

Grid density and disk size. We use larger disks and a sparser grid for clips with larger rotational motion. To set these parameters, we measure the mean inter-frame rotation angle scaled by the mean horizontal focal length:

$$v = \frac{\bar{f}_x}{T-1} \sum_{t=0}^{T-2} \arccos\left(\mathrm{clip}\left(\frac{\mathrm{tr}(R_t^\mathsf{T} R_{t+1}) - 1}{2}, -1, 1 \right) \right), \qquad \bar{f}_x = \frac{1}{T} \sum_{t=0}^{T-1} f_{x,t}. \tag{8}$$

The scalar v is a rotation-based image-motion proxy in pixels per frame. The minimum disk radius is

$$\rho = \mathrm{clip}\big(5 + 0.3(v - 2), 5, 11\big) \text{ pixels}. \tag{9}$$

We use $\mathbf{n} = (8, 6, 9)$ when $\rho < 9$ and $(6, 5, 7)$ otherwise. Subsequent layout comparisons use the same grid and color identities.

Spatial configuration selection. For each layout q, we select the smallest scale $s_q \in \mathcal{S}$ that accommodates the camera path with a spatial margin and provides a minimum forward extent. This limits landmark spacing at the fixed identity count. The conditions are, componentwise for all frames,

$$s_q(\mathbf{b}_q + 0.5\mathbf{1}) \le 1.3\mathbf{p}_t \le s_q(\mathbf{b}_q + \mathbf{h}_q - 0.5\mathbf{1}), \qquad s_q(b_{q,z} + h_{q,z}) \ge 4. \tag{10}$$

The containment inequalities expand camera-center offsets by 1.3 and reserve a $0.5 s_q$ inset along each axis. The forward-extent floor uses the input translation units.

Coverage is measured by projecting unperturbed grid centers at frames $0, 4, 8, \ldots$ using K_0. For an image of width W and height H, we count centers with camera depth greater than 0.25 and image coordinates in $[0, W) \times [0, H)$, before rasterization, occlusion, and configuration-marker overlays.

Algorithm 1 specifies scale selection and candidate ranking, including the fallback rules. The selected configuration remains fixed throughout the clip.

Table 4: ViSPaC spatial codebook in the first-camera coordinate system. Multiplying the origin and span by $s \in \mathcal{S}$ specifies the reference volume; grid dimensions determine the landmark spacing within it.

q	Layout	Origin $\mathbf{b}_q$	Span $\mathbf{h}_q$
0	Standard	$(-3.5, -2.5, -2)$	$(7, 5, 8)$
1	Symmetric	$(-4, -4, -4)$	$(8, 8, 8)$
2	Extended $+x$	$(-2, -3.5, -2.5)$	$(22, 7, 5)$
3	Extended $-x$	$(-20, -3.5, -2.5)$	$(22, 7, 5)$
4	Extended $+y$	$(-3.5, -2, -2.5)$	$(7, 22, 5)$
5	Extended $-y$	$(-3.5, -20, -2.5)$	$(7, 22, 5)$
6	Extended $+z$	$(-3.5, -2.5, -2)$	$(7, 5, 22)$
7	Extended $-z$	$(-3.5, -2.5, -20)$	$(7, 5, 22)$

Algorithm 1: Trajectory-dependent spatial configuration selection

Input: Trajectory $\{R_t, \mathbf{p}_t\}_{t=0}^{T-1}$ and intrinsics $\{K_t\}_{t=0}^{T-1}$
Output: Configuration $\Lambda = (s, \mathbf{n}, q)$ and disk-size parameter ρ

Compute v and ρ using Eqs. (8)–(9);
Set $\mathbf{n} = (8, 6, 9)$ if $\rho < 9$, otherwise $(6, 5, 7)$;
for $q = 0, \ldots, 7$ **do**
  Select the smallest $s_q \in \mathcal{S}$ satisfying Eq. (10); use 16 if none exists;
  Set $\mathbf{d}_q = s_q \mathbf{h}_q / (\mathbf{n} - \mathbf{1})$;
  Project unperturbed centers at frames $0, 4, 8, \ldots$ using K_0;
  Set $a_q = 1$ if containment holds and each sampled view contains at least 12 centers; otherwise $a_q = 0$;
  Set $\mathbf{k}_q = (1 - a_q, \mathbb{1}[q \neq 0], \max_a d_{q,a}, q)$;
end
Choose $q^\star$ with the lexicographically smallest $\mathbf{k}_q$;
return $\Lambda = (s_{q^\star}, \mathbf{n}, q^\star)$ *and* ρ;

A.3 Rendering and Configuration Encoding

Landmark rasterization. We project the reference landmarks using the per-frame intrinsics K_t and Eq. (3), discarding points with camera depth $z_{t,\mathbf{u}} \leq 0.25$. Each remaining landmark whose disk intersects the image is rendered as an opaque disk in its assigned identity color. The clip-level parameter ρ selected in Appendix A.2 sets the minimum radius; the radius of each projected landmark varies with depth as

$$r_{t,\mathbf{u}} = \mathrm{clip}\left(\frac{1.4\rho}{z_{t,\mathbf{u}}}, \rho, 2\rho\right), \tag{11}$$

where depth is expressed in the input trajectory's translation units. Disks are drawn from far to near, so nearer landmarks occlude farther ones. Background values lie in $[-1, -0.85]$, below the foreground threshold used for landmark extraction.

Configuration markers. To let the decoder reconstruct the reference geometry, we encode the scale, grid choice, and layout as three fixed-position color markers. We reserve a black 64×64 region in the top-left corner and draw radius-9-pixel disks at $(18, 18)$, $(46, 18)$, and $(18, 46)$, respectively, after rendering the landmarks.

Using the identity color bounds $c_{\min}$ and $c_{\max}$, we define the ordered configuration palette

$$\begin{aligned}\mathcal{C} = \big(&(c_{\max}, c_{\min}, c_{\min}), (c_{\min}, c_{\max}, c_{\min}), (c_{\min}, c_{\min}, c_{\max}),\\ &(c_{\max}, c_{\max}, c_{\min}), (c_{\max}, c_{\min}, c_{\max}), (c_{\min}, c_{\max}, c_{\max}),\\ &(c_{\max}, c_{\max}, c_{\max}), (-0.1, -0.1, -0.1)\big).\end{aligned} \tag{12}$$

With zero-based indexing, the six scales in ascending order use $\mathcal{C}_0, \ldots, \mathcal{C}_5$; the dense and sparse grids use $\mathcal{C}_6$ and $\mathcal{C}_7$; and layout q uses $\mathcal{C}_q$. The same three markers are repeated in every frame of the clip.

A.4 Correspondence Extraction and Pose Recovery

Configuration readout. We recover the scale, grid choice, and layout from the three markers defined in Appendix A.3. At each marker location, we take the componentwise median RGB value in a centered 9×9 window and select the nearest codeword in the corresponding palette by Euclidean distance. The resulting configuration $\hat{\Lambda}$ reconstructs the reference landmarks, including their deterministic perturbations. The marker region is excluded from subsequent landmark extraction.

Landmark extraction. We extract candidate disks from the foreground and filter components whose size or shape makes their centers unreliable. Foreground pixels have a maximum RGB channel greater than -0.75 and are grouped into 8-connected components. We reject components with area outside $[8, 1200]$ pixels, those touching the image boundary, and those with area below $0.55\pi r_{\mathrm{enc}}^2$, where r_{enc} is the minimum enclosing-circle radius.

For each retained component, let $D(\mathbf{q})$ denote its interior Euclidean distance transform. Its image center is the weighted mean $\sum_{\mathbf{q}} D(\mathbf{q})\mathbf{q} / \sum_{\mathbf{q}} D(\mathbf{q})$, giving greater weight to interior pixels. We estimate its identity color by the componentwise median over core pixels satisfying $D(\mathbf{q}) \geq 0.6 \max D$, reducing the influence of boundary color mixing.

Identity matching. We convert each observed color $\hat{\mathbf{c}}$ to a landmark index by inverting Eq. (2) and rounding to the nearest color levels:

$$\boldsymbol{\eta} = \frac{\hat{\mathbf{c}} - c_{\min}\mathbf{1}}{c_{\max} - c_{\min}} \odot (\hat{\mathbf{n}} - \mathbf{1}), \qquad \hat{\mathbf{u}} = \mathrm{round}(\boldsymbol{\eta}). \tag{13}$$

Indices outside the decoded grid and matches with $\|\boldsymbol{\eta} - \hat{\mathbf{u}}\|_\infty > 0.42$ are discarded. Each accepted index identifies the 3D point $\Phi_{\hat{\Lambda}}(\hat{\mathbf{u}})$, which is paired with the component's image center to form a 2D–3D correspondence.

Pose estimation. Using the supplied intrinsics K_t and a pinhole camera model without lens distortion, we estimate pose from at least four correspondences with SQPnP inside RANSAC (Fischler & Bolles, 1981; Terzakis & Lourakis, 2020). We use a 4-pixel reprojection threshold, 2,000 iterations, and confidence 0.9999. Estimates with at least four inliers are refined by Levenberg–Marquardt minimization of their reprojection errors.

B Experimental Setup and Evaluation

B.1 Datasets and Preprocessing

Training data. The training mixture contains 121,255 clips after preprocessing and filtering from the datasets listed in Section 4.1. Each example pairs RGB frames and a text description with camera parameters and the corresponding visual-pose sequence. The DL3DV-Evaluation split and TNT are excluded from training.

Near-metric scale alignment. DL3DV uses approximately metric trajectories estimated by ViPE (Huang et al., 2025b). For RealEstate10K (Zhou et al., 2018) and Stereo4D (Jin et al., 2025), we rescale source translations using a metric Depth Anything 3 model (Lin et al., 2025). The remaining datasets retain their provided metric annotations.

For each source sequence, let $\mathbf{p}_t^{\mathrm{src}}$ and $\mathbf{p}_t^{\mathrm{DA3}}$ denote translations relative to the first camera in the source and DA3 trajectories. We estimate one positive scale factor

$$\alpha = \mathrm{median}_{t \in \mathcal{J}} \frac{\|\mathbf{p}_t^{\mathrm{DA3}}\|_2}{\|\mathbf{p}_t^{\mathrm{src}}\|_2}, \qquad \mathbf{p}_t = \alpha \mathbf{p}_t^{\mathrm{src}}, \tag{14}$$

where $\mathcal{J}$ contains frames whose source displacement exceeds 5% of the sequence's maximum displacement. We require at least ten such frames to estimate the factor; low-motion sequences without a stable estimate retain their source scale. Sequences with unusable predictions or severe disagreement between source and DA3 motion magnitudes are excluded.

Video and camera preprocessing. Training clips contain 81 temporally matched RGB and camera frames at 480×640. Intrinsics are adjusted consistently with image resizing and cropping. We express the scale-aligned poses in the first-camera coordinate system using Eq. (1), then render ViSPaC from these poses and the adjusted intrinsics.

Evaluation clips. For in-domain evaluation, we use all 55 scenes of the official DL3DV-Evaluation split, which is disjoint from DL3DV-10K (Ling et al., 2024). Each scene contributes two non-overlapping 81-frame clips: the middle clip and an earlier clip at the quarter position of the chronologically ordered complete clips, yielding 110 clips. For out-of-domain evaluation, we use 21 scenes from Tanks and Temples (TNT) (Knapitsch et al., 2017): seven from its training subset, eight from its intermediate subset, and six from its advanced subset. Three fixed clips per scene yield 63 evaluation clips. All methods receive the same clips and target trajectories.

B.2 Model Training and Inference

Model and training. We initialize the RGB–pose transformer from Wan2.2-TI2V-5B (Wan et al., 2025) and freeze the shared video VAE. The first six transformer blocks use stream-specific query/key/value, attention-output, and feed-forward projections; later blocks share these projections. We train for 30K steps with AdamW (Loshchilov & Hutter, 2017), a global batch size of 128, and a base learning rate of 4×10^{-5} using the DeepSpeed WarmupDecayLR schedule. P2V and V2P batches are sampled with equal probability. The flow-matching noise schedule uses a shift factor of 5, as in Section 4.1.

Inference. ViSPaC generates 81-frame clips at 480×640 using 50 UniPC steps (Zhao et al., 2023) and classifier-free guidance 1.0. For camera-controlled generation, the visual-pose latents and first RGB latent remain fixed while the target RGB latents are initialized with Gaussian noise and iteratively denoised. For video-conditioned pose recovery, the RGB video conditions the generation of a ViSPaC sequence, which is decoded into camera poses using the geometric readout in Appendix A.4. If direct readout fails, we retry generation with a synthetic first-frame visual pose fixed, testing codebook configurations from the standard layout and using the first-frame configuration for sequence decoding.

B.3 Camera-Control Evaluation

Camera following. Frozen Depth Anything 3 (DA3) (Lin et al., 2025) and ViPE (Huang et al., 2025b) independently estimate camera trajectories from generated RGB. For each estimator and clip, we fit an Umeyama similarity transform (Umeyama, 1991) between the estimated and target camera centers and apply it to the estimated trajectory. Let $\tilde{G}_t$ and $G_t^\star$ denote the aligned and target camera-to-world poses, with camera centers $\tilde{\mathbf{p}}_t$ and $\mathbf{p}_t^\star$. For a clip of T frames, absolute trajectory error is

$$e_t = \|\tilde{\mathbf{p}}_t - \mathbf{p}_t^\star\|_2, \qquad \mathrm{ATE} = \sqrt{\frac{1}{T}\sum_{t=0}^{T-1} e_t^2}. \tag{15}$$

Relative pose errors compare consecutive-frame motions. We define

$$\begin{aligned} \Delta\tilde{G}_t &= \tilde{G}_t^{-1}\tilde{G}_{t+1}, \qquad \Delta G_t^\star = (G_t^\star)^{-1}G_{t+1}^\star, \\ E_t &= (\Delta G_t^\star)^{-1}\Delta\tilde{G}_t, \qquad t = 0, \ldots, T-2. \end{aligned} \tag{16}$$

Writing E_t as a rotation $R_{E,t}$ and translation $\mathbf{p}_{E,t}$, we compute

$$\begin{aligned} \mathrm{RPE\text{-}t} &= \sqrt{\frac{1}{T-1}\sum_{t=0}^{T-2} \|\mathbf{p}_{E,t}\|_2^2}, \\ \mathrm{RPE\text{-}r} &= \sqrt{\frac{1}{T-1}\sum_{t=0}^{T-2} \theta(R_{E,t})^2}, \end{aligned} \tag{17}$$

where $\theta(R)$ is the geodesic rotation angle in degrees. ATE and RPE-t are reported in meters. Similarity alignment removes global scale discrepancies, so these metrics evaluate aligned camera following rather than absolute metric-scale recovery.

Appearance and aggregation. We compute PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018) against the target RGB frames at a common 480×640 resolution and average over frames $1, \ldots, T-1$, excluding the conditioning frame. LPIPS uses the AlexNet backbone. For each camera metric, we average the DA3 and ViPE scores within each clip, then average over clips. Each dataset-level appearance score is the mean of its clip-level scores.

Short-clip protocol. The standard protocol evaluates 81-frame clips; the short-clip protocol evaluates their first 16 consecutive frames. CamI2V directly generates these 16 frames, while methods producing 81 frames use the prefix of the same generated video. We rerun both DA3 and ViPE on the 16-frame inputs, independently align the resulting trajectories to the corresponding target prefix, and recompute the metrics above with $T = 16$. Baseline configurations are detailed in Appendix B.4.

B.4 Baseline Configurations

For all camera-control baselines, we use the codebases and checkpoints released by their authors. Intrinsics are adjusted for input resizing, and outputs are evaluated at a common resolution.

Depth and trajectory scale alignment. Gen3C (Ren et al., 2025), Uni3C (Cao et al., 2025), and Lyra-2 (Shen et al., 2026) reconstruct scene geometry at inference. Following Gen3C, we align input translations with the reconstruction scale using conditioning-image depth. For Gen3C and Lyra-2, translations are multiplied by the ratio of median MoGe (Wang et al., 2025b) depth to median metric Depth Anything V2 (Yang et al., 2024) depth. Uni3C uses metric depth from Depth Pro (Bochkovskiy et al., 2025). This input alignment uses no ground-truth depth and is separate from the evaluation alignment in Appendix B.3.

B.5 Pose Representation and Codec Design Ablations

Compared configurations. Table 3 compares raw numerical pose tokens, the three-channel Raxel ray-field representation (Jang et al., 2026), and ViSPaC within the same joint modeling framework. Two ViSPaC variants isolate choices of identity budget and spatial configuration.

Raw pose tokens. For each canonical pose $G_t = (R_t, \mathbf{p}_t)$, we form a 13-dimensional vector containing the first two columns of R_t, the translation $\mathbf{p}_t$ in the input trajectory's units, and intrinsics $(f_x/W, f_y/H, c_x/W, c_y/H)$, where W, H are the image dimensions. The parameter sequence bypasses the VAE. Following its temporal grouping, the first frame forms one group and subsequent frames are grouped in fours, yielding 21 pose tokens for an 81-frame clip. A two-layer MLP embeds each group, with learned temporal embeddings in place of spatial rotary embeddings. These tokens participate in joint attention with video tokens. A two-layer MLP reads out per-frame parameters; Gram–Schmidt orthogonalization recovers the rotation, while translation is read out directly in the input units.

Raxel. At pixel $\mathbf{x} = (x, y, 1)^\top$, let $\mathbf{m}_t(\mathbf{x}) = K_t^{-1}\mathbf{x}/\|K_t^{-1}\mathbf{x}\|_2$ be the unit camera ray. Following Rays as Pixels (Jang et al., 2026), we encode $\mathbf{r}_t(\mathbf{x}) = R_t\mathbf{m}_t(\mathbf{x}) + \mathbf{p}_t$ in three channels. For our metric-scale data, a single dataset-wide normalization factor maps raxel values to $[-1, 1]$, preserving relative scales across clips. Within our shared framework, the floating-point sequences use the same resolution, frozen Wan VAE, and latent patch embedding as ViSPaC. For pose readout, we VAE-decode the predicted latents and undo the global normalization. Following the reference-frame Procrustes formulation, we recover each relative pose by fitting the predicted first-frame raxel map to the current predicted map through centering and SVD.

Codec variants. The larger-grid variant uses $\mathbf{n} = (12, 9, 13)$, giving 1,404 identities, while retaining the full spatial adaptation procedure. The fixed-configuration variant retains ViSPaC's grid-selection policy but uses the standard layout with one global scale. We select this scale from $\mathcal{S}$ using only the training subset, maximizing the mean per-clip minimum coverage defined in Appendix C.2; ties favor the smaller scale. Both variants retain the color-coding range, deterministic perturbation rule, disk-size policy, rasterizer, and geometric decoder. Each model is trained and evaluated with its corresponding representation.

Table 5: Landmark identity mismatch across grid resolutions, before and after frozen Wan VAE compression (Wan et al., 2025).

Grid	Points	Identity mismatch (%)	
		Clean	VAE
$4 \times 3 \times 5$	60	0.00	0.00
$6 \times 5 \times 7$	210	0.00	0.19
$8 \times 6 \times 9$	432	0.17	2.98
$10 \times 8 \times 11$	880	0.55	9.73
$12 \times 9 \times 13$	1404	1.00	18.53

Training protocol. All configurations are trained for 10K steps on the same 10K-clip subset of the approximately 121K-clip training mixture and evaluated on DL3DV. They share the joint architecture, initialization, data order, optimization and inference settings, and equal P2V/V2P task sampling.

Generation and pose readout. Generation follows the camera-control evaluation protocol in Appendix B.3. Video-conditioned pose readout is evaluated directly from the predicted pose signals. Following the joint relative-pose AUC evaluation used by VGGT (Wang et al., 2025a), we evaluate all unordered frame pairs within each clip. For pair (i, j), let e_{ij}^R and e_{ij}^t denote the relative-rotation and translation-direction angular errors, respectively. The translation-direction error accounts for the 180° sign ambiguity. We use the joint error $e_{ij} = \max\{e_{ij}^R, e_{ij}^t\}$ and report AUC at 15° and 30°. Following VGGT, AUC is the mean of the normalized cumulative error histogram with 1° bins up to the threshold, multiplied by 100. We compute AUC per clip and average across clips.

C Additional Representation Analyses

These analyses examine the two design considerations underlying ViSPaC: selecting a landmark identity budget compatible with visual compression, and allocating those landmarks to maintain coverage across camera views. Section C.1 examines identity confusion as the grid grows, while Section C.2 illustrates how spatial configuration changes coverage at a fixed identity count.

C.1 Identity Capacity Under Visual Compression

We examine how increasing the landmark identity count affects identity distinguishability before and after visual compression. This analysis informs the choice of identity budget by measuring color-identity confusion among matched landmark detections.

Probe construction. We evaluate five grid resolutions, ranging from 60 to 1,404 landmark identities (Table 5), on 100 clips each from DL3DV (Ling et al., 2024), ScanNet++ (Yeshwanth et al., 2023), and TartanAir v2 (Wang et al., 2020). For each grid, we select the scale and layout using Appendix A.2 and render the visual-pose sequence using Appendix A.3. We compare the rendered sequence directly (Clean) with its reconstruction through the frozen Wan VAE (VAE).

Measurements. We extract landmarks using Appendix A.4, with the known rendering configuration supplied for this probe. Detections are landmark candidates retained by the component and color filters. We assess their identities against the reference landmark projections: a detection is correct if its decoded identity corresponds to a reference landmark and its center lies within the larger of that landmark's projected radius and 1.5 pixels of the reference center. Otherwise, a detection near a reference projection is an identity mismatch; one remote from the reference projections is spurious. The framewise identity mismatch rate is the number of mismatches divided by the total number of correct detections and mismatches, excluding spurious detections.

We average per-clip temporal medians of the defined mismatch rates across clips. Table 5 reports the results plotted in Figure 3(b).

Table 6: Additional camera-control results on DL3DV (Ling et al., 2024). Appearance metrics compare generated and target RGB; camera metrics are reported separately for DA3 (Lin et al., 2025) and ViPE (Huang et al., 2025b). PSNR is in dB, ATE and RPE-t in meters, and RPE-r in degrees. Best results are bold and second-best underlined within each column; – denotes an unevaluated setting.

	Standard clip (81 frames)			Short clip (16 frames)		
	Appearance					
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Gen3C	14.41	0.3953	0.3879	18.35	0.5637	0.2114
Uni3C	13.82	0.3766	0.4000	17.30	0.5126	0.2269
Lyra-2	<u>14.67</u>	<u>0.4069</u>	<u>0.3474</u>	<u>18.88</u>	<u>0.5873</u>	<u>0.1656</u>
RealCam-I2V	12.75	0.3170	0.4343	16.14	0.4525	0.2415
CamI2V	–	–	–	16.23	0.4570	0.3228
Geometry Forcing	11.28	0.2865	0.5916	14.28	0.3907	0.4219
SCoPE	11.62	0.2763	0.5309	14.22	0.3604	0.3451
ViSPaC (ours)	**15.90**	**0.4582**	**0.3096**	**20.43**	**0.6488**	**0.1473**
	Camera control: DA3					
Method	ATE↓	RPE-t↓	RPE-r↓	ATE↓	RPE-t↓	RPE-r↓
Gen3C	0.02242	0.01355	0.19390	0.01139	0.01405	0.19978
Uni3C	0.03128	0.00991	0.16444	0.01029	0.00901	0.16704
Lyra-2	<u>0.01055</u>	<u>0.00605</u>	<u>0.07134</u>	<u>0.00531</u>	<u>0.00628</u>	<u>0.07567</u>
RealCam-I2V	0.08612	0.01726	0.20918	0.01700	0.01497	0.19661
CamI2V	–	–	–	0.00872	0.00778	0.15591
Geometry Forcing	0.03745	0.00892	0.12907	0.00864	0.00820	0.13962
SCoPE	0.09780	0.01639	0.31334	0.01525	0.01173	0.30432
ViSPaC (ours)	**0.00987**	**0.00431**	**0.05336**	**0.00411**	**0.00447**	**0.05987**
	Camera control: ViPE					
Method	ATE↓	RPE-t↓	RPE-r↓	ATE↓	RPE-t↓	RPE-r↓
Gen3C	0.02187	0.01341	0.19286	0.01089	0.01387	0.19729
Uni3C	0.03217	0.01096	0.16498	0.01026	0.00886	0.16222
Lyra-2	<u>0.00916</u>	<u>0.00518</u>	<u>0.06044</u>	<u>0.00386</u>	<u>0.00487</u>	<u>0.05979</u>
RealCam-I2V	0.08760	0.01658	0.20567	0.01673	0.01470	0.18679
CamI2V	–	–	–	0.00949	0.00807	0.16215
Geometry Forcing	0.03495	0.00783	0.11147	0.00722	0.00712	0.10861
SCoPE	0.10089	0.01647	0.31194	0.01479	0.01179	0.30446
ViSPaC (ours)	**0.00866**	**0.00291**	**0.03244**	**0.00268**	**0.00269**	**0.03519**

Results. After VAE reconstruction, increasing the grid from 432 to 1,404 identities raises the mismatch rate from 2.98% to 18.53%. Clean mismatch rates remain at or below 1.00% throughout the sweep. These results support limiting the identity budget under visual compression; ViSPaC uses the 210- and 432-identity grids in the lower-mismatch range.

C.2 Projected Coverage Across Spatial Configurations

We examine how spatial allocation changes the number of landmarks available in camera views when the identity count is fixed. The example illustrates the role of scale and layout in maintaining projected coverage along a trajectory.

Coverage measurement. For each 81-frame trajectory, we count unperturbed landmark centers that project inside the image with camera depth greater than 0.25, then compute the minimum count over the clip. These counts measure geometric coverage before rasterization, occlusion, and visual compression. The analysis uses all frames and their individual intrinsics.

Configurations in Figure 4. Figure 4(b) shows the forward-moving MatrixCity (Li et al., 2023) example, comparing the standard layout at $s = 16$ with the $+z$-extended layout at $s = 8$, both with 432 identities.

Table 7: Additional camera-control results on TNT (Knapitsch et al., 2017). Appearance metrics compare generated and target RGB; camera metrics are reported separately for DA3 (Lin et al., 2025) and ViPE (Huang et al., 2025b). PSNR is in dB, ATE and RPE-t in meters, and RPE-r in degrees. Best results are bold and second-best underlined within each column; – denotes an unevaluated setting.

	Standard clip (81 frames)			Short clip (16 frames)		
	Appearance					
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Gen3C	14.58	0.4143	0.4273	18.94	0.5674	0.2536
Uni3C	12.65	0.3570	0.5014	16.07	0.4666	0.3270
Lyra-2	<u>15.45</u>	<u>0.4453</u>	<u>0.3514</u>	<u>19.92</u>	<u>0.6079</u>	<u>0.1840</u>
RealCam-I2V	13.20	0.3538	0.4466	17.20	0.4906	0.2463
CamI2V	–	–	–	15.33	0.4381	0.3962
Geometry Forcing	10.80	0.2800	0.6625	13.78	0.3565	0.5046
SCoPE	11.74	0.3058	0.5450	14.72	0.3923	0.3596
ViSPaC (ours)	**15.76**	**0.4540**	**0.3368**	**20.47**	**0.6283**	**0.1514**
	Camera control: DA3					
Method	ATE↓	RPE-t↓	RPE-r↓	ATE↓	RPE-t↓	RPE-r↓
Gen3C	0.10659	0.03171	0.19393	0.01864	0.02261	0.16439
Uni3C	0.14491	0.04422	0.29040	0.01578	0.01541	0.18182
Lyra-2	<u>0.04762</u>	<u>0.01706</u>	<u>0.11537</u>	<u>0.01185</u>	**0.01290**	<u>0.09414</u>
RealCam-I2V	0.15175	0.04461	0.17789	0.03409	0.03371	0.15891
CamI2V	–	–	–	0.02467	0.01995	0.46767
Geometry Forcing	0.16412	0.03540	0.19024	0.02017	0.02391	0.18580
SCoPE	0.19114	0.06934	0.32511	0.03045	0.03126	0.34353
ViSPaC (ours)	**0.03730**	**0.01660**	**0.06926**	**0.01161**	<u>0.01333</u>	**0.06626**
	Camera control: ViPE					
Method	ATE↓	RPE-t↓	RPE-r↓	ATE↓	RPE-t↓	RPE-r↓
Gen3C	0.09786	0.02737	0.20741	0.01748	0.02422	0.21946
Uni3C	0.09856	0.02873	0.30709	0.01841	0.01735	0.20770
Lyra-2	<u>0.04820</u>	<u>0.01632</u>	<u>0.11329</u>	<u>0.01075</u>	<u>0.01353</u>	<u>0.09038</u>
RealCam-I2V	0.15802	0.03283	0.17129	0.02526	0.02934	0.18244
CamI2V	–	–	–	0.02265	0.02062	0.31018
Geometry Forcing	0.19242	0.04241	0.34314	0.01652	0.02306	0.24046
SCoPE	0.18181	0.04140	0.39256	0.03063	0.02788	0.45895
ViSPaC (ours)	**0.03576**	**0.01220**	**0.09414**	**0.00902**	**0.01037**	**0.08315**

D Additional Quantitative Results

D.1 Camera Control

Tables 6 and 7 expand the main camera-control comparisons with PSNR, SSIM, and LPIPS, together with camera errors from DA3 (Lin et al., 2025) and ViPE (Huang et al., 2025b). Each table places standard and short clips side by side, following the evaluation protocol in Appendix B.3.

ViSPaC leads all appearance metrics across both datasets and clip protocols. It also leads the camera metrics in both estimator groups, except for DA3 translation RPE on short TNT clips, where Lyra-2 scores lower. The average reductions reported in the abstract use the standard-clip scores aggregated over DA3 and ViPE: we compute each dataset's relative reduction against Lyra-2, then average the two percentages with equal weight, yielding 24.7% for RPE-t and 31.7% for RPE-r.

D.2 Video-conditioned Pose Recovery

As a supplementary assessment, Table 8 evaluates video-conditioned pose recovery on DL3DV (Ling et al., 2024) and TNT (Knapitsch et al., 2017), using the same 30K-step joint model as the camera-control comparisons. We compare ViSPaC with DUSt3R (Wang et al., 2024a), VGGT (Wang et al., 2025a), and DA3 (Lin et al., 2025). We use the joint relative-pose AUC de-

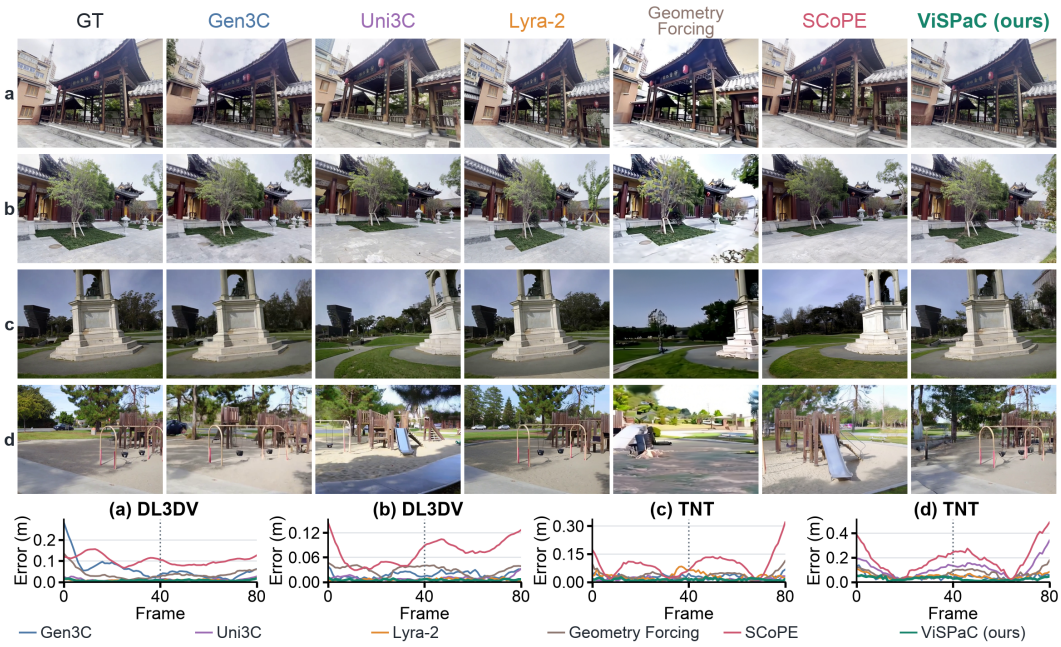

Figure 7: Additional camera-control comparisons with the methods in Figure 5. Rows (a)–(d) show generated views at frame 40 for two DL3DV (Ling et al., 2024) and two TNT (Knapitsch et al., 2017) clips. The plots show full-sequence camera-position errors from DA3 (Lin et al., 2025) trajectories estimated from generated RGB, using the same method colors; dotted lines mark the displayed frame.

Table 8: Video-conditioned pose recovery on DL3DV and TNT. Joint relative-pose AUC is reported in percent; higher is better. Best results are bold and second-best underlined within each column.

	DL3DV		TNT	
Method	AUC@15° ↑	AUC@30° ↑	AUC@15° ↑	AUC@30° ↑
DUSt3R	23.80	40.49	26.41	45.10
VGGT	<u>89.32</u>	<u>94.44</u>	<u>68.67</u>	<u>80.70</u>
DA3	**96.54**	**98.22**	**80.63**	**89.21**
ViSPaC (ours)	87.23	92.52	65.16	76.56

fined in Appendix B.5, evaluating all unordered pairs among the final 80 frames of each clip and averaging over clips. All methods retain the same frame-pair denominator. TNT scores measure agreement with the ViPE-derived reference trajectories.

E Additional Qualitative Comparisons

Figures 7 and 8 extend the camera-control and representation comparisons in Figures 5 and 6, respectively, to additional scenes.

E.1 Extended Camera-Control Comparisons

In Figure 7(c), several baselines shift the monument toward the image boundary, changing how much of it is visible. In row (d), the methods differ in the placement of the playground equipment relative to the foreground tree. ViSPaC more closely matches the target framing in these examples, while its position errors remain low across the displayed DL3DV and TNT sequences. For the trajectory displays in Figure 5, we subtract each path's first camera center to give all paths a common starting origin. Position-error curves retain the evaluation alignment described in Appendix B.3.



Figure 8: Additional representation ablations on Lounge, Gym, and Bookshelves (top to bottom), comparing ViSPaC with Raxel (Jang et al., 2026) and raw pose. DA3 (Lin et al., 2025) camera paths estimated from generated RGB share a projection, scale, and starting origin per scene. Dashed black denotes GT. Markers locate the displayed frame at $t = 48$: an open circle for GT, a filled circle for ViSPaC, a triangle for Raxel, and a square for raw pose.

E.2 Extended Representation Ablations

In Figure 8, the Gym example shows differences in the arrangement of exercise machines and columns; in Bookshelves, they appear in the orientation of the foreground shelf and the extent of its visible side panel. ViSPaC more closely matches these target views, and its estimated camera paths track GT more closely than those of Raxel and raw pose.

F Limitations

Our spatial codebook is designed around priors observed in the training data. Although the diverse training scenes support a broad range of camera trajectories, trajectories with spatial extents or viewing patterns substantially different from these priors may require adapting the codebook. Geometric recovery also assumes known camera intrinsics; extending it to uncalibrated videos would require accounting for unknown intrinsics.