LIFT: LAYOUT-IN-FUTURE VIDEO GENERATION UNDER LARGE VIEWPOINT CHANGE VIA ON-POLICY SELF-DISTILLATION

Anonymous authors
Paper under double-blind review

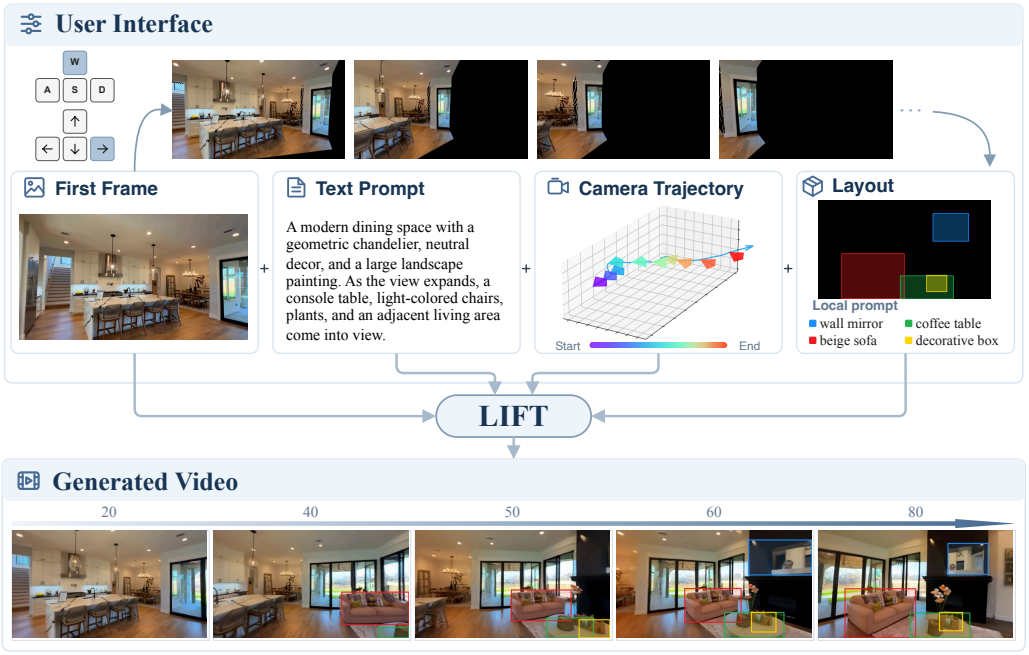


Figure 1: Given a first frame, users can navigate from the first-frame view along a desired camera path and specify layouts using bounding boxes with local text prompts in the final frame. Then, **LIFT** generates the intended shot that transitions from the input image to the user-defined last-frame layout following the prescribed camera trajectory.

ABSTRACT

We introduce **LIFT**, a unified image-to-video generation framework that complements camera control with **L**ayout-**I**n-**F**u**T**ure control, enabling users to specify what should appear in a future view and where it should appear. This addresses a practical need in controllable video generation: given an initial image, users often care not only about how the camera moves, but also about what the scene should look like at key future moments, especially the final frame. Existing camera controls specify viewpoint trajectories, while text prompts provide only coarse semantic guidance; neither precisely determines the content and spatial layout of future views. This limitation becomes particularly pronounced under large viewpoint changes, where the camera reveals regions that are not visible in the first frame. LIFT therefore uses the last-frame layout as an explicit control signal for the desired future scene. Since learning from such sparse layout guidance is substantially more challenging than conditioning on dense per-frame layouts, we introduce on-policy self-distillation (OPSD) to transfer the control capability of a dense-layout teacher to a last-frame-layout student. We further curate **LIFT-Vista**, a dataset featuring large viewpoint changes with camera and temporally consistent layout annotations. Experiments show that LIFT improves video quality, future-layout controllability, and camera controllability over other methods.

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

Recent advances in video generation foundation models (Wan et al., 2025; HaCohen et al., 2026; Seedance et al., 2026) have greatly improved the ability to synthesize high-fidelity, temporally coherent videos from text prompts or a single reference image. Yet precise controllability remains a major barrier to using these models as practical creative tools, especially when the desired camera motion extends far beyond the initial view. As illustrated in fig. 1, a creator may want the camera to move past the dining table and turn toward an unseen living room, while also specifying its composition—for example, a sofa facing the camera, a round coffee table in front of it, and a mirror above the fireplace. Although these elements are not visible in the input image, their content and spatial layout determine what the newly revealed view should look like.

Existing controllable video generation methods address only part of this problem. Camera-controlled video generation (He et al., 2024; Bai et al., 2025a; Team et al., 2026) conditions on a prescribed camera trajectory to determine **how the viewpoint should move**. However, when large camera motion reveals substantial regions outside the reference view, their content remains unspecified: the camera trajectory alone cannot determine **what should appear or where it should be placed**. Layout guidance offers a natural complementary control by explicitly specifying the semantic content and spatial composition of such future views.

While layout-conditioned generation has been extensively studied for images (Zhang et al., 2025c; Huang et al., 2026), it remains far less explored for video. Existing video methods (Li et al., 2025c; Feng et al., 2025) typically rely on dense per-frame boxes, masks, or trajectories, and primarily focus on controlling the motion of objects already visible in the first frame. Moreover, such dense frame-wise guidance places a substantial annotation burden on users.

To address these problems, we introduce **L**ayout-**I**n-**F**u**T**ure (LIFT), a unified video generation framework for large viewpoint changes that supports both camera control and last-frame layout conditioning. LIFT operates in two inference modes: a *single-condition mode*, conditioned only on the camera trajectory, and a *dual-condition mode*, conditioned on both the camera trajectory and the last-frame layout. As shown in fig. 1, LIFT enables users to control not only how the camera moves, but also what should appear in newly revealed regions and where it should appear.

Learning from only a last-frame layout is challenging: without dense per-frame layout guidance, the model must infer how specified objects evolve with camera motion and how the observed scene transitions toward the target composition. We find that direct training with last-frame-only layouts under standard supervised flow matching struggles to exploit the sparse layout condition, leading to inferior future-layout control, while progressively reducing layout density through SFT incurs substantial training cost with limited gains. We therefore first train a dense-layout model and use it as a teacher to supervise the last-frame-layout student on its own rollout states through on-policy self-distillation (OPSD) (Zhao et al., 2026; Jiang et al., 2026; Li et al., 2026b). Experiments demonstrate that OPSD achieves stronger layout control with fewer training sample updates than the SFT baselines. Moreover, camera and layout conditioning are inherently coupled, as dense layouts also capture scene evolution induced by camera motion. To exploit this coupling, we train a shared student across both the single-condition and dual-condition modes while distilling from the same dense-layout teacher. This dual-mode training encourages the two forms of control to reinforce each other, improving both camera and future-layout controllability.

Our contributions are summarized as follows:

- We introduce **LIFT**, a unified video generation framework. It enables users to control both camera motion and the semantic-spatial composition of newly revealed regions using only a last-frame layout.
- We introduce dual-mode OPSD to this task, using dense spatiotemporal layouts as privileged information to train a shared student in both single-condition and dual-condition modes.
- We curate **LIFT-Vista**, a dataset tailored to large viewpoint changes. Our automatic pipeline identifies videos with substantial future-region revelation and produces temporally consistent camera and layout annotations.

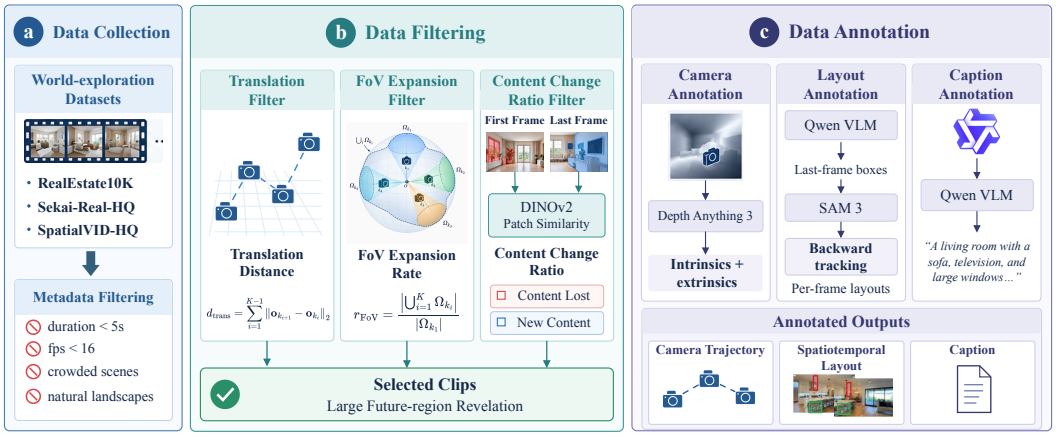


Figure 2: **Data Curation Pipeline.** (a) World-exploration video collection and metadata filtering. (b) Clip selection based on translation distance, FoV expansion, and content change. (c) Annotation of camera trajectories, spatiotemporal layouts, and captions.

2 DATA: LIFT-VISTA

Existing datasets don't directly support our target setting. Camera-annotated video datasets (Zhou et al., 2018; Li et al., 2026e; Wang et al., 2025b) generally lack object-level layout labels, whereas datasets with bounding-box or layout (Li et al., 2025c) typically lack camera trajectories and focus on the first-frame objects. We therefore curate LIFT-VISTA, a dataset specifically for future-view layout control under large viewpoint changes. Our data curation pipeline is illustrated in Fig. 2.

Data Collection. We build LIFT-VISTA from RealEstate10K (Zhou et al., 2018), Sekai (Li et al., 2026e), and SpatialVID (Wang et al., 2025b). The resulting data jointly provides camera trajectories and spatiotemporal object layouts, with an emphasis on scenes in which camera motion reveals regions outside the initial view.

Data Filtering. We first remove clips with undesirable scene properties, such as crowded scenes and natural landscapes. We then retain clips with substantial future-region revelation using three complementary metrics: FoV expansion ratio, accumulated translation, and content change ratio.

We uniformly sample K keyframes from each clip. For the i-th keyframe, let $\Omega_{k_i} \subseteq \mathbb{S}^2$ denote the set of visible viewing directions in a common world coordinate system. We estimate its spherical area using uniformly sampled directions on the unit sphere. The FoV expansion ratio is defined as

$$r_{\text{FoV}} = \frac{\left|\bigcup_{i=1}^{K} \Omega_{k_i}\right|}{|\Omega_{k_1}|}, \tag{1}$$

which measures the total viewing region covered by the clip relative to the first frame, and the accumulated camera translation as

$$d_{\text{trans}} = \sum_{i=1}^{K-1} \left\|\mathbf{o}_{k_{i+1}} - \mathbf{o}_{k_i}\right\|_2, \tag{2}$$

where $\mathbf{o}_{k_i}$ denotes the camera origin of the i-th keyframe. Since camera motion alone does not directly measure changes in visible scene content, we additionally compute a patch-level CCR between the first and last frames using DINOv2 (Oquab et al., 2023). Let $\{\mathbf{p}_i\}_{i=1}^{N}$ and $\{\mathbf{q}_j\}_{j=1}^{N}$ denote their ℓ_2-normalized patch embeddings. The last-frame CCR is

$$r_{lf\text{-}CCR} = \frac{1}{N} \sum_{j=1}^{N} \mathbb{I}\left[\max_i \langle \mathbf{p}_i, \mathbf{q}_j \rangle < \tau\right], \tag{3}$$

where $\mathbb{I}[\cdot]$ denotes the indicator function, $\langle \cdot, \cdot \rangle$ denotes the cosine similarity, and τ is a similarity threshold. $r_{lf\text{-}CCR}$ measures the fraction of last-frame patches unmatched in the first frame. We analogously compute $r_{ff\text{-}CCR}$ in the reverse direction to measure content leaving the initial view.

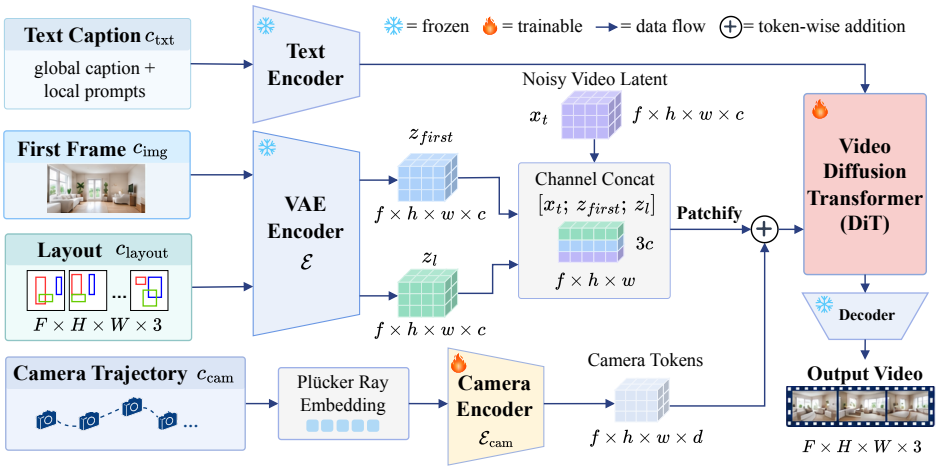


Figure 3: **Model Architecture.** Layout latent is channel-concatenated with the noisy video latent and first-frame latent. Camera tokens are injected into the DiT stream through token-wise addition. Color-referenced layout local prompts are appended to the global caption.

Data Annotation. For camera trajectories, we apply Depth Anything 3 (Lin et al., 2025) to annotate camera intrinsics and extrinsics across all datasets, unifying coordinate systems. For spatiotemporal layout annotation, we first detect and annotate object-level bounding boxes in the last frame of each clip. Then, we use SAM3 (Carion et al., 2025) to track these objects throughout the entire clip, producing dense per-frame layouts.

3 Method: LIFT

3.1 Preliminary

Diffusion On-Policy Self-Distillation. OPSD uses the same model to act as both student and teacher. The student is conditioned only on the inference-time context c, whereas the teacher additionally observes privileged information r. In the LLM domain, the student is trained to match the teacher distribution using reverse KL. Recent works (Fang et al., 2026; Li et al., 2026d; Zhou et al., 2026) study on-policy distillation for diffusion models. In our ODE-based rollout setting, we use the following velocity-matching surrogate objective:

$$\mathcal{L}^{\text{FM}}_{\text{OPSD}}(\theta) = \mathbb{E}_{x_{t_0:t_N}\sim p_\theta(\cdot|c)} \left[\sum_{j=0}^{N-1} w(t_j) \left\| v_\theta(x_{t_j}, t_j, c) - \text{sg}\left[v_{\theta_{old}}(x_{t_j}, t_j, c, r) \right] \right\|_2^2 \right]. \quad (4)$$

where θ_{old} denotes the frozen teacher parameters, $w(t_j)$ is an optional timestep-dependent weighting function, and $\text{sg}[\cdot]$ denotes the stop-gradient operation.

3.2 Joint Camera and Layout Conditioned DiT

We build an image-to-video diffusion model jointly conditioned on four signals: a reference first frame c_{img} that specifies the initial scene appearance, a text caption c_{txt} describing the video content, a target camera trajectory c_{cam} specifying the viewpoint change, and a spatiotemporal layout c_{layout} specifying the locations and semantics of objects at the conditioned frames. An overview of the architecture is shown in fig. 3.

Layout Control. We introduce layout maps to explicitly control layouts throughout the generated video. Using the layout annotations described in Sec. 2, we render the bounding boxes into a pixel-aligned layout map video m_l, where each object instance is assigned a unique color that remains consistent across frames to preserve its identity. The layout map is encoded by the shared VAE encoder $\mathcal{E}$: $z_l = \mathcal{E}(m_l)$. We then concatenate the noisy video latent x_t, the first-frame latent z_{first}, and the layout latent z_l along the channel dimension:

$$\tilde{x}_t = \text{Concat}_{\text{ch}}\left(x_t, z_{\text{first}}, z_l\right), \quad (5)$$

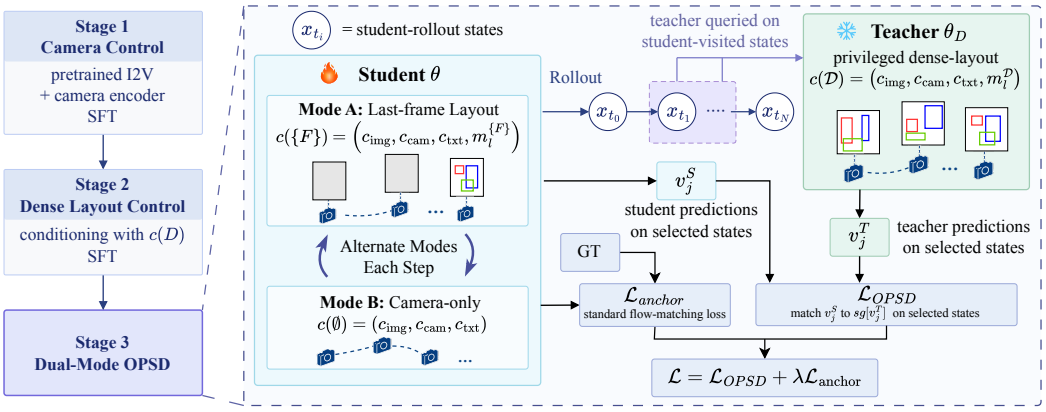


Figure 4: **Dual-mode OPSD.** The student alternates between last-frame-layout and camera-only modes and distills selected rollout states from a frozen dense-layout teacher with a flow-matching anchor loss. The student and teacher are both initialized from $\theta_{\mathcal{D}}$.

where $\tilde{x}_t$ is subsequently projected into visual tokens by the patchification layer. The layout map specifies *where* objects should appear, while their semantic information is provided through text. Specifically, we associate each object description with its corresponding bbox color and append these local object prompts to the global video caption. Together, the layout map and color-referenced local prompts provide geometric and semantic control.

Camera Control. We adopt Plücker ray embeddings $\mathcal{P} \in \mathbb{R}^{F\times H\times W\times 6}$ (He et al., 2024; Bahmani et al., 2025a) as the camera representation, which provide strong per-pixel geometric information. A lightweight camera encoder $\mathcal{E}_{\text{cam}}$ transforms the Plücker representation into camera latent tokens that are spatiotemporally aligned with the patchified video tokens (He et al., 2025; Wan et al., 2025). These camera tokens are then injected into the DiT stream through token-wise addition:

$$\mathcal{H}_{in} = \text{patchify}(\tilde{x}_t) + \mathcal{E}_{\text{cam}}(\mathcal{P}), \tag{6}$$

where $\mathcal{H}_{\text{in}}$ is fed into the diffusion Transformer. This spatiotemporally aligned camera conditioning allows the denoising network to directly associate video contents with the prescribed camera motion.

3.3 Dual-Mode OPSD Training

Our model needs to integrate two controls —camera trajectory and future-view layout. In particular, we find that directly learning last-frame-only layout conditioning with standard SFT is highly challenging. We therefore adopt OPSD to reach the final last-frame-layout regime, which is much more data-efficient and effective.

Conditioning Modes. Let $\mathcal{S} \subseteq \{1, \ldots, F\}$ denote the set of frames at which the layout is exposed to the model. The layout map $m_l^{\mathcal{S}}$ renders object boxes only at frames in $\mathcal{S}$ and leaves other frames empty. The corresponding conditioning context is

$$c(\mathcal{S}) = \left(c_{\text{img}}, c_{\text{cam}}, c_{\text{txt}}, m_l^{\mathcal{S}}\right) \tag{7}$$

We define $\mathcal{S} = \mathcal{D} \triangleq \{1, \ldots, F\}$ as the *dense-layout* mode, $\mathcal{S} = \{F\}$ as the *lastframe-layout* mode (i.e. dual-condition mode), and $\mathcal{S} = \emptyset$ as the *camera-only* mode (i.e. single-condition mode).

Our training has 3 stages: camera control, dense layout control, and dual-mode OPSD. For stage 1, we train the camera controllability, adapting the model to our task setting (i.e. large viewpoint changes and future-region revelation) section 2. For stage 2, we introduce the layout conditioning and continue SFT under the dense layout context $c(\mathcal{D})$. The resulting weights, denoted as $\theta_{\mathcal{D}}$, serve both as the teacher and as the student initialization for stage 3.

Dual-Mode OPSD. Camera and layout control are not fully independent. A dense spatiotemporal layout implicitly describes how the scene evolves under viewpoint changes and can therefore convey part of the camera-induced motion (Li et al., 2025c; Wang et al., 2025c). As illsustrated in fig. 4, we perform OPSD in two student conditioning modes: $\mathcal{S} \in \{\{F\}, \emptyset\}$, to jointly improve lastframe-layout control and camera-only control. Both student modes share the same parameters and are distilled from the same dense-layout teacher.

OPSD Objective. We freeze the Stage 2 model $\theta_{\mathcal{D}}$ as the teacher and initialize the student θ from the same parameters. For each training sample, we first perform on-policy rollout with the student under $c(\mathcal{S})$, without gradient tracking. The frozen dense-layout teacher is then queried at selected states along the student trajectory:

$$\mathcal{L}_{\mathrm{OPSD}}(\theta;\mathcal{S}) = \mathbb{E}_{x_{t_0:t_N}\sim p_\theta(\cdot|c(\mathcal{S}))}\left[\frac{1}{|\mathcal{K}_{\mathcal{S}}|}\sum_{j\in\mathcal{K}_{\mathcal{S}}} w(t_j)\left\|v_j^S - \mathrm{sg}[v_j^T]\right\|_2^2\right], \tag{8}$$

where $x_{t_0:t_N}$ denotes the student rollout trajectory, $v_j^S = v_\theta(x_{t_j}, t_j, c(\mathcal{S}))$ and $v_j^T = v_{\theta_{\mathcal{D}}}(x_{t_j}, t_j, c(\mathcal{D}))$ denote the student and teacher velocity predictions, respectively, $\mathcal{K}_{\mathcal{S}} \subseteq \{0,\dots,N-1\}$ denotes the subset of student-visited states queried for distillation, and sg denotes stop-gradient.

Anchoring Loss. Although dense layout provides the teacher with better spatiotemporal control, the teacher itself is imperfect. Optimizing the OPSD objective eq. (8) alone can degrade generation quality. Therefore, we maintain the standard flow-matching objective as an anchoring loss:

$$\mathcal{L}_{\mathrm{anchor}}(\theta;\mathcal{S}) = \mathbb{E}_{x_0,\epsilon,t}\left[\left\|v_\theta\big(x_t^{\mathrm{FM}}, t, c(\mathcal{S})\big) - v_t^\star(x_0,\epsilon)\right\|_2^2\right]. \tag{9}$$

where $v_t^\star(x_0,\epsilon)$ denotes the flow matching velocity target. This term helps maintain generation fidelity while OPSD transfers dense-layout knowledge to sparse conditioning modes.

Full Objective. The overall Stage 3 objective is

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathcal{S}\sim\pi}\Big[\mathcal{L}_{\mathrm{OPSD}}(\theta;\mathcal{S}) + \lambda\,\mathcal{L}_{\mathrm{anchor}}(\theta;\mathcal{S})\Big], \tag{10}$$

where π is the sampling distribution over the two target conditioning modes $\mathcal{S}\in\{\{F\},\emptyset\}$, and λ is the anchor weight, where we set the anchor weight to $\lambda = 0.1$ for Stage 3 training.

Selective State Distillation. Not all states along the student rollout provide equally useful distillation signals. The global spatial layout configuration is largely determined during the early, high-noise stage of the denoising trajectory (Hertz et al., 2023). At these states, we observe that the teacher with privileged dense-layout conditioning can correct the student to the desired layout. In contrast, at later low-noise states, the teacher produces nearly no corrections, making the corresponding distillation signal less informative. We therefore concentrate OPSD supervision on the first 10 high-noise states of each student rollout.

4 Experiment

4.1 Implementation Details

We build our model on top of Wan2.1-Fun-V1.1-1.3B-Control-Camera (Wan et al., 2025). All experiments are conducted at a resolution of 352×640, using 81-frame clips at 16 FPS. Training is performed on 4 NVIDIA H100 GPUs. For the three training stages, we optimize the model for 8,000, 4,000, and 500 steps, respectively. The corresponding global batch sizes are 32, 32, and 16, with learning rates of 1×10^{-5}, 1×10^{-4}, and 5×10^{-5}. We use AdamW as the optimizer. For Stage 3, the sampling probabilities for the two OPSD modes are $P(\mathcal{S}=\{F\}) = 0.7$ and $P(\mathcal{S}=\emptyset) = 0.3$ for the lastframe-layout and camera-only modes, respectively. For inference, we use 50 denoising steps and a cfg scale of 6.0. More implementation details are included in section A.2.1 and section A.1.

4.2 Quantitative and Qualitative Comparisons

Baselines. We compare against three categories of controllable video generation methods: camera control, object motion control, and joint camera-and-object motion control. For camera control, we evaluate against two recent state-of-the-art methods, Uni3C (Cao et al., 2025) and GEN3C (Ren et al., 2025). For object motion control, we compare with MagicMotion (Li et al., 2025c), which uses bounding-box trajectories to specify object motion. We further include Direct-a-Video (Yang et al., 2024) as a joint-control baseline. Direct-a-Video supports training-free control of object motion using bounding-box trajectories, whereas its camera control is restricted to horizontal/vertical panning and zooming. For a favorable comparison to baselines, we provide MagicMotion and Direct-a-Video with dense per-frame layout trajectories, whereas our model uses only a last-frame layout.

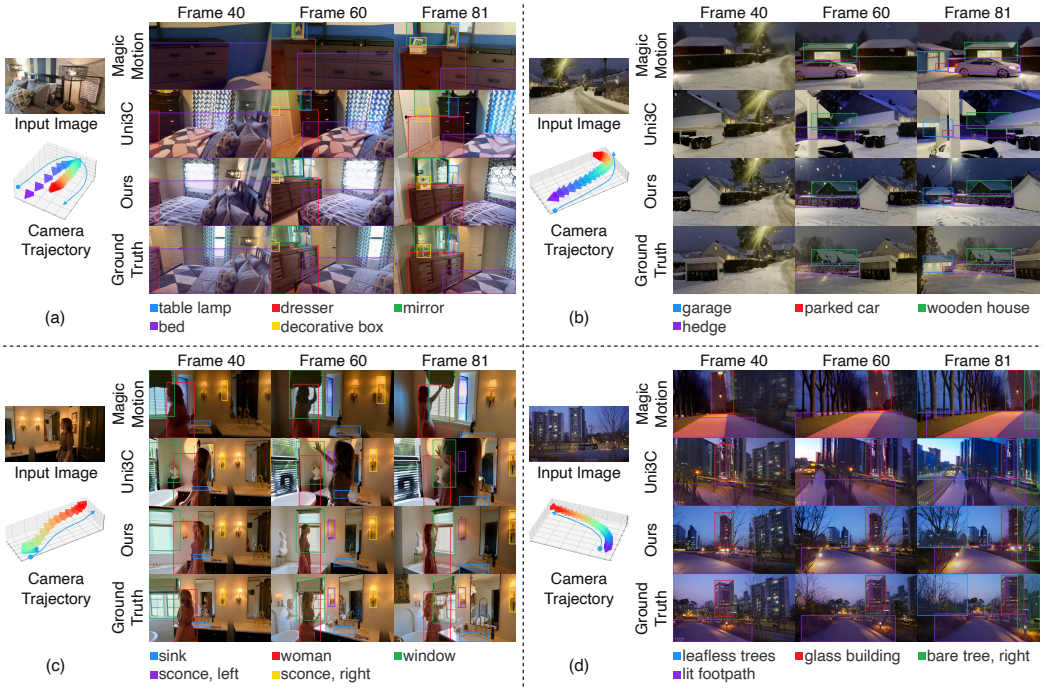


Figure 5: **Qualitative comparison.** Bounding boxes indicate the target locations of conditioned objects. MagicMotion lacks explicit camera control and often produces inconsistent scene evolution or incorrect objects, while Uni3C follows the prescribed camera trajectory but leaves newly revealed regions uncontrolled, resulting in unspecified contents, e.g., a pillar in (b) and a car street in (d). In contrast, ours follows the prescribed camera motion and realizes the specified future-view layout.

Table 1: **Quantitative comparison** with state-of-the-art controllable video generation methods. We report video quality, camera trajectory accuracy, and semantic consistency metrics. The best, second-best, and third-best results are highlighted accordingly.

Method	Control			Video Quality			Camera Error		Semantic Consistency		
	Camera	Object Motion	Future Layout	FVD ↓	FID ↓	LPIPS ↓	RotErr ↓	TransErr ↓	mIoU ↑	SR_e ↑	$\text{CLIP}_{\text{local}}$ ↑
Direct-a-Video	✓	✓	✗	539.28	71.56	0.80	28.48	2.28	0.10	0.11	0.14
MagicMotion	✗	✓	✗	277.94	18.32	0.57	15.74	1.59	0.41	0.53	0.21
GEN3C	✓	✗	✗	89.59	15.60	0.51	3.91	2.62	0.18	0.38	0.19
Uni3C	✓	✗	✗	111.60	12.75	0.42	3.23	0.70	0.31	0.47	0.21
LIFT	✓	✓	✓	99.35	12.84	0.42	2.97	0.59	0.51	0.59	0.24

Metrics. We evaluate generated videos in terms of visual quality, camera controllability, and layout controllability. For visual quality, we report FVD (Unterthiner et al., 2018), FID (Heusel et al., 2017), and LPIPS (Zhang et al., 2018). For camera control, we measure rotation error (RotErr) and translation error (TransErr) between the generated and target camera trajectories ((Zhang et al., 2025b)). For layout control, following OverLayBench (Li et al., 2026a), we report mIoU, entity success rate (SR_e), and $\text{CLIP}_{\text{local}}$ (Radford et al., 2021) to evaluate spatial alignment, entity-level success, and local semantic consistency, respectively.

Quantitative and Qualitative Results. As shown in Tab. 1, our method achieves strong performance across all three evaluation dimensions. LIFT (1.3B) achieves competitive video quality against 14B Uni3C and 7B GEN3C. LIFT also achieves the best camera-control accuracy among the compared methods while supporting future-layout control. More importantly, LIFT consistently achieves the best layout-control performance. Compared with MagicMotion, which is additionally provided with dense per-frame bounding-box trajectories, LIFT improves mIoU from 0.41 to 0.51, despite requiring only a last-frame layout. These results demonstrate that LIFT effectively combines camera control with future-view spatial control while maintaining competitive generation quality. We show more visualization qualitative results in fig. 5 and supplementary materials.

Table 2: **Ablation study of SFT and OPSD.** D2S-SFT denotes dense-to-sparse curriculum SFT. Training cost is measured by the number of training sample updates (steps $\times$ global batch size). OPSD achieves the best results with much fewer training samples of the SFT-based alternatives.

Method	Training Sample (Steps $\times$ BS)	Video Quality			Camera Error		Semantic Consistency		
		FVD ↓	FID ↓	LPIPS ↓	RotErr ↓	TransErr ↓	mIoU ↑	SR_e ↑	$\text{CLIP}_{\text{local}}$ ↑
Direct last-frame SFT	$4\text{K} \times 32 = 128\text{K}$	114.03	12.96	0.43	2.98	0.56	0.44	0.56	0.23
D2S-SFT	$4\text{K} \times 32 = 128\text{K}$	110.53	12.99	0.42	3.02	0.51	0.47	0.59	0.23
Ours	$500 \times 16 = 8\text{K}$	99.35	12.84	0.42	2.97	0.59	0.51	0.59	0.24
Ours w/o SFT anchor	–	129.14	17.16	0.45	3.36	0.68	0.49	0.57	0.22

Table 3: **Ablation study of dual-mode OPSD.** The dense-layout teacher and the student before OPSD training are included as references.

OPSD Variant	Lastframe-Layout Mode							Camera-Only Mode			
	FVD ↓	FID ↓	RotErr ↓	TransErr ↓	mIoU ↑	SR_e ↑	$\text{CLIP}_{\text{local}}$ ↑	FVD ↓	FID ↓	RotErr ↓	TransErr ↓
Teacher (Dense layout)	123.50	13.72	2.92	0.51	0.60	0.60	0.23	147.20	14.25	3.85	0.63
Student @ step0	137.10	14.08	3.58	0.58	0.40	0.56	0.22	147.20	14.25	3.85	0.63
Lastframe-layout single-mode	102.07	13.07	2.88	0.58	0.49	0.59	0.23	121.88	13.42	3.09	0.67
Camera-only single-mode	108.30	14.00	3.04	0.59	0.38	0.57	0.22	105.22	13.77	3.03	0.63
Ours (dual-mode)	99.35	12.84	2.97	0.59	0.51	0.59	0.24	95.66	13.18	3.30	0.63

Table 4: **Ablation study of mode sampling probability in dual-mode OPSD.** p_{layout} denotes the probability of sampling the lastframe-layout mode during training.

$\mathbf{p}_{\text{layout}}$	Lastframe-Layout Mode							Camera-Only Mode			
	FVD ↓	FID ↓	RotErr ↓	TransErr ↓	mIoU ↑	SR_e ↑	$\text{CLIP}_{\text{local}}$ ↑	FVD ↓	FID ↓	RotErr ↓	TransErr ↓
0.5	102.63	13.40	2.936	0.620	0.4898	0.5862	0.2306	106.01	13.85	3.190	0.703
0.7	99.35	12.84	2.968	0.593	0.5094	0.5861	0.2351	95.66	13.18	3.304	0.632
0.9	93.05	12.96	2.978	0.653	0.5060	0.5832	0.2321	98.62	13.34	3.357	0.757

4.3 Ablation Study

SFT vs. OPSD. We compare OPSD with direct last-frame SFT and dense-to-sparse curriculum SFT (D2S-SFT) in table 2 and fig. 7. Directly optimizing the last-frame layout condition with SFT yields limited controllability. Introducing a dense-to-sparse curriculum improves mIoU from 0.44 to 0.47, suggesting that this curriculum facilitates adaptation to sparse layout conditioning. However, SFT still requires substantial optimization to adapt to the last-frame-only condition. In contrast, our OPSD surpasses the SFT baselines in 5 out of 8 metrics using only 8K training sample updates, compared with 128K for the SFT baselines. For simplicity, the training cost reported in table 2 excludes the first 4K SFT steps for all three methods and measures only the subsequent adaptation cost. A full cost comparison, including the dense-layout SFT preceding OPSD, is provided in section A.2.1. This demonstrates that OPSD provides a substantially more data-efficient and effective way to achieve the last-frame layout control.

SFT Anchor Loss in OPSD. As shown in table 2, OPSD alone without the SFT anchor loss can effectively transfer privileged layout information, but may drift away from the original data distribution and degrade performance.

Dual-Mode OPSD vs. Single-Mode OPSD. Our future-layout control is built upon camera control, and the two control modalities are therefore not fully independent. As shown in table 3, lastframe-layout single-mode OPSD also improves camera-only inference over the step-0 student. Camera-only single-mode OPSD improves camera-only performance compared to lastframe-layout single-mode OPSD, but degrades layout controllability under the lastframe-layout inference setting. In contrast, our dual-mode OPSD jointly distills both modes from the same dense-layout teacher and achieves the best overall performance across the two inference settings. It delivers the best video quality and layout controllability while maintaining comparable camera accuracy, suggesting that dual-mode training promotes beneficial interaction between camera and layout conditioning.

Mode Sampling Probability. We further study the sampling probability between the lastframe-layout and camera-only modes in table 4. $p_{\text{layout}} = 0.7$ achieves the best overall results, so we use it as our default setting and in other experiments.

More ablation studies are included in section A.2.3.

5 Related Work

5.1 Controllable Video Generation

Camera Control. Camera-controllable video generation (Bahmani et al., 2025b; Bai et al., 2025a;b; Zheng et al., 2024; Li et al., 2025d; Yu et al., 2025) aims to explicitly control viewpoint trajectory during synthesis. Recent works use camera extrinsics (Wang et al., 2024b; Bai et al., 2025a), Plücker-ray embeddings (He et al., 2024; Bahmani et al., 2025a; He et al., 2025), or explicit 3D priors (Ren et al., 2025; Cao et al., 2025; Wang et al., 2025d) as camera representations and inject them into video diffusion models. Some approaches also incorporate relative camera geometry into attention through positional encodings (Zhang et al., 2026; Li et al., 2026c). More recently, world models extend camera control towards long-horizon scene exploration (Li et al., 2025a; Sun et al., 2026; Mao et al., 2025; Team et al., 2026). However, these methods only determine how the viewpoint should move. Our work complements camera control with object-level layout guidance, enabling explicit control of the semantic and spatial composition of future views beyond the initially observed regions.

Object Motion and Layout Control. Spatiotemporal control has been extensively studied for manipulating object motion. Video generation models use point trajectories (Geng et al., 2025; Wang et al., 2025a; Gu et al., 2025), bounding boxes (Jain et al., 2024; Wu et al., 2024a; Wang et al., 2025c; Li et al., 2025c), or masks (Wu et al., 2024b; Yariv et al., 2025) to specify object motion over time. Some works can also jointly control object motion and camera (Yang et al., 2024; Wu et al., 2024a; Chen et al., 2025; Xing et al., 2025; Zheng et al., 2026). These methods specify how an object moves, but typically assume that the controlled object is already present in the first frame. Relatedly, layout-conditioned generation specifies what objects should appear and where. While it has been widely explored in image generation (Wang et al., 2024a; Zhang et al., 2025a;c; Huang et al., 2026), it remains less explored in video generation. Existing methods rely on dense per-frame object descriptions (Li et al., 2025b; Feng et al., 2025), mainly targeting object motion. In contrast, LIFT targets future-view layout control under large viewpoint changes, supporting last-frame-only layout conditioning.

5.2 On-Policy Self-Distillation

On-policy distillation (OPD) (Agarwal et al., 2024) has recently emerged as an effective alternative post-training method complementary to SFT and RL. OPD trains a student to match a teacher on trajectories generated by the student itself, thereby reducing the train–inference distribution mismatch of SFT. Compared with reinforcement learning with verifiable rewards (RLVR) such as GRPO (Shao et al., 2024), which typically relies on a scalar sequence-level reward, OPD offers dense token-level supervision from the teacher. Recent works (Fang et al., 2026; Li et al., 2026d; Zhou et al., 2026; Xu et al., 2026; Fu et al., 2026) extend this method to diffusion and flow-matching models. On-policy self-distillation (OPSD) (Zhao et al., 2026; Jiang et al., 2026) further uses the model itself as the teacher by providing it with richer context, known as privileged information, while the student receives only inference-time conditions. It removes the need for a separately trained or larger teacher. While OPSD has recently received increasing attention in LLMs, it remains relatively underexplored in image and video generation (Li et al., 2026b; Liu et al., 2026). LIFT uses OPSD to achieve last-frame layout control and encourage the synergy between camera and layout conditioning.

6 Conclusion

In this paper, we introduce LIFT, a unified framework for future-view layout control under large viewpoint changes. LIFT enables users to control both camera motion and the semantic and spatial composition of newly revealed regions through two inference modes: a single mode conditioned on the camera trajectory and a dual mode additionally conditioned on the last-frame layout. To support this setting, we curate LIFT-Vista, a dataset featuring substantial future-region revelation with temporally consistent camera and layout annotations. To effectively learn from sparse last-frame layout guidance, we adopt on-policy self-distillation (OPSD) and apply it in a dual-mode training scheme, transferring privileged dense-layout knowledge to a shared student across both inference modes. Together, these designs improve camera and future-layout controllability within a unified video generation framework.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation, polish writing, visualize results, and draw figures. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our user study in section A.2.4 asks participants to evaluate generated videos based on control adherence and visual quality. All models and datasets we use are publicly available.

REPRODUCIBILITY STATEMENT

We will release the code and dataset upon acceptance. We describe the details of our model architecture, training procedure and datasets in section 2, section 3, and section A.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/5be69a584901a26c521c2b51e40a4c20-Paper-Conference.pdf.

Sherwin Bahmani, Ivan Skorokhodov, Guocheng Qian, Aliaksandr Siarohin, Willi Menapace, Andrea Tagliasacchi, David B Lindell, and Sergey Tulyakov. Ac3d: Analyzing and improving 3d camera control in video diffusion transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22875–22889, 2025a.

Sherwin Bahmani, Ivan Skorokhodov, Aliaksandr Siarohin, Willi Menapace, Guocheng Qian, Michael Vasilkovsky, Hsin-Ying Lee, Chaoyang Wang, Jiaxu Zou, Andrea Tagliasacchi, David B. Lindell, and Sergey Tulyakov. Vd3d: Taming large video diffusion transformers for 3d camera control. In *International Conference on Learning Representations (ICLR)*, 2025b.

Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, and Di Zhang. Recammaster: Camera-controlled generative rendering from a single video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025a.

Jianhong Bai, Menghan Xia, Xintao Wang, Ziyang Yuan, Xiao Fu, Zuozhu Liu, Haoji Hu, Pengfei Wan, and Di Zhang. Syncammaster: Synchronizing multi-camera video generation from diverse viewpoints. In *International Conference on Learning Representations (ICLR)*, 2025b.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025c. URL https://arxiv.org/abs/2511.21631.

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 *ACM SIGGRAPH Asia 2025 Conference Papers*, 2025.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

Yingjie Chen, Yifang Men, Yuan Yao, Miaomiao Cui, and Liefeng Bo. Perception-as-control: Fine-grained controllable image animation with 3d-aware motion representation. *arXiv preprint arXiv:2501.05020*, 2025.

Zhen Fang, Wenxuan Huang, Yu Zeng, Yiming Zhao, Shuang Chen, Kaituo Feng, Yunlong Lin, Lin Chen, Zehui Chen, Shaosheng Cao, and Feng Zhao. Flow-opd: On-policy distillation for flow matching models, 2026. URL https://arxiv.org/abs/2605.08063.

Weixi Feng, Chao Liu, Sifei Liu, William Yang Wang, Arash Vahdat, and Weili Nie. Blobgen-vid: Compositional text-to-video generation with blob video representations. *arXiv preprint*, 2025.

Siming Fu, Zheming Fu, Ruizhe He, Hualiang Wang, Jie Huang, Xiaoxiao Ma, Mingchen Zhong, Weihu Huang, Xiaoxuan He, and Haojun Xu. Any-opd: Heterogeneous on-policy distillation for flow-matching models via representation-space bridging, 2026. URL https://arxiv.org/abs/2608.03316.

Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Tatiana Lopez-Guevara, Carl Doersch, Yusuf Aytar, Michael Rubinstein, Chen Sun, Oliver Wang, Andrew Owens, and Deqing Sun. Motion prompting: Controlling video generation with motion trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Zekai Gu, Rui Yan, Jiahao Lu, Peng Li, Zhiyang Dou, Chenyang Si, Zhen Dong, Qifeng Liu, Cheng Lin, Ziwei Liu, Wenping Wang, and Yuan Liu. Diffusion as shader: 3d-aware video diffusion for versatile video generation control. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, SIGGRAPH Conference Papers '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400715402. doi: 10.1145/3721238.3730607. URL https://doi.org/10.1145/3721238.3730607.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, et al. Ltx-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026.

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

Hao He, Ceyuan Yang, Shanchuan Lin, Yinghao Xu, Meng Wei, Liangke Gui, Qi Zhao, Gordon Wetzstein, Lu Jiang, and Hongsheng Li. Cameractrl ii: Dynamic scene exploration via camera-controlled video diffusion models. *arXiv preprint arXiv:2503.10592*, 2025.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. In *International Conference on Learning Representations*, 2023.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Sida Huang, Siqi Huang, Ping Luo, and Hongyuan Zhang. Laytrol: Preserving pretrained knowledge in layout control for multimodal diffusion transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Yash Jain, Anshul Nasery, Vibhav Vineet, and Harkirat Behl. Peekaboo: Interactive video generation via masked-diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Dengyang Jiang, Xin Jin, Dongyang Liu, Zanyi Wang, Mingzhe Zheng, Ruoyi Du, Xiangpeng Yang, Qilong Wu, Zhen Li, Peng Gao, Harry Yang, and Steven Hoi. D-opsd: On-policy self-distillation for continuously tuning step-distilled diffusion models. *arXiv preprint arXiv:2605.05204*, 2026.

Bingnan Li, Chen-Yu Wang, Haiyang Xu, Xiang Zhang, Ethan Armand, Divyansh Srivastava, Shan Xiaojun, Zeyuan Chen, Jianwen Xie, and Zhuowen Tu. Overlaybench: A benchmark for layout-to-image generation with dense overlaps. *Advances in Neural Information Processing Systems*, 38, 2026a.

Bingnan Li, Haozhe Wang, Haozhong Xiong, Fangtai Wu, Jinpeng Yu, Yang Shi, Jiaming Liu, and Ruihua Huang. Rethinking classifier-free guidance in on-policy diffusion distillation, 2026b. URL https://arxiv.org/abs/2607.24731.

Chunyang Li, Yuanbo Yang, Jiahao Shao, Hongyu Zhou, Katja Schwarz, and Yiyi Liao. Rerope: Repurposing rope for relative camera control, 2026c. URL https://arxiv.org/abs/2602.08068.

Jiaqi Li, Junshu Tang, Zhiyong Xu, Longhuang Wu, Yuan Zhou, Shuai Shao, Tianbao Yu, Zhiguo Cao, and Qinglin Lu. Hunyuan-gamecraft: High-dynamic interactive game video generation with hybrid history condition, 2025a.

Pengxiang Li, Kai Chen, Zhili Liu, Ruiyuan Gao, Lanqing Hong, Dit-Yan Yeung, Huchuan Lu, and Xu Jia. Trackdiffusion: Tracklet-conditioned video generation via diffusion models. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3539–3548. IEEE, 2025b.

Quanhao Li, Zhen Xing, Rui Wang, Hui Zhang, Qi Dai, and Zuxuan Wu. Magicmotion: Controllable video generation with dense-to-sparse trajectory guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025c.

Quanhao Li, Junqiu Yu, Kaixun Jiang, Yujie Wei, Zhen Xing, Pandeng Li, Ruihang Chu, Shiwei Zhang, Yu Liu, and Zuxuan Wu. Diffusionopd: A unified perspective of on-policy distillation in diffusion models. *SIGGRAPH Asia 2026 Conference Papers*, 2026d.

Teng Li, Guangcong Zheng, Rui Jiang, Shuigen Zhan, Tao Wu, Yehao Lu, Yining Lin, and Xi Li. Realcam-i2v: Real-world image-to-video generation with interactive complex camera control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025d.

Zhen Li, Chuanhao Li, Xiaofeng Mao, Shaoheng Lin, Ming Li, Shitian Zhao, Zhaopan Xu, Xinyue Li, Yukang Feng, Jianwen Sun, et al. Sekai: A video dataset towards world exploration. *Advances in Neural Information Processing Systems*, 2026e.

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.

Hongyu Liu, Chun Wang, Feng Gao, Xuanhua He, Yue Ma, Ziyu Wan, Yong Zhang, Xiaoming Wei, and Qifeng Chen. Opsd-v: On-policy self-distillation for post-training few-step autoregressive video generators. *arXiv preprint arXiv:2607.08766*, 2026.

Xiaofeng Mao, Zhen Li, Chuanhao Li, Xiaojie Xu, Kaining Ying, Tong He, Jiangmiao Pang, Yu Qiao, and Kaipeng Zhang. Yume-1.5: A text-controlled interactive world generation model. *arXiv preprint arXiv:2512.22096*, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. Worldplay: Towards long-term geometric consistency for real-time interactive world modeling, 2026. URL https://arxiv.org/abs/2512.14614.

Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, Yihang Chen, Jie Liu, Yansong Cheng, Yao Yao, Jiayi Zhu, Yihao Meng, Kecheng Zheng, Qingyan Bai, Jingye Chen, Zehong Shen, Yue Yu, Xing Zhu, Yujun Shen, and Hao Ouyang. Advancing open-source world models, 2026. URL https://arxiv.org/abs/2601.20540.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

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.

Hanlin Wang, Hao Ouyang, Qiuyu Wang, Wen Wang, Ka Leong Cheng, Qifeng Chen, Yujun Shen, and Limin Wang. Levitor: 3d trajectory oriented image-to-video synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a.

Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Yi Zhang, Chang Zeng, Yanxi Zhou, et al. Spatialvid: A large-scale video dataset with spatial annotations. *arXiv preprint arXiv:2509.09676*, 2025b.

Qinghe Wang, Yawen Luo, Xiaoyu Shi, Xu Jia, Huchuan Lu, Tianfan Xue, Xintao Wang, Pengfei Wan, Di Zhang, and Kun Gai. Cinemaster: A 3d-aware and controllable framework for cinematic text-to-video generation. In *ACM SIGGRAPH 2025 Conference Papers*, 2025c.

Xudong Wang, Trevor Darrell, Sai Saketh Rambhatla, Rohit Girdhar, and Ishan Misra. Instancediffusion: Instance-level control for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024a.

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.

Zun Wang, Jaemin Cho, Jialu Li, Han Lin, Jaehong Yoon, Yue Zhang, and Mohit Bansal. Epic: Efficient video camera control learning with precise anchor-video guidance. *arXiv preprint arXiv:2505.21876*, 2025d.

Jianzong Wu, Xiangtai Li, Yanhong Zeng, Jiangning Zhang, Qianyu Zhou, Yining Li, Yunhai Tong, and Kai Chen. Motionbooth: Motion-aware customized text-to-video generation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a.

Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. Draganything: Motion control for anything using entity representation. In *European Conference on Computer Vision (ECCV)*, 2024b.

Jinbo Xing, Long Mai, Cusuh Ham, Jiahui Huang, Aniruddha Mahapatra, Chi-Wing Fu, Tien-Tsin Wong, and Feng Liu. Motioncanvas: Cinematic shot design with controllable image-to-video generation. In *ACM SIGGRAPH 2025 Conference Papers*, 2025.

Yixian Xu, Kaiyuan Gao, Yuxiang Chen, Yilei Chen, Zecheng Tang, Zihao Liu, Zikai Zhou, Deqing Li, Hao Meng, Kuan Cao, Jiahao Li, Jie Zhang, Liang Peng, Lihan Jiang, Ningyuan Tang, Shengming Yin, Tianhe Wu, Xiaoyue Chen, Yan Shu, Yanran Zhang, Yi Wang, Yu Wu, Yujia Wu, Zekai Zhang, Zhendong Wang, Xiao Xu, Kun Yan, and Chenfei Wu. Qwen-image-2.0-rl technical report, 2026. URL https://arxiv.org/abs/2606.27608.

Shiyuan Yang, Liang Hou, Haibin Huang, Chongyang Ma, Pengfei Wan, Di Zhang, Xiaodong Chen, and Jing Liao. Direct-a-video: Customized video generation with user-directed camera movement and object motion. In *ACM SIGGRAPH 2024 Conference Papers*, 2024. doi: 10.1145/3641519.3657481.

Guy Yariv, Yuval Kirstain, Amit Zohar, Shelly Sheynin, Yaniv Taigman, Yossi Adi, Sagie Benaim, and Adam Polyak. Through-the-mask: Mask-based motion trajectories for image-to-video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18198–18208, June 2025.

Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Cheng Zhang, Boying Li, Meng Wei, Yan-Pei Cao, Camilo Gambardella, Dinh Phung, and Jianfei Cai. Unified camera positional encoding for controlled video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 38027–38037, June 2026.

Hong Zhang, Zhongjie Duan, Xingjun Wang, Yingda Chen, and Yu Zhang. Eligen: Entity-level controlled image generation with regional attention, 2025a.

Hongfei Zhang, Kanghao Chen, Zixin Zhang, Harold Haodong Chen, Yuanhuiyi Lyu, Yuqi Zhang, Shuai Yang, Kun Zhou, and Yingcong Chen. Dualcamctrl: Dual-branch diffusion model for geometry-aware camera-controlled video generation. *arXiv preprint arXiv:2511.23127*, 2025b.

Hui Zhang, Dexiang Hong, Yitong Wang, Jie Shao, Xinglong Wu, Zuxuan Wu, and Yu-Gang Jiang. Creatilayout: Siamese multimodal diffusion transformer for creative layout-to-image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025c.

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

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Jpxfof0EaS.

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.

Sixiao Zheng, Minghao Yin, Wenbo Hu, Xiaoyu Li, Ying Shan, and Yanwei Fu. Versecrafter: Dynamic realistic video world model with 4d geometric control. *arXiv preprint arXiv:2601.05138*, 2026.

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

Wei Zhou, Xiongwei Zhu, Zelin Xu, Bo Dong, Lixue Gong, Yongyuan Liang, Meng Chu, Leigang Qu, Lingdong Kong, Wei Liu, et al. Danceopd: On-policy generative field distillation. *arXiv preprint arXiv:2606.27377*, 2026.

A Appendix

A.1 Data Curation Details

We provide additional details of the data curation pipeline described in section 2 and fig. 2.

Metadata Filtering and Clip Extraction. We curate our dataset from SpatialVID-HQ (Wang et al., 2025b), Sekai-HQ (Li et al., 2026e), and RealEstate10K (Zhou et al., 2018). For SpatialVID and Sekai, we first remove crowded scenes using the scene metadata, since dense crowds often lead to ambiguous object correspondence and noisy layout annotations. For SpatialVID, we additionally remove broad natural-landscape categories that contain few meaningful foreground objects for layout control. For RealEstate10K, we retain only clips with at least 16 FPS, a duration of at least 5 seconds, and a resolution no smaller than 352×640. Then, we extract candidate windows from source videos. To match our training configuration, each source video is resampled to 16 fps and cut into 81-frame windows (≈ 5 s) with a stride of 40 frames.

Camera-Motion Filtering. We compute the FoV expansion ratio and accumulated translation distance using $K = 8$ uniformly sampled keyframes. For r_{FoV}, we use $50{,}000$ directions sampled on the unit sphere; a direction is visible in a keyframe if it projects inside the image with positive depth under that frame's pinhole camera. For intuition, $r_{\mathrm{FoV}} = 1$ indicates no expansion of angular viewing coverage, while $r_{\mathrm{FoV}} \approx 1.5$, 2.0, and 3.0 roughly correspond to $45°$, $90°$, and $180°$ yaw rotations, respectively, for a $90°$ horizontal FoV. Because the original camera annotations from different source datasets follow different translation scales, we use dataset-specific translation thresholds during the initial filtering stage. Specifically, candidate windows are retained if they satisfy either the FoV or translation criterion:

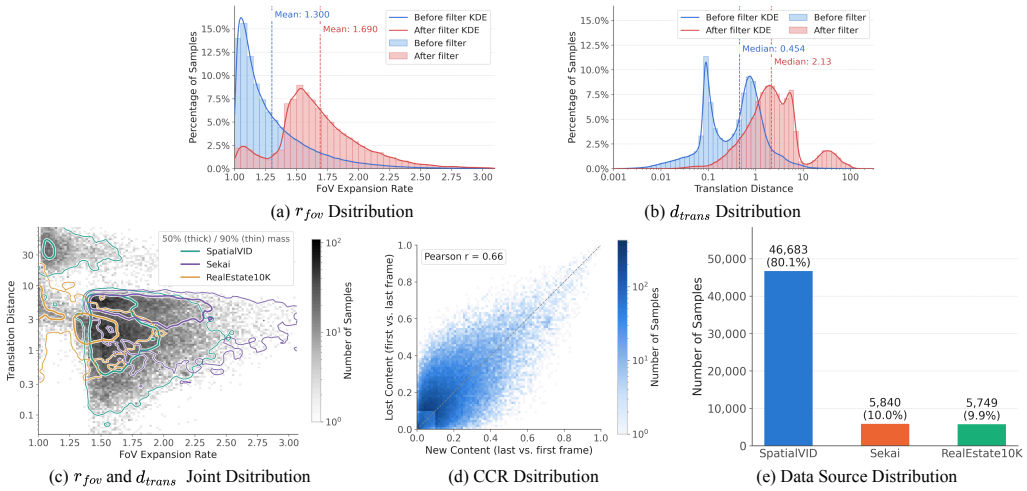
Dataset	r_{FoV}	d_{trans}
SpatialVID	≥ 1.4	≥ 2.0
Sekai	≥ 1.4	≥ 0.16
RealEstate10K	≥ 1.3	≥ 10.0

To reduce highly redundant windows from long source videos and keep diversity, we further rank candidates with r_{FoV} in decreasing order, apply temporal non-maximum suppression, and retain at most two non-overlapping windows with the highest r_{FoV} from each source video.

Content-Change Filtering. To better match our target setting of future-view layout control, we further apply content-change filtering to focus on clips with substantial future-region revelation for Stage 2 and 3 training, where the camera motion exposes content that is not visible in the first frame. The content change ratios are computed between the first and last frames with DINOv2-giant (Oquab et al., 2023). We resize the frames without cropping while preserving their aspect ratio so that the longer side is approximately 574 px (a multiple of the 14 px patch size). For every patch in one frame, we search for its most similar patch in the other frame and use a cosine-similarity threshold of $\tau = 0.5$. We compute both directions, corresponding to newly revealed content in the last frame ($r_{lf\text{-}CCR}$) and content leaving the first frame ($r_{ff\text{-}CCR}$). We remove clips with $r_{\mathrm{FoV}} < 1.5$, $r_{lf-CCR} < 0.1$ and $r_{ff-CCR} < 0.1$, i.e. clips with limited expansion of angular viewing coverage whose content is nearly unchanged in both directions.

Layout Annotation. We annotate object layouts in the last frame using Qwen3-VL-32B (Bai et al., 2025c), which we find produces more reliable and selective annotations of meaningful foreground objects. We prompt the model to select salient and spatially meaningful foreground instances while excluding tiny clutter, background regions, severely occluded objects, and excessively large regions. Clips with no valid candidates are discarded.

Data Statistics. After curation, we obtain 120,898 training samples for Stage 1, a subset of 58,272 samples for Stages 2 and 3, and 600 test samples. The test set is randomly sampled from the curated data according to three FoV-expansion ranges, $[1.0, 1.5)$, $[1.5, 2.0)$, and $[2.0, \infty)$, with a sampling ratio of 1:2:2, so as to cover different levels of viewpoint change. All test samples are excluded from the training sets. The distribution of the curated dataset is visualized in fig. 6. The resulting layout annotations contain an average of 5.5 objects per clip and cover $8,302$ categories, including indoor objects (chair, window, lamp, cabinet, sofa, etc.) and outdoor objects (person, building, car, boat, tree, sign, etc.). 84.8% of clips contain objects that are invisible in the first frame but appear in future views, directly supporting our future-view layout-control setting.



(a) r_{fov} Dsitribution (b) d_{trans} Dsitribution

(c) r_{fov} and d_{trans} Joint Dsitribution (d) CCR Dsitribution (e) Data Source Distribution

Figure 6: **Visualization of data distribution.** (a, b) Distributions of the camera-motion metrics over all candidate windows before filtering (blue) and over the final layout training set (red). Filtering removes the mass of near-static windows and shifts both metrics toward larger viewpoint changes. (c) Joint distribution of the two camera-motion metrics in the training set, with 50% and 90% mass contours per source dataset. (d) Joint distribution of the two content change ratios. The two directions are correlated and complementary but not redundant. (e) Number of training clips per dataset.

Table 5: **Training configuration of LIFT across the three stages.**

Setting	**Stage 1**	**Stage 2**	**Stage 3**
Training mode	Camera Control (SFT)	Dense Layout Control (SFT)	Dual-Mode OPSD
Base Model	Wan2.1-Fun-V1.1-1.3B-Control-Camera	Stage 1	Stage 2
Dataset size	120,898	58,272	58,272
Training steps	8,000	4,000	500
Global batch size	32	32	16
Learning rate	1×10^{-5}	1×10^{-4}	5×10^{-5}
Optimizer	AdamW	AdamW	AdamW

Table 6: **State selection of OPSD.**

Method	**Video Quality**			**Camera Error**		**Semantic Consistency**			**Time**
	FVD ↓	FID ↓	LPIPS ↓	RotErr ↓	TransErr ↓	mIoU ↑	SR_e ↑	$\text{CLIP}_{\text{local}}$ ↑	s/step ↓
All 50 rollout states	127.24	<u>15.02</u>	0.441	3.467	0.624	0.4786	0.5740	<u>0.2234</u>	544.82
Equal 10 states	<u>120.23</u>	15.17	<u>0.439</u>	3.228	0.589	0.4807	<u>0.5777</u>	0.2223	185.28
Late 10 states	197.63	17.82	0.494	6.347	0.906	0.3089	0.5227	0.2024	183.56
Prefix 10 states	127.02	17.77	0.442	<u>2.961</u>	**0.544**	<u>0.4922</u>	0.5664	0.2219	**117.21**
Prefix 10 states + SFT anchor	**102.07**	**13.07**	**0.422**	**2.879**	<u>0.583</u>	**0.4934**	**0.5883**	**0.2313**	<u>125.20</u>

A.2 More Experiments

A.2.1 More Implementation Details

We summarize the training configuration of the three stages in table 5. Stage 1 establishes camera controllability, Stage 2 learns dense spatiotemporal layout control, and Stage 3 transfers the dense-layout capability to last-frame-layout and camera-only modes through dual-mode OPSD. All stages are trained using AdamW on 4 NVIDIA H100 GPUs at a resolution of 352×640 with 81-frame clips at 16 FPS.

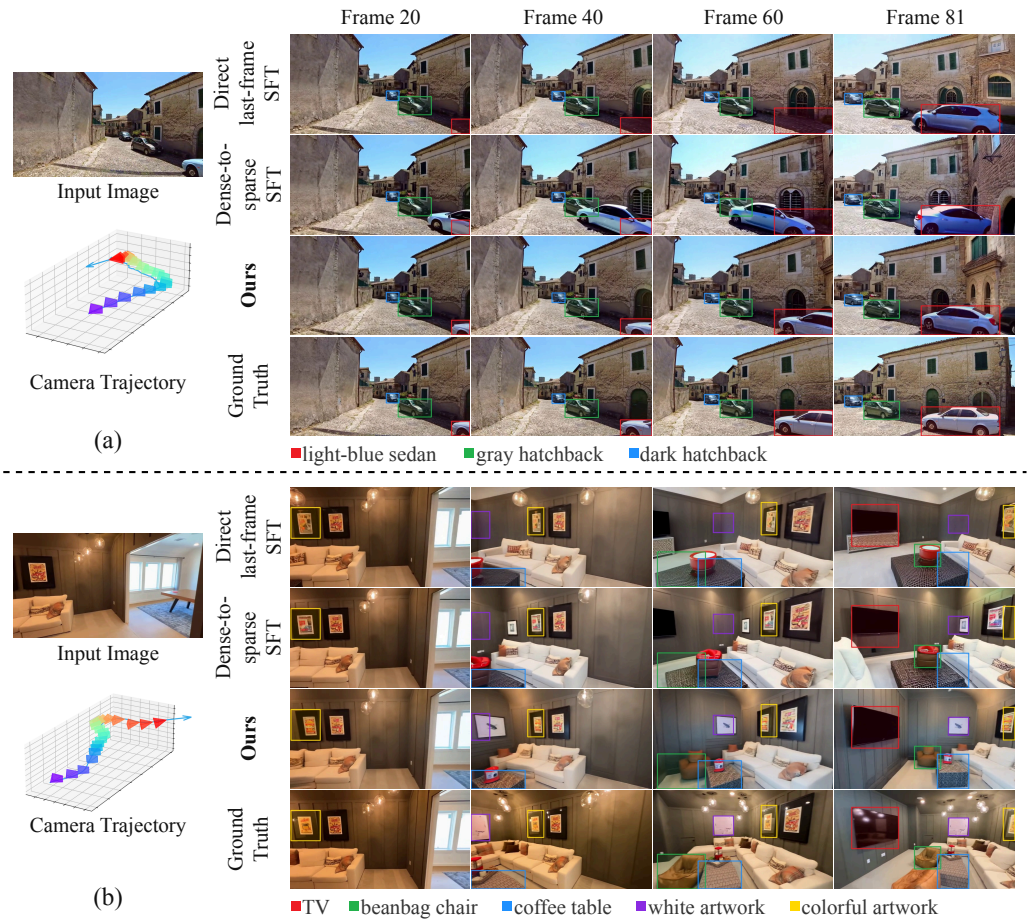


Figure 7: **SFT and OPSD Comparison.** (a) Direct last-frame SFT largely ignores the sparse layout condition, leaving the target sedan absent until the final frame, while dense-to-sparse SFT introduces it too early and with inaccurate spatial alignment. In contrast, OPSD produces a trajectory that better matches the ground-truth evolution. (b) Direct last-frame SFT fails to realize several conditioned objects, whereas dense-to-sparse SFT improves object presence but still exhibits poor temporal alignment before the last frame. OPSD more faithfully follows both the target layout and its temporal evolution.

A.2.2 Evaluation Metrics.

We evaluate the generated videos along three dimensions: visual quality, camera controllability, and layout controllability.

Visual quality. We report Fréchet Video Distance (FVD) (Unterthiner et al., 2018), Fréchet Inception Distance (FID) (Heusel et al., 2017), and Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018). FVD measures the distributional discrepancy between generated and real videos in a learned video feature space, jointly reflecting visual realism and temporal coherence. FID measures the distributional similarity between generated and real frames in the image feature space and primarily evaluates frame-level visual fidelity. LPIPS measures the perceptual distance between generated and corresponding reference frames using deep visual features. Lower values indicate better performance for all three metrics.

Camera controllability. Following (Zhang et al., 2025b), we evaluate camera trajectory accuracy using rotation error (RotErr) and translation error (TransErr). RotErr measures the angular discrepancy between the estimated camera rotations of the generated video and the target camera trajectory, while TransErr measures their translation discrepancy. We use the off-the-shelf model (Lin et al.,

Table 7: **Implementation details for the SFT and OPSD ablations.** Direct LF-SFT denotes direct last-frame SFT. D2S-SFT denotes dense-to-sparse curriculum SFT. For a fair comparison, all three methods are counted from the same Stage 1 camera-control checkpoint, and the dense-layout SFT stage required before OPSD is included in the training cost of our method. Sample updates are computed as training steps $\times$ global batch size over the full adaptation process.

Setting	Direct LF-SFT	D2S-SFT	OPSD (Ours)
Starting checkpoint	Stage 1	Stage 1	Stage 1
Training schedule	8K LF	4K Dense + 2K 8-frame + 2K LF	4K Dense + 500 OPSD
Global batch size	32	32	32 (Dense) / 16 (OPSD)
Learning rate	1×10^{-4}	1×10^{-4}	1×10^{-4} (Dense) / 5×10^{-5} (OPSD)
Total training steps	8,000	8,000	4,500
Sample updates (Steps $\times$ BS)	256K	256K	136K
Dataset size in pool	58,272	58,272	58,272
Optimizer	AdamW	AdamW	AdamW

2025) to estimate the camera pose of generated videos and then compare with the ground truth. Lower RotErr and TransErr indicate more accurate adherence to the prescribed camera motion.

Layout controllability. Following OverLayBench (Li et al., 2026a), we evaluate object-level spatial and semantic control using mean Intersection over Union (mIoU), Success Rate of Entity (SR_e), and $CLIP_{local}$ (Radford et al., 2021). We use Qwen3.6-27B (Bai et al., 2025c) to detect objects and their locations in the generated videos. mIoU measures the spatial overlap between generated object regions and their target layout boxes, reflecting object placement accuracy. SR_e measures the fraction of conditioned entities that are successfully generated with the intended semantics and spatial placement. $CLIP_{local}$ computes the CLIP similarity between local regions corresponding to conditioned objects and their text descriptions, measuring local semantic consistency. Higher mIoU, SR_e, and $CLIP_{local}$ indicate better layout controllability.

A.2.3 More Ablation Studies

SFT vs. OPSD. A more detailed training cost comparison between SFT and OPSD is illustrated in table 7. OPSD is substantially more data-efficient and effective at achieving the last-frame layout control. The qualitative comparisons in fig. 7 illustrate the layout-following limitations of the evaluated SFT baselines in our setting. Because the last-frame layout provides only a weak endpoint signal, direct last-frame SFT often ignores conditioned objects. Dense-to-sparse SFT partially alleviates this issue, but the model still has to infer, from the last-frame layout alone, how the specified objects should emerge and evolve throughout the preceding frames under camera motion. In contrast, OPSD provides direct supervision from a dense-layout teacher on the student's own rollout states, supplying explicit guidance for the intermediate scene evolution and making the sparse future-layout condition substantially easier to learn.

Dual-mode OPSD. The qualitative comparisons are shown in fig. 8. Lastframe-layout single-mode OPSD learns to follow the future layout, but specializing only to this conditioning mode can yield lower visual quality than dual-mode OPSD under camera-only inference, e.g., causing scene drift or geometric distortion. Conversely, camera-only single-mode OPSD maintains strong camera-conditioned generation but lacks sufficient supervision for future-layout control, often producing incorrect objects or failing to place them at the specified locations. By alternating between both modes and distilling from the same dense-layout teacher, dual-mode OPSD preserves camera controllability while retaining accurate future-layout control, leading to more consistent behavior across both inference settings.

ODE State Sampling Strategy in OPSD. We study where along the ODE trajectory the teacher provides the most effective distillation signal. As visualized in fig. 9, during the early high-noise stage, the dense-layout teacher produces clear corrections to the student-visited state, especially in the global object layout and spatial arrangement. In contrast, after roughly the first 10 denoising steps, the student and teacher predictions become much closer, and the teacher correction is significantly weaker and provides less informative distillation signals. In our ablation, prefix-state distillation provides a favorable trade-off between training time, camera accuracy, and spatial layout control. Based on this observation, we select the first 10 high-noise rollout states for OPSD. This selective state sampling focuses optimization on the phase where the global layout structure is es-

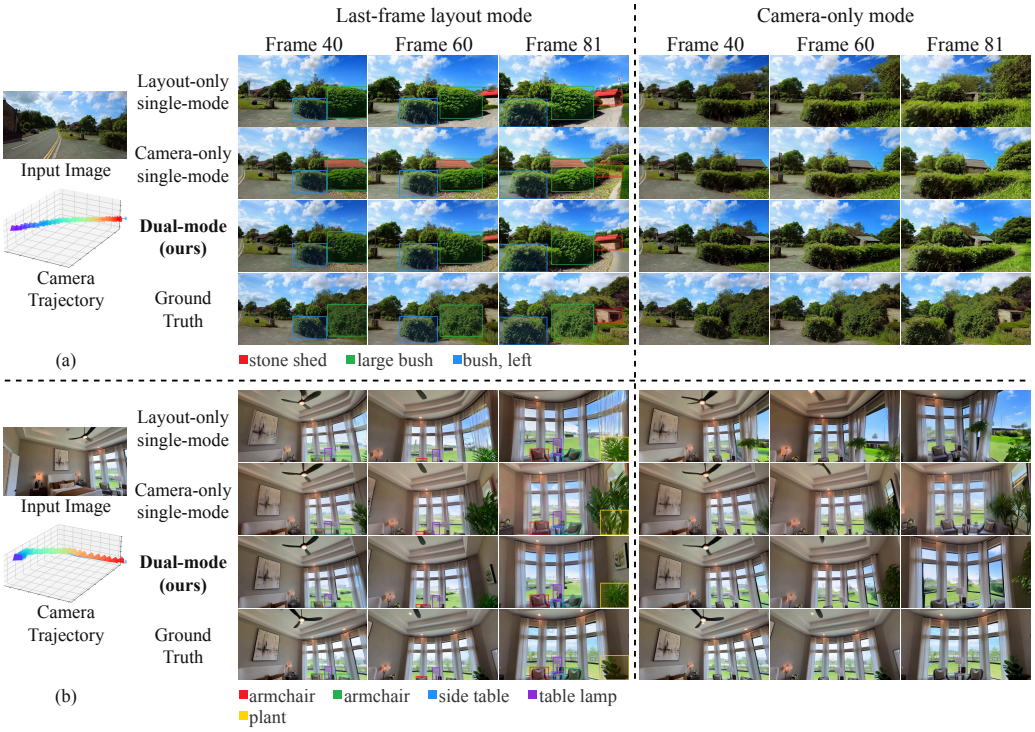


Figure 8: **Dual-mode OPSD Comparison.** The last-frame-layout mode uses both the camera trajectory and last-frame layout, whereas the camera-only mode uses only the camera trajectory. Last-frame-layout single-mode OPSD follows the prescribed layout, but yields lower video quality than dual-mode OPSD in camera-only inference, while camera-only single-mode OPSD preserves camera control but often fails to realize the specified future layout. In contrast, dual-mode OPSD maintains strong camera control in both inference settings while faithfully following the future-layout condition when provided.

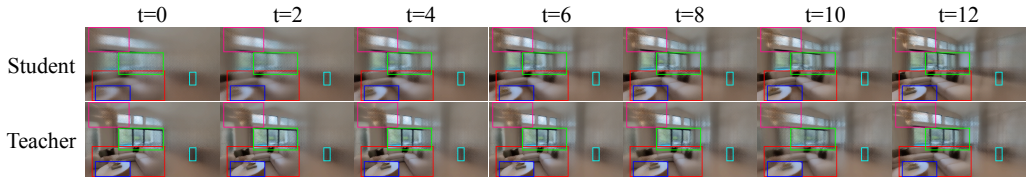


Figure 9: **Student–teacher comparison** in dense-to-lastframe layout OPSD. The labels $t = 0, 2, \ldots, 12$ denote denoising-step indices rather than the continuous diffusion time used in the equations. The global layout structure is largely determined during the early high-noise stage. Within the first 10 denoising steps, the dense-layout teacher provides substantial corrections on the student-visited states, producing results that better conform to the specified layout. Afterwards, the teacher correction becomes much weaker, motivating us to concentrate OPSD supervision on the first 10 high-noise states.

tablished while reducing the cost of student rollout and distillation. This ablation is conducted on lastframe-layout single-mode OPSD. Distilling only the last 10 states performs substantially worse across all quality and controllability metrics, indicating that low-noise states provide little useful signal for correcting the global layout. In contrast, supervising the prefix 10 high-noise states yields substantially better camera and layout control. Combined with the SFT anchor, prefix-10-state distillation achieves the strongest overall performance while introducing little computation overhead. This is substantially more efficient than querying all 50 rollout states. These results support our observation that the dense-layout teacher provides its most informative corrections during the early

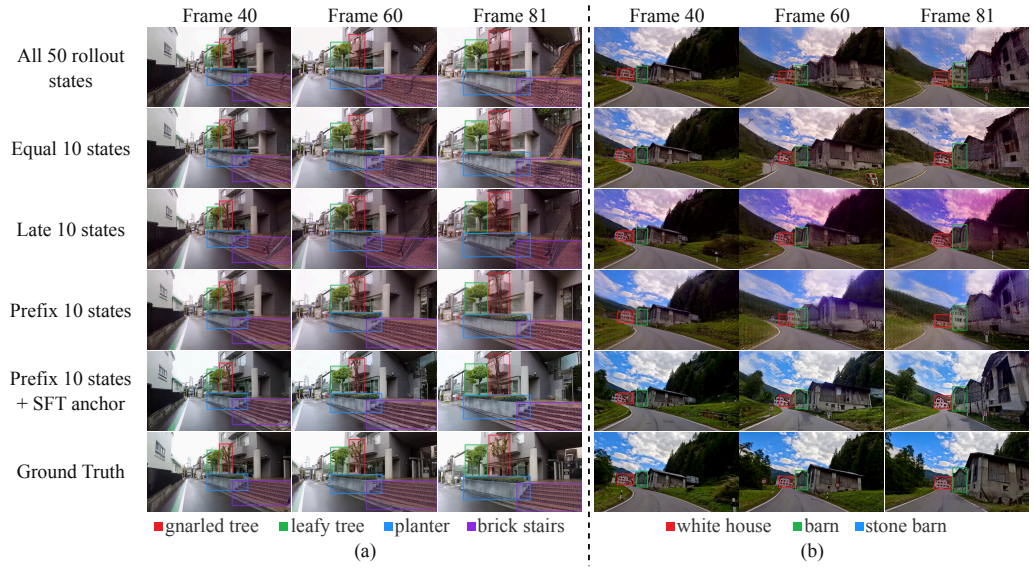


Figure 10: **Comparison of different ODE state sampling in OPSD.** Distilling only on the late 10 states (the third row) yields the weakest layout following, indicating that global layout structure is mainly determined during the early high-noise stage. Prefix-state distillation provides stronger layout control, while adding the SFT anchor further prevents drifting and visual-quality degradation.

Table 8: **Ablation of the proposed camera-motion filtering strategy.** We compare models trained on randomly sampled and filtered data under the same training configuration. The proposed filtering strategy consistently improves both visual quality and camera-control accuracy.

Method	FVD ↓	FID ↓	LPIPS ↓	RotErr ↓	TransErr ↓
w/o filter	278.85	32.88	0.52	15.62	4.05
w/ filter	**266.58**	**30.99**	**0.51**	**13.14**	**3.87**

high-noise stage, where the global scene and layout structure are primarily established. Visualization comparisons are shown in fig. 10.

Data Filtering. Table. 8 and Fig. 11 demonstrate the effectiveness of our camera-motion filtering strategy. For a fair comparison, we construct two equally sized training sets of 30K clips, one using our proposed filtering strategy and the other using random sampling. We additionally hold out a test set of 250 clips featuring substantial future-region revelation. Compared with random sampling, our filtered training data consistently improves all evaluation metrics, including video quality and camera-control accuracy, and better preserves scene structure and object appearance under large viewpoint changes. These results highlight the importance of curating training data with large viewpoint changes for our target setting.

A.2.4 User Study

To complement automatic evaluation, we conduct a randomized user study with 7 participants. For each trial, participants are shown the input condition and anonymized generated videos from different methods in random order, and are asked to choose the result that best satisfies the control intent, considering camera motion, layout adherence, object consistency, and visual quality. We compare with the best two methods in table 1 in this user study. Each participant evaluates 30 comparisons, yielding 210 responses in total. Participants may also select a "None preferred" option, and the reported preference rates are computed over responses that select one of the three methods. As shown in table 9, our method achieves the highest preference rate, 65.96%. This result indicates that the advantages of our method are not only reflected in automatic metrics but are also perceptible to hu-



Figure 11: **Comparison of the proposed camera-motion filtering strategy.** The first row shows results from the model trained on randomly sampled clips, exhibiting degraded visual quality and less stable object appearance. The second row shows results from the model trained on filtered dynamic clips, with better preservation of scene structure and object appearance under camera motion. The third row shows ground-truth frames.

Table 9: **User study** on preference rates among different camera-controllable video generation methods. Participants select the result that best satisfies interactive editing requirements.

	Ours	Uni3C	MagicMotion
Preference	**65.96%**	22.34%	11.70%

man observers. The user study further confirms that our approach produces visually convincing and controllable videos that better align with user intent.

A.3 LIMITATION

LIFT currently uses 2D bounding boxes with local text prompts to specify the desired future-view composition. Although this representation is simple and user-friendly, it provides only coarse spatial constraints and does not explicitly capture depth, orientation, or occlusion relationships between objects. Extending the layout representation with richer geometric or instance-level controls could enable more fine-grained specification of future views.