PISCO: Precise Video Instance Insertion with Sparse Control

Xiangbo Gao[1] **Renjie Li**[1] **Xinghao Chen**[1] **Yuheng Wu**[3]
Suofei Feng[4] **Jie Yang**[2] **Qing Yin**[2] **Zhengzhong Tu**[1,2]

[1]Texas A&M University [2]Visko Platform
[3]KAIST [4]Stanford University

Abstract

AI video generation is moving beyond general generation, which relies on exhaustive prompt-engineering and "cherry-picking", towards fine-grained, controllable generation and high-fidelity post-processing. In professional AI-assisted filmmaking, the core requirement is the ability to perform precise, targeted modifications. A key task is video instance insertion, which requires precise spatial-temporal placement, physically consistent scene interaction (e.g., shadows and reflections), and the faithful preservation of original dynamics - all achieved under minimal user effort. In this paper, we propose PISCO, a video diffusion model for precise video instance insertion with arbitrary sparse keyframe control. PISCO allows users to specify a single keyframe, start-and-end keyframes, or sparse keyframes at arbitrary timestamps, and automatically propagates object appearance, motion, and interaction. To stabilize generation under sparse conditioning, we introduce Variable-Information Guidance and Distribution-Preserving Temporal Masking, complemented by geometry-aware conditioning. We further construct PISCO-Bench, a benchmark with verified instance annotations and paired clean background videos, and evaluate performance using both reference-based and reference-free perceptual metrics. Experiments demonstrate that PISCO consistently outperforms existing baselines and scales effectively as additional control signals are provided.

1 Introduction

Recent advances in large-scale video generative models [63, 24] are able to generate videos with high visual fidelity and realistic motion. As these models continue to evolve, the focus of video generation is shifting beyond producing visually plausible content toward highly controllable video generation and editing [57, 79, 73], with the long-term goal of enabling **Hollywood-grade AI-assisted filmmaking** [55, 23]. In this new paradigm, the priority is no longer to **cherry-pick** a barely acceptable output from numerous generations, but to reliably **achieve precise user intent** with minimal iteration.

A particularly demanding yet under-explored capability is **Precise Video Instance Insertion**: inserting a specific object into an existing video at a user-specified spatial location and temporal position, while preserving the identity and dynamics of the original video. In professional video editing workflows, instance insertion is not merely about adding a visually plausible object, but about achieving precise control with minimal iteration. In this setting, precision entails several requirements:

❶ **Instance-level controllability**, enabling users to specify when and where an object appears;
❷ **Physically plausible temporal propagation**, where the inserted object automatically evolves over time with coherent pose and motion, following physically reasonable dynamics;
❸ **Consistent scene adaptation:** adjusting the background for insertion-induced effects (e.g., shadows, reflections) [44];

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



Figure 1: PISCO enables precise video instance insertion with arbitrary sparse keyframe control. Given a clean input video and a few user-provided instance cutouts at selected timestamps, PISCO inserts the instance with coherent temporal propagation and physical effects while preserving the original background dynamics.

- ❹ **Background dynamics preservation:** maintaining original motions, identities, and temporal patterns without disruption;
- ❺ **Low-effort user interaction**, where achieving the above goals does not require dense per-frame annotations or manual editing across the entire video.

Despite recent advances [75], these requirements remain largely unmet by existing paradigms. **Video inpainting** [59, 86] demands exhaustive per-frame mask annotations (❺), and its copy-and-fill nature struggles with holistic scene adaptation like realistic shadowing (❸). **Video-to-video editing** [71, 35] propagates appearance but lacks fine-grained spatiotemporal controllability for precise object placement (❶). **Agentic pipelines** (image editing + I2V) [31, 37] discard temporal context, causing severe background drift and disrupted scene dynamics (❹). Finally, **geometry-heavy 4D approaches** [53, 36] resolve spatial relationships but incur massive computational overhead, rely on fragile geometric priors, and struggle with complex dynamic object motions (❷).

To overcome these limitations, we propose PISCO (Precise Instance insertion with Sparse COntrol), a video diffusion framework that accepts instance RGB, masks, depth, and availability signals at arbitrary sparse keyframes. Sparse control, however, creates distribution shifts in pretrained temporal VAEs when missing frames are naïvely masked. PISCO addresses this with **Variable-Information Guidance (VIG)** and **Distribution-Preserving Temporal Masking (DPTM)**, enabling robust conditioning, coherent propagation, and scene-consistent interactions.

We further construct PISCO-Bench from BURST [6], with verified masks and paired clean background videos. Reference-based and reference-free experiments show that PISCO outperforms strong inpainting and editing baselines, with consistent gains as more sparse control frames are provided.

2 Related Works

Precise video instance insertion is related to video inpainting, reference-guided video editing, image-to-video edit propagation, and 3D/4D object insertion. Inpainting methods [59, 86, 8, 28] can preserve local context but typically formulate insertion as masked-region completion, relying on dense masks or keyframe edits rather than sparse instance-level RGB/depth controls. Reference-guided video editors [71, 35, 69, 18, 62, 15] improve appearance conditioning, but still struggle to jointly control instance identity and spatial-temporal placement. Image-editing plus I2V propagation pipelines [52, 11, 72, 51] can animate edited keyframes, but may regenerate full-frame content and introduce background drift. Geometry-based methods [53, 7, 36] provide explicit placement cues,

yet depend on reliable 3D/4D reconstruction and are less flexible for dynamic in-the-wild insertion. In contrast, PISCO uses sparse keyframe RGB/mask/depth conditions to anchor both appearance and location while preserving the clean video as persistent context. A complete related-work discussion is provided in the Appendix.

3 Methodology

3.1 Problem Formulation

Given paired videos $\{\hat{V}, V\}$ of length T, where $\hat{V} = \{\hat{V}_t\}_{t=1}^T$ contains a foreground instance and $V = \{V_t\}_{t=1}^T$ depicts the same scene without that instance, our goal is to synthesize an edited video $\tilde{V}$ in which the instance is naturally inserted into V with spatially accurate placement, temporally coherent motion, and geometrically plausible occlusion relationships. We represent the instance-side conditions as an RGB instance clip $I = \{I_t\}_{t=1}^T$, its spatial mask $M = \{M_t\}_{t=1}^T$, and the instance depth $D_I = \{D_{I,t}\}_{t=1}^T$. Here, each I_t contains only the segmented foreground instance on a constant zero-valued background, and M_t indicates the corresponding foreground region. In addition, we compute a background depth map D_V from V to provide geometric cues for depth ordering and occlusion reasoning. Details of how these signals are constructed are described in Sec. 4.

Variable-density user guidance via availability mask. User guidance can range from a single keyframe to dense per-frame supervision. We model this variability with an availability mask $A = \{A_t\}_{t=1}^T$, where $A_t \in \{0, 1\}$ indicates whether instance-side information is available at time t. During training, we sample $A \sim p_\gamma(A)$, where the density ratio $\gamma \in [0, 1]$ controls the expected fraction of available frames as well as their temporal placement. The mask is applied exclusively to instance-specific signals:

$$I^A = A \odot I, \quad M^A = A \odot M, \quad D_I^A = A \odot D_I, \tag{1}$$

while the background depth D_V remains unmasked. This design reflects practical scenarios: while a user specifies instance-level details sparsely, the background scene geometry is often fully observable.

Conditional video diffusion with context adapter. Let z_0 denote the latent representation of the target video obtained using a pretrained temporal video VAE, and let z_t be the noisy latent at diffusion step t. We train a denoising network ϵ_θ to predict the added noise conditioned on the background video and the masked instance-side signals, ending up with the training objective

$$\mathcal{L} = \mathbb{E}_{z_0, \epsilon, t, A \sim p_\gamma} \left[\left\| \epsilon - \epsilon_\theta\big(z_t,\, t,\, V,\, D_V,\, I^A,\, D_I^A,\, M^A,\, A\big) \right\|_2 \right]. \tag{2}$$

In our implementation, ϵ_θ uses the Wan video diffusion backbone [63] with a VACE context adapter [35] for injecting multi-channel conditions. To accommodate our condition format, we modify the first linear layer of the VACE adapter to match the new input dimensionality, while reusing pretrained parameters for the remaining layers. Training details are provided in the Appendix.

3.2 Variable-Information Guidance via Dynamic Contextual Dropout

To ensure robust instance propagation across varying levels of user control, we introduce Variable-Information Guidance (VIG), a dynamic contextual dropout strategy that samples the availability mask $A \sim p_\gamma(A)$ during training. Rather than applying uniform dropout, we employ a hybrid sampling strategy over the density ratio γ to expose the model to four distinct supervision: *Extreme Sparsity* ($p = 0.4$, retaining exactly one frame), *Sparse* ($p = 0.3, \gamma \in [0.02, 0.1]$), *Dense* ($p = 0.2, \gamma \in [0.2, 0.8]$), and *Anchor* ($p = 0.1$, full per-frame supervision with $A_t \equiv 1$). This targeted sampling enables a smooth trade-off: under sparse conditions, the model learns to infer plausible motion and interactions by leveraging the background context, while the dense and anchor regimes prevent identity drift and preserve detailed instance appearance when strong supervision is available.

3.3 Distribution-Preserving Temporal Masking (DPTM)

Modern video generative models [63, 16, 48] often rely on pretrained temporal video VAEs [76, 66, 84, 1] that compress the temporal dimension by a factor C_t. Recent arbitrary-frame conditioning methods [13, 47] address a related temporal anchoring issue by assigning sparse condition tokens to target timestamps through temporal positional indexing, such as Temporal RoPE interpolation [13]

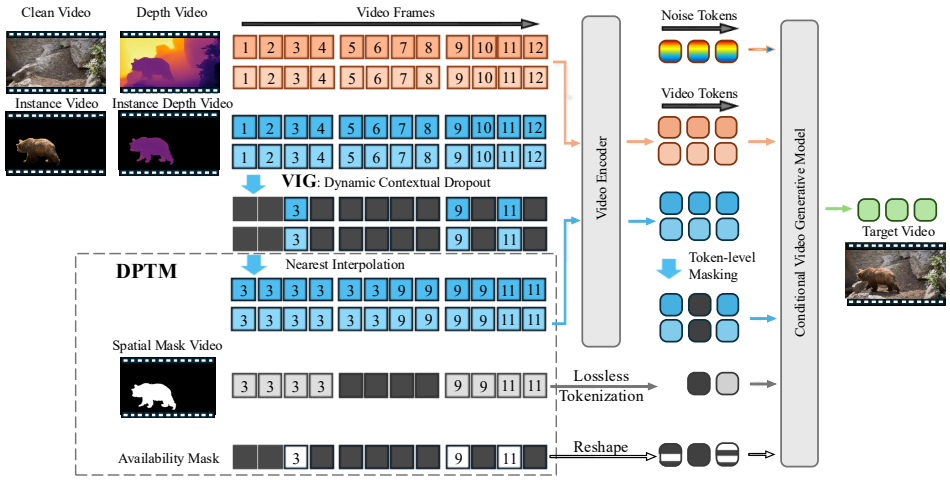

Figure 2: **Overview of PISCO pipeline.** We train a conditional video diffusion model with sparse keyframe control via Variable-Information Guidance (VIG), and stabilize the training with pretrained temporal VAEs using Distribution-Preserving Temporal Masking (DPTM): pixel-space nearest-frame interpolation followed by token-level masking, with spatial mask and availability signals injected alongside RGB/depth conditions.

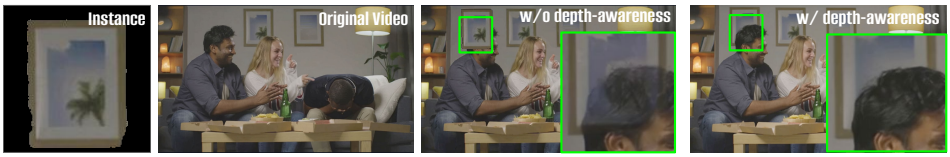

Figure 3: **Visualization of the depth-aware insertion**. Conditioning on depth improves depth ordering and occlusion handling, reducing blending artifacts compared to a depth-agnostic variant.

or shared RoPE [47]. However, positional indexing alone is insufficient for our localized instance insertion setting, where the inserted foreground may occupy only a small spatial region and can be diluted by spatiotemporal VAE compression when encoded together with mostly background or unavailable condition frames. If we naïvely mask missing frames using A in pixel space, the temporal VAE receives out-of-distribution inputs, which leads to flickering, miscoloring, and incomplete instance rendering as illustrated in Figure 4. To address this, we propose **Distribution-Preserving Temporal Masking (DPTM)** to decouple distribution preservation from information masking, while staying fully consistent with the sampled availability mask A.

Pixel-space temporal completion. As shown in Figure 2, given a masked instance clip I^A, we first perform nearest interpolation in pixel space. Concretely, missing frames are filled by propagating the temporally nearest available instance frame forward and backward. This produces a temporally complete clip with natural video statistics, which helps keep the input distribution compatible with pretrained temporal VAEs.

Token-space masking. We then encode the interpolated clip into video tokens. Tokens corresponding to the originally unobserved frames are masked out in the latent space. This explicit masking enables the model to distinguish valid signals from interpolated fillers, thereby accelerating convergence.

Availability channel aligned to token resolution. To distinguish observed frames from interpolated ones, we introduce an availability channel. Let $T' =$

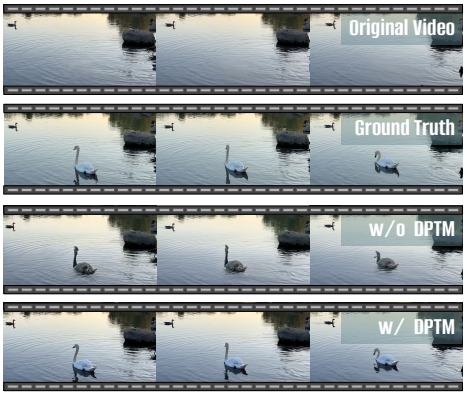

Figure 4: **DPTM under sparse guidance.** With segmented instance inputs on odd frames only, naive masking causes severe temporal artifacts, while DPTM preserves encoder input statistics and improves temporal stability.

$\lceil T/C_t \rceil$ be the compressed token length and (H', W') be the spatial resolution. We reshape the binary frame-level availability mask $A \in \{0,1\}^{T\times H\times W}$ into: $A \in \mathbb{R}^{C_t\times T'\times H'\times W'}$. This transformation moves the local temporal window C_t into the channel dimension, preserving fine-grained observability patterns within each compressed token. The resulting tensor is concatenated with other conditions and processed by the VACE context adapter. Combined with pixel-space completion and token-space masking, this strategy enables precise sparse conditioning and minimizes temporal artifacts, as evidenced in Figure 4.

3.4 Geometry- and Appearance-Robust Training

We further incorporate three strategies to improve geometric plausibility and appearance robustness: depth-aware conditioning, occlusion-aware completion augmentation, and relighting augmentation.

Depth-aware insertion. Purely 2D video editing lacks explicit geometric reasoning, making scale, depth ordering, and occlusion ambiguous. We therefore condition the model on two depth signals: the background depth D_V computed from V, and the instance depth D_I extracted from $\hat{V}$ using mask M and masked by availability as D_I^A. These depth cues are provided alongside RGB, spatial mask, and availability signals, enabling the model to reason about relative depth ordering and produce physically plausible layer compositing. As shown in Figure 3, Depth conditioning helps the model respect depth ordering and occlusion relations during insertion.

Amodal instance augmentation. During inference, users typically provide spatially complete (amodal [4, 3]) instances, expecting the model to resolve occlusions based on scene geometry. However, training instances extracted directly from segmentation are often fragmented by scene occlusions. To align training with this inference requirement, we introduce an amodal augmentation strategy. Specifically, we reconstruct missing regions to generate a complete pseudo-amodal instance as the input condition, while supervising the model with the original occluded video. This pairing forces the model to learn the compositing logic required to correctly occlude a complete input instance according to depth cues. Details of the completion construction are provided in Sec. 4. Some examples of amodal completion results are shown in Figure 5.

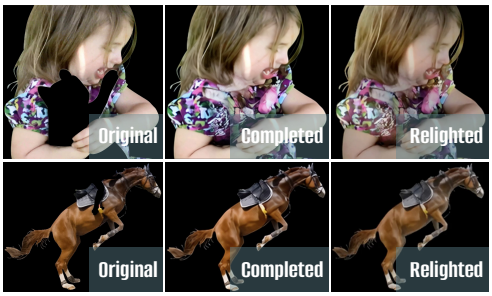

Figure 5: **Amodal instance completion and relighting augmentations.** We complete occluded instance cutouts to form pseudo-amodal inputs and apply relighting to improve illumination adaptation to the target scene.

Instance relighting augmentation. In video instance insertion, strictly preserving the provided instance appearance can lead to incompatible lighting between the instance and the background scene. To improve automatic lighting adaptation, we augment training with relighted instances generated by IC-Light [80]. For each instance clip, we synthesize a relighted version $\tilde{I}$ under randomly sampled background lighting conditions. The relighted instance $\tilde{I}$ is then used as additional training data for instance conditioning. Figure 6 shows the qualitative results of the relighting augmentation.



Figure 6: **Relighting augmentation.** Without relighting, the model over-preserves the input instance appearance, causing illumination mismatch. Relighting augmentation encourages lighting adaptation, yielding more coherent blending with the surrounding scene.

4 Implementation Details

We construct training tuples of the form $(\hat{V}, V, I, M, D_V, D_I)$, where $\hat{V}$ is the target video containing the instance, V is the paired background video, (I, M) are the segmented instance clip and its mask, and (D_V, D_I) are depth maps computed for background and instance signals.

Data sources. We use real-world videos from DAVIS [87], VPData [8], MOSE [20, 21], and DAVIS [10, 58]. We filter the segmentation results to favor clips with clean and logically removable instances and discard cases with ambiguous or non-detachable masks. After filtering, we obtain 16,642 video clips with corresponding foreground segmentation masks, each with at least 49 frames.

Background video generation. To construct paired videos $\{\hat{V}, V\}$, we train a side-effect-aware video instance removal model on ROSE [54]. Given a target video $\hat{V}$ and an instance mask M, we extract the masked instance I from $\hat{V}$ and apply the instance removal model to obtain the paired background video V. This procedure produces training pairs that share the same scene dynamics while differing in the presence of the target instance, which matches the insertion objective.

Geometry- and Appearance-Robust Training. We compute D_V by running Depth Anything V3 [46] on the background video V. We compute D_I by running it on the target video $\hat{V}$ and then cropping the instance region using M. To support the amodal instance augmentation described in Sec. 3.4, we construct a paired dataset of over 70K occluded and pseudo-amodal instance clips using a fine-tuned image editing model. Complete details of the data synthesis pipeline and curation process are deferred to the Appendix.

Training Strategy. We optimize PISCO using a progressive, multi-stage training strategy that first warms up the VACE context adapter before jointly fine-tuning it with the Wan diffusion backbone. Full training hyperparameters, stage definitions, and hardware details are provided in the Appendix.

5 Experiments

Evaluation data. We introduce `PISCO-Bench`, a curated benchmark of 100 diverse, real-world videos derived from the BURST dataset [6]. To ensure reliable evaluation, we manually refined the instance segmentation masks and applied the side-effect-aware removal model ROSE [54] to erase the instances, yielding paired background videos. Hereafter, we refer to these processed backgrounds as "clean videos" and the original footage containing the instances as "target videos".

Evaluation protocol. We report results for both `PISCO-1.3B` and `PISCO-14B` under two sparse-control settings: (i) **First-frame control**, where the segmented instance image and mask are provided only at the first frame; (ii) **First & last-frame control**, where the segmented instance image and mask are provided at both the first and last frames. For all experiments, we perform inference using 50 diffusion denoising steps.

Baselines. Since no existing open-source models support identical sparse keyframe conditioning, we adapt representative baselines from three categories: (1) Agentic pipelines (Image Editing + I2V Generation), (2) Video Inpainting models (CoCoCo, VideoPainter), and (3) Reference-guided V2V editing models (VACE, UniVideo). For methods requiring text prompts, we utilize a Vision-Language Model to generate detailed instance descriptions. Comprehensive details on baseline adaptations and inference settings are provided in the Appendix.

As outlined in Table 1, while several baselines demand labor-intensive dense per-frame masks, `PISCO` achieves precise control using only sparse keyframes (first or first & last). This contrast underscores our core design philosophy: minimizing user effort without compromising output quality. To ensure fair comparison across diverse architectures, all experiments are standardized to 49 frames at an 832×480 resolution.

Table 1: **Input conditions for all compared methods.** We indicate whether each approach uses the clean video, reference instance frames, instance masks, or optional text prompts. Several baselines require dense per-frame masks, whereas `PISCO` operates with sparse keyframe masks only.

Method		Instance		Instance Mask			
	Video	First	Last	First	Last	All	Text
Image Editing + I2V Generation							
First		✓		✓			✓
First&Last		✓	✓	✓	✓		✓
Video Inpainting							
CoCoCo	✓			✓	✓	✓	✓
VideoPainter	✓			✓	✓	✓	✓
Video-to-Video Editing							
VACE (14B)	✓	✓		✓	✓	✓	✓
UniVideo	✓	✓					✓
`PISCO` 1.3B (Ours)							
First	✓	✓		✓			
First&Last	✓	✓	✓	✓	✓		
`PISCO` 14B (Ours)							
First	✓	✓		✓			
First&Last	✓	✓	✓	✓	✓		

Table 3: **VBench Quantitative Comparison.** We compare PISCO against recent baselines. To ensure precise evaluation for instance insertion, *Background* and *Subject Consistency* are computed using masked regions to isolate foreground/background signals. Blue and red indicate the best and second-best results among standard settings. Gray rows indicate the Five-Frame setting.

Method	Background Consistency	Subject Consistency	Aesthetic Quality	Imaging Quality	Motion Smoothness	Overall Consistency	Temporal Flickering	Average
Image Editing + I2V Generation								
First Only	91.43	83.86	48.83	59.01	97.89	15.11	95.89	70.29
First + Last	92.28	85.09	49.47	59.71	98.18	15.58	96.26	70.94
Video Inpainting								
CoCoCo	94.63	89.55	47.81	55.76	98.67	14.61	97.69	71.25
VideoPainter	94.51	89.14	47.82	57.68	98.97	13.85	97.77	71.39
Video-to-Video Editing								
VACE_{14B}	94.21	90.29	48.69	60.95	98.90	14.87	97.56	72.21
UniVideo	94.04	89.88	49.41	60.84	98.80	15.51	97.01	72.21
PISCO $_{1.3B}$ (Ours)								
First Only	93.72	87.16	47.70	60.67	98.85	14.92	97.54	71.51
First + Last	94.07	91.33	49.48	61.18	98.86	15.58	97.51	72.57
Five Frames	94.23	91.45	50.01	62.09	98.86	15.61	97.67	72.85
PISCO $_{14B}$ (Ours)								
First Only	93.84	87.26	48.24	60.80	98.79	15.14	97.26	71.62
First + Last	94.20	91.57	50.08	62.00	98.79	15.64	97.21	72.78
Five Frames	94.42	91.98	51.45	62.87	98.89	15.82	97.34	73.25

5.1 Quantitative Experiments

Reference-based Video Quality Assessment. Using ground-truth pairs $(V, \hat{V})$ from PISCO-Bench, we evaluate generated videos V' across four metrics: FVD [61], LPIPS [81], PSNR [34], and SSIM [70]. To ensure a comprehensive analysis, we compute these at two levels: (1) **Whole-video assessment** calculates metrics between V' and $\hat{V}$ to measure global consistency and physical effects like shadows. (2) **Foreground assessment** computes Metric$(V' \odot M, \hat{V} \odot M)$ to isolate the inserted object, verifying its adherence to reference conditions and the intended trajectory.

Table 2 shows that PISCO consistently outperforms all baselines. The *Image Editing + I2V* pipeline suffers from severe background hallucination (low PSNR/SSIM), while *Video Inpainting* models preserve backgrounds but lack temporal coherence (high FVD). Although VACE is the strongest baseline, PISCO-14B (First & Last) significantly surpasses it, reducing whole-video FVD from 371 to 204. In foreground assessment, PISCO-14B achieves an FVD of 138 and LPIPS of 0.022, demonstrating superior instance fidelity. Furthermore, the clear improvement from "First Only" to "First & Last" confirms that endpoint constraints effectively stabilize complex instance dynamics.

Table 2: **Quantitative Results (FVD, LPIPS, PSNR, SSIM).** Comparisons on Whole Video and Foreground Region. Blue and red indicate best and second-best among standard settings. Gray rows indicate the Five-Frame reference setting (not included in ranking).

Method	Whole Video				Foreground			
	FVD ↓	LPIPS ↓	PSNR ↑	SSIM ↑	FVD ↓	LPIPS ↓	PSNR ↑	SSIM ↑
Image Editing + I2V Generation								
First Only	826	0.451	15.47	0.55	297	0.030	30.24	0.97
First + Last	624	0.392	16.44	0.56	250	0.030	30.38	0.98
Video Inpainting								
CoCoCo	590	0.191	23.62	0.80	398	0.031	30.26	0.97
VideoPainter	524	0.154	23.11	0.78	384	0.035	29.27	0.97
Video-to-Video Editing								
VACE_{14B}	371	0.103	25.55	0.88	273	0.028	30.55	0.98
UniVideo	485	0.211	19.22	0.61	310	0.031	29.21	0.97
PISCO $_{1.3B}$ (Ours)								
First Only	398	0.121	25.35	0.87	243	0.029	30.51	0.98
First + Last	269	0.103	27.01	0.88	171	0.024	32.99	0.98
Five Frames	172	0.089	28.53	0.89	104	0.018	35.07	0.98
PISCO $_{14B}$ (Ours)								
First Only	337	0.116	24.81	0.88	222	0.029	30.61	0.97
First + Last	204	0.097	26.58	0.89	138	0.022	33.58	0.98
Five Frames	136	0.084	28.01	0.90	75	0.015	35.94	0.98

We also evaluate a "Five Frames" setting (first, last, and three random intermediate frames). As a result, PISCO leverages additional sparse signals to monotonically refine quality. For instance, PISCO-14B improves whole-video FVD to 136 and Foreground LPIPS to 0.015, highlighting its unique scalable controllability.

Reference-free Video Quality Assessment. For reference-free evaluation, we employ VBench [33]. Since standard VBench metrics compute feature similarity over entire frames—conflating foreground and background—we adapt it using instance masks. This isolates region-specific CLIP/DINO features to compute *Subject Consistency* and *Background Consistency* independently. Other metrics follow the official implementation.

As shown in Table 3, PISCO excels across perceptual dimensions. For Subject Consistency, PISCO-14B (First & Last) scores 91.57, outperforming VACE (90.29) and inpainting models (∼89), demonstrating its ability to faithfully maintain instance identity. PISCO also achieves the best Imaging Quality, Aesthetic Quality, Overall Consistency, and Average score among standard settings. While inpainting methods inherently score high in Background Consistency via exact pixel-copying, PISCO remains highly competitive (94.20) without sacrificing dynamic foreground control. Consistent with reference-based results, the "Five Frames" setting further elevates performance (e.g., 91.98 Subject Consistency and 73.25 Average), reinforcing PISCO's capacity to leverage additional visual prompts.

Human Preference Study. We further conduct an arena-style user study with 16 users and 265 pairwise annotations. We fit a Bradley–Terry model [9] and report the Chatbot-Arena-style score [19], using the following formula: $1000 + 400 \log_{10}(s)$, where s is the normalized Bradley–Terry strength. Ties count as half wins, and 95% confidence intervals are estimated with 300 bootstrap rounds. As shown in Table 4, users consistently prefer PISCO variants, with PISCO-Full ranked first.

Table 4: **Human Preference Study.** Parentheses denote the 95% confidence interval.

Method	Arena Score
PISCO-Full	1161.2 (1093.6–1192.3)
PISCO-Three	1101.0 (1020.9–1152.0)
PISCO-First	1010.8 (872.2–1096.5)
UniVideo	971.1 (848.8–1017.6)
VideoPainter	830.7 (687.1–901.7)
CoCoCo	735.0 (602.1–801.8)
VACE_{14B}	698.2 (535.9–781.5)



Figure 7: **Qualitative Comparison**. Visual results of instance insertion in two dynamic scenes (A) and (B). Existing methods struggle with background preservation (Agentic), instance identity (Inpainting), or scale consistency (UniVideo/VACE). Our method, PISCO, achieves superior visual fidelity and spatiotemporal alignment, particularly when utilizing sparse "First & Last" frame control.

5.2 Qualitative Experiments

Figure 7 highlights severe baseline limitations: agentic pipelines regenerate sequences and destroy background consistency; inpainting models hallucinate blurry objects due to missing reference inputs; and V2V models (UniVideo/VACE) struggle with long-term spatial control, often distorting perspective (e.g., the oversized car) or losing the target entirely. In comparison, PISCO demonstrates superior spatiotemporal controllability. While "First Only" control captures appearance well (despite minor trajectory drift), providing sparse "First & Last" frames effectively anchors the generation, ensuring precise alignment and physically plausible dynamics throughout the entire sequence.

5.3 Ablation Studies

Protocol. Ablations are conducted with the same backbone, training data, and sparse first&last-frame inference setting. To focus on instance controllability, we report foreground-region FVD, LPIPS,

PSNR, and SSIM. We additionally conduct pairwise preference studies for each ablation group, using the same Bradley–Terry fitting and score mapping as in Table 4.

Sparse-conditioning mechanism. We isolate DPTM and compare it with a *Temporal Positional Indexing* baseline inspired by arbitrary-frame conditioning [13, 47]. This baseline uses RoPE-based temporal indexing to anchor available condition tokens, with the architecture and training setting unchanged. As shown in Table 5, temporal indexing helps, but DPTM performs best by combining pixel-space completion, token-space masking, and an explicit availability channel.

Table 5: **Sparse-conditioning mechanism.**

Method	FVD↓	LPIPS↓	PSNR↑	SSIM↑	Arena↑
Pixel mask	638.5	0.066	25.84	0.93	742.6
Temp. index	218.6	0.030	30.94	0.97	997.8
w/o pix. comp.	571.3	0.058	26.57	0.94	801.4
w/o token mask	164.8	0.025	32.47	0.98	1088.9
w/o avail.	276.4	0.038	29.35	0.96	904.7
DPTM	138.0	0.022	33.58	0.98	1161.2

Depth, amodal, and relighting ablations. Finally, we quantify the three auxiliary components in our training pipeline: depth conditioning, amodal instance augmentation, and relighting augmentation. As shown in Table 6, depth improves geometric alignment and occlusion reasoning, amodal augmentation improves robustness to occluded training instances, and relighting improves scene-compatible appearance adaptation.

Table 6: **Depth, amodal, and relighting ablations.**

Method	FVD↓	LPIPS↓	PSNR↑	SSIM↑	Arena↑
RGB + mask only	187.6	0.029	31.28	0.97	998.5
+ depth map	161.9	0.025	32.44	0.98	1084.6
+ amodal aug	146.7	0.023	33.86	0.98	1123.8
+ relighting aug (Ours)	138.0	0.022	33.58	0.98	1161.2

6 Beyond Instance Insertion

While designed for precise insertion, PISCO's instance-level conditioning and temporal propagation machinery naturally generalizes to broader video editing and simulation tasks. As illustrated in Figure 8, PISCO enables diverse spatiotemporal manipulations: **background replacement** while strictly preserving foreground identity and motion; **instance repositioning and rescaling** with physically consistent shadows and occlusions; **speed manipulation** via temporal condition subsampling; and **dynamics simulation** for generating counterfactual trajectories to stress-test downstream perception systems. These extensions highlight PISCO as a versatile, general-purpose framework for instance-centric video manipulation.

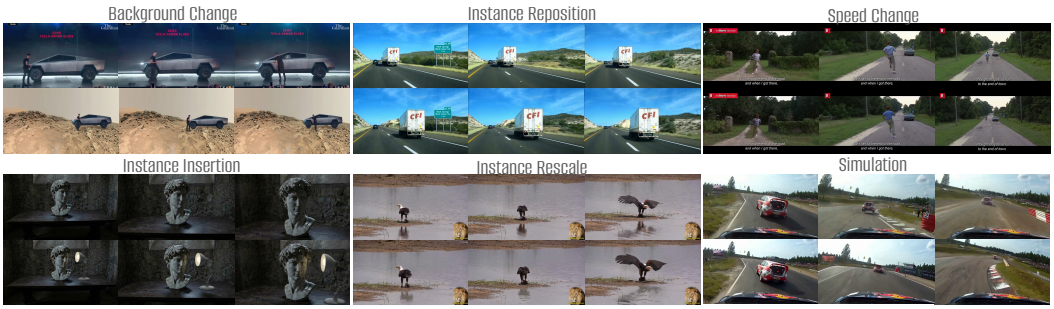

Figure 8: **Broader applications beyond instance insertion.** PISCO supports diverse instance-centric edits under spatiotemporal guidance, including background change, instance repositioning/rescaling, speed manipulation, and dynamics simulation.

7 Conclusion

We presented PISCO, a novel video diffusion framework for precise video instance insertion under sparse user control. By integrating Variable-Information Guidance (VIG) and Distribution-Preserving Temporal Masking (DPTM) with geometry-aware conditioning, PISCO successfully mitigates the distribution shifts and temporal instability inherent in sparse-control video generation. Extensive evaluations on the curated PISCO-Bench demonstrate that PISCO significantly outperforms existing inpainting, video editing, and agentic baselines, while offering monotonic quality improvements as additional sparse control frames are provided. Ultimately, PISCO establishes a highly controllable, scalable, and low-effort paradigm for professional-grade, AI-assisted filmmaking.

Acknowledgments and Disclosure of Funding

This work received no third-party funding. The authors declare no competing interests.

References

[1] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025. 3

[2] Anastasios Nikolas Angelopoulos, Wei-Lin Chiang, and Shishir G. Patil. Statistical extensions of the bradley-terry and elo models. https://blog.lmarena.ai/blog/2024/extended-arena/, 2024. 18

[3] Jiayang Ao, Yanbei Jiang, Qiuhong Ke, and Krista A Ehinger. Open-world amodal appearance completion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6490–6499, 2025. 5

[4] Jiayang Ao, Qiuhong Ke, and Krista A Ehinger. Image amodal completion: A survey. *Computer Vision and Image Understanding*, 229:103661, 2023. 5

[5] Arena Team. Arena's ranking method. https://arena.ai/blog/ranking-method/, 2025. 18

[6] Ali Athar, Jonathon Luiten, Paul Voigtlaender, Tarasha Khurana, Achal Dave, Bastian Leibe, and Deva Ramanan. Burst: A benchmark for unifying object recognition, segmentation and tracking in video. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1674–1683, 2023. 2, 6, 17

[7] Chen Bai, Zeman Shao, Guoxiang Zhang, Di Liang, Jie Yang, Zhuorui Zhang, Yujian Guo, Chengzhang Zhong, Yiqiao Qiu, Zhendong Wang, et al. Anything in any scene: Photorealistic video object insertion. *arXiv preprint arXiv:2401.17509*, 2024. 2, 16

[8] Yuxuan Bian, Zhaoyang Zhang, Xuan Ju, Mingdeng Cao, Liangbin Xie, Ying Shan, and Qiang Xu. Videopainter: Any-length video inpainting and editing with plug-and-play context control. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–12, 2025. 2, 6, 16, 17, 19

[9] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. 8, 18

[10] Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevis-Kokitsi Maninis, and Luc Van Gool. The 2019 davis challenge on vos: Unsupervised multi-object segmentation. *arXiv*, 2019. 6, 17

[11] Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Steven Hoi, Shijie Huang, Zhaohui Hou, Dengyang Jiang, Xin Jin, Liangchen Li, et al. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025. 2, 16

[12] Jiayin Cai, Changlin Li, Xin Tao, Chun Yuan, and Yu-Wing Tai. Devit: Deformed vision transformers in video inpainting. In *Proceedings of the 30th ACM international conference on multimedia*, pages 779–789, 2022. 16

[13] Minghong Cai, Qiulin Wang, Zongli Ye, Wenze Liu, Quande Liu, Weicai Ye, Xintao Wang, Pengfei Wan, Kun Gai, and Xiangyu Yue. Videocanvas: Unified video completion from arbitrary spatiotemporal patches via in-context conditioning. *arXiv preprint arXiv:2510.08555*, 2025. 3, 9

[14] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Free-form video inpainting with 3d gated convolution and temporal patchgan. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9066–9075, 2019. 16

[15] Jinshu Chen, Xinghui Li, Xu Bai, Tianxiang Ma, Pengze Zhang, Zhuowei Chen, Gen Li, Lijie Liu, Songtao Zhao, Bingchuan Li, and Qian He. Omniinsert: Mask-free video insertion of any reference via diffusion transformer models, 2025. 2, 16, 20

[16] Shoufa Chen, Chongjian Ge, Yuqi Zhang, Yida Zhang, Fengda Zhu, Hao Yang, Hongxiang Hao, Hui Wu, Zhichao Lai, Yifei Hu, et al. Goku: Flow based video generative foundation models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23516–23527, 2025. 3

[17] Yinan Chen, Jiangning Zhang, Teng Hu, Yuxiang Zeng, Zhucun Xue, Qingdong He, Chengjie Wang, Yong Liu, Xiaobin Hu, and Shuicheng Yan. Ivebench: Modern benchmark suite for instruction-guided video editing assessment. *arXiv preprint arXiv:2510.11647*, 2025. 19

[18] Yiyang Chen, Xuanhua He, Xiujun Ma, and Jack Ma. Contextflow: Training-free video object editing via adaptive context enrichment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 3129–3137, 2026. 2, 16

[19] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*, 2024. 8, 17, 18

[20] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, Philip HS Torr, and Song Bai. MOSE: A new dataset for video object segmentation in complex scenes. In *ICCV*, 2023. 6, 17

[21] Henghui Ding, Kaining Ying, Chang Liu, Shuting He, Xudong Jiang, Yu-Gang Jiang, Philip HS Torr, and Song Bai. MOSEv2: A more challenging dataset for video object segmentation in complex scenes. *arXiv preprint arXiv:2508.05630*, 2025. 6, 17

[22] Chen Gao, Ayush Saraf, Jia-Bin Huang, and Johannes Kopf. Flow-edge guided video completion. In *European Conference on Computer Vision*, pages 713–729. Springer, 2020. 16

[23] Xiangbo Gao, Sicong Jiang, Bangya Liu, Xinghao Chen, Minglai Yang, Siyuan Yang, Mingyang Wu, Jiongze Yu, Qi Zheng, Haozhi Wang, Jiayi Zhang, Jie Yang, Zihan Wang, Qing Yin, and Zhengzhong Tu. VEFX-Bench: A holistic benchmark for generic video editing and visual effects. *arXiv preprint arXiv:2604.16272*, 2026. 1

[24] Xiangbo Gao, Siyuan Yang, Ping He, Mingyang Wu, Yuheng Wu, Yushen Zuo, Jiongze Yu, Ryan Cui, Hongyuan Hua, Devin Ma, Xiao Jin, Yubo Yuan, Qing Yin, Jie Yang, and Zhengzhong Tu. Visko orbis 1.0: A live model for real-time interactive long video generation. *arXiv preprint arXiv:2607.26694*, 2026. 1

[25] Google Cloud. Nano banana pro: State-of-the-art image generation for enterprise. Google Cloud Blog, November 2025. 19

[26] Dylan Green, William Harvey, Saeid Naderiparizi, Matthew Niedoba, Yunpeng Liu, Xiaoxuan Liang, Jonathan Lavington, Ke Zhang, Vasileios Lioutas, Setareh Dabiri, et al. Semantically consistent video inpainting with conditional diffusion models. *arXiv preprint arXiv:2405.00251*, 2024. 16

[27] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023. 16

[28] Yuwei Guo, Ceyuan Yang, Anyi Rao, Chenlin Meng, Omer Bar-Tal, Shuangrui Ding, Maneesh Agrawala, Dahua Lin, and Bo Dai. Keyframe-guided creative video inpainting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13009–13020, 2025. 2, 16, 20

[29] Haoyang He, Jie Wang, Jiangning Zhang, Zhucun Xue, Xingyuan Bu, Qiangpeng Yang, Shilei Wen, and Lei Xie. Openve-3m: A large-scale high-quality dataset for instruction-guided video editing. *arXiv preprint arXiv:2512.07826*, 2025. 19

[30] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 19

[31] Yaosi Hu, Chong Luo, and Zhenzhong Chen. Make it move: controllable image-to-video generation with text descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18219–18228, 2022. 2, 16

[32] Yuan-Ting Hu, Heng Wang, Nicolas Ballas, Kristen Grauman, and Alexander G Schwing. Proposal-based video completion. In *European Conference on Computer Vision*, pages 38–54. Springer, 2020. 16

[33] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024. 8

[34] Bernd Jähne. *Digital image processing*. Springer, 2005. 7

[35] Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. *arXiv preprint arXiv:2503.07598*, 2025. 2, 3, 16, 17, 19

[36] Hoiyeong Jin, Hyojin Jang, Jeongho Kim, Junha Hyung, Kinam Kim, Dongjin Kim, Huijin Choi, Hyeonji Kim, and Jaegul Choo. Insertanywhere: Bridging 4d scene geometry and diffusion models for realistic video object insertion. *arXiv preprint arXiv:2512.17504*, 2025. 2, 16

[37] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024. 2, 16

[38] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep video inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5792–5801, 2019. 16

[39] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025. 16

[40] Minhyeok Lee, Suhwan Cho, Chajin Shin, Jungho Lee, Sunghun Yang, and Sangyoun Lee. Video diffusion models are strong video inpainter. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 4526–4533, 2025. 16

[41] Ang Li, Shanshan Zhao, Xingjun Ma, Mingming Gong, Jianzhong Qi, Rui Zhang, Dacheng Tao, and Ramamohanarao Kotagiri. Short-term and long-term context aggregation network for video inpainting. In *European Conference on Computer Vision*, pages 728–743. Springer, 2020. 16

[42] Xiaowen Li, Haolan Xue, Peiran Ren, and Liefeng Bo. Diffueraser: A diffusion model for video inpainting. *arXiv preprint arXiv:2501.10018*, 2025. 16

[43] Zhen Li, Cheng-Ze Lu, Jianhua Qin, Chun-Le Guo, and Ming-Ming Cheng. Towards an end-to-end framework for flow-guided video inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17562–17571, 2022. 16

[44] Zirui Li, Xinghao Chen, Lingyu Jiang, Dengzhe Hou, Fangzhou Lin, Kazunori Yamada, Xiangbo Gao, and Zhengzhong Tu. Physics-aware video instance removal benchmark. *arXiv preprint arXiv:2604.05898*, 2026. 1

[45] Ziye Li, Hao Luo, Xincheng Shuai, and Henghui Ding. Anyi2v: Animating any conditional image with motion control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17302–17311, 2025. 16

[46] 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. 6, 17

[47] Jiawei Liu, Junqiao Li, Jiangfan Deng, Gen Li, Siyu Zhou, Zetao Fang, Shanshan Lao, Zengde Deng, Jianing Zhu, Tingting Ma, et al. Dreamontage: Arbitrary frame-guided one-shot video generation. *arXiv preprint arXiv:2512.21252*, 2025. 3, 4, 9

[48] Jinlai Liu, Jian Han, Bin Yan, Hui Wu, Fengda Zhu, Xing Wang, Yi Jiang, Bingyue Peng, and Zehuan Yuan. Infinitystar: Unified spacetime autoregressive modeling for visual generation. *arXiv preprint arXiv:2511.04675*, 2025. 3

[49] Rui Liu, Hanming Deng, Yangyi Huang, Xiaoyu Shi, Lewei Lu, Wenxiu Sun, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Decoupled spatial-temporal transformer for video inpainting. *arXiv preprint arXiv:2104.06637*, 2021. 16

[50] Rui Liu, Hanming Deng, Yangyi Huang, Xiaoyu Shi, Lewei Lu, Wenxiu Sun, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fuseformer: Fusing fine-grained information in transformers for video inpainting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14040–14049, 2021. 16

[51] Shaoteng Liu, Tianyu Wang, Jui-Hsien Wang, Qing Liu, Zhifei Zhang, Joon-Young Lee, Yijun Li, Bei Yu, Zhe Lin, Soo Ye Kim, et al. Generative video propagation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17712–17722, 2025. 2, 16, 20

[52] Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025. 2, 16

[53] Ziling Liu, Jinyu Yang, Mingqi Gao, and Feng Zheng. Place anything into any video. *arXiv preprint arXiv:2402.14316*, 2024. 2, 16

[54] Chenxuan Miao, Yutong Feng, Jianshu Zeng, Zixiang Gao, Hantang Liu, Yunfeng Yan, Donglian Qi, Xi Chen, Bin Wang, and Hengshuang Zhao. Rose: Remove objects with side effects in videos. *arXiv preprint arXiv:2508.18633*, 2025. 6, 17

[55] Justine Moore. Why 2023 was ai video's breakout year, and what to expect in 2024, jan 2024. Accessed: 2024-05-20. 1

[56] Koichi Namekata, Sherwin Bahmani, Ziyi Wu, Yash Kant, Igor Gilitschenski, and David B Lindell. Sg-i2v: Self-guided trajectory control in image-to-video generation. *arXiv preprint arXiv:2411.04989*, 2024. 16

[57] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. 1

[58] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 6, 17

[59] Weize Quan, Jiaxi Chen, Yanli Liu, Dong-Ming Yan, and Peter Wonka. Deep learning-based image and video inpainting: A survey. *International Journal of Computer Vision*, 132(7):2367–2400, 2024. 2, 16

[60] Shuwei Shi, Biao Gong, Xi Chen, Dandan Zheng, Shuai Tan, Zizheng Yang, Yuyuan Li, Jingwen He, Kecheng Zheng, Jingdong Chen, et al. Motionstone: Decoupled motion intensity modulation with diffusion transformer for image-to-video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22864–22874, 2025. 16

[61] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3626–3636, 2022. 7

[62] Yuanpeng Tu, Hao Luo, Xi Chen, Sihui Ji, Xiang Bai, and Hengshuang Zhao. Videoanydoor: High-fidelity video object insertion with precise motion control. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–11, 2025. 2, 16, 20

[63] 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. 1, 3, 16, 17, 19

[64] Zhen Wan, Chenyang Qi, Zhiheng Liu, Tao Gui, and Yue Ma. Unipaint: Unified space-time video inpainting via mixture-of-experts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1861–1871, 2025. 16

[65] Chuan Wang, Haibin Huang, Xiaoguang Han, and Jue Wang. Video inpainting by jointly learning temporal structure and spatial details. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5232–5239, 2019. 16

[66] Junke Wang, Yi Jiang, Zehuan Yuan, Bingyue Peng, Zuxuan Wu, and Yu-Gang Jiang. Omnitokenizer: A joint image-video tokenizer for visual generation. *Advances in Neural Information Processing Systems*, 37:28281–28295, 2024. 3

[67] Wenhao Wang and Yi Yang. Tip-i2v: A million-scale real text and image prompt dataset for image-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14898–14908, 2025. 16

[68] Xiaojuan Wang, Boyang Zhou, Brian Curless, Ira Kemelmacher-Shlizerman, Aleksander Holynski, and Steven M Seitz. Generative inbetweening: Adapting image-to-video models for keyframe interpolation. *arXiv preprint arXiv:2408.15239*, 2024. 16

[69] Yukun Wang, Longguang Wang, Zhiyuan Ma, Qibin Hu, Kai Xu, and Yulan Guo. Videodirector: Precise video editing via text-to-video models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2589–2598, 2025. 2, 16

[70] 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. 7

[71] Cong Wei, Quande Liu, Zixuan Ye, Qiulin Wang, Xintao Wang, Pengfei Wan, Kun Gai, and Wenhu Chen. Univideo: Unified understanding, generation, and editing for videos. *arXiv preprint arXiv:2510.08377*, 2025. 2, 16, 19

[72] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025. 2, 16, 17, 19

[73] Yuheng Wu, Xiangbo Gao, Tianhao Chen, Xinghao Chen, Qing Yin, Zhengzhong Tu, and Dongman Lee. Delta forcing: Trust region steering for interactive autoregressive video generation. *arXiv preprint arXiv:2605.14382*, 2026. 1

[74] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Deep flow-guided video inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3723–3732, 2019. 16

[75] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM computing surveys*, 56(4):1–39, 2023. 2, 16

[76] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 3

[77] Zuhao Yang, Jiahui Zhang, Yingchen Yu, Shijian Lu, and Song Bai. Versatile transition generation with image-to-video diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16981–16990, 2025. 16

[78] 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 Computer Vision and Pattern Recognition Conference*, pages 18198–18208, 2025. 16

[79] Jiongze Yu, Xiangbo Gao, Pooja Verlani, Akshay Gadde, Yilin Wang, Balu Adsumilli, and Zhengzhong Tu. SparkVSR: Interactive video super-resolution via sparse keyframe propagation. *arXiv preprint arXiv:2603.16864*, 2026. 1

[80] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Scaling in-the-wild training for diffusion-based illumination harmonization and editing by imposing consistent light transport. In *The Thirteenth International Conference on Learning Representations*, 2025. 5, 17

[81] 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*, pages 586–595, 2018. 7

[82] Zhixing Zhang, Bichen Wu, Xiaoyan Wang, Yaqiao Luo, Luxin Zhang, Yinan Zhao, Peter Vajda, Dimitris Metaxas, and Licheng Yu. Avid: Any-length video inpainting with diffusion model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7162–7172, 2024. 16

[83] Zhongwei Zhang, Fuchen Long, Zhaofan Qiu, Yingwei Pan, Wu Liu, Ting Yao, and Tao Mei. Motionpro: A precise motion controller for image-to-video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27957–27967, 2025. 16

[84] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024. 3

[85] Shangchen Zhou, Chongyi Li, Kelvin CK Chan, and Chen Change Loy. Propainter: Improving propagation and transformer for video inpainting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10477–10486, 2023. 16

[86] Bojia Zi, Shihao Zhao, Xianbiao Qi, Jianan Wang, Yukai Shi, Qianyu Chen, Bin Liang, Rong Xiao, Kam-Fai Wong, and Lei Zhang. Cococo: Improving text-guided video inpainting for better consistency, controllability and compatibility. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 11067–11076, 2025. 2, 16, 19

[87] Xueyan Zou, Linjie Yang, Ding Liu, and Yong Jae Lee. Progressive temporal feature alignment network for video inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16448–16457, 2021. 6, 16, 17

Supplementary Material

A Related Work

Video Inpainting and Keyframe-Guided Inpainting. Video inpainting [59] is designed to recover masked regions to achieve object removal, insertion, or restoration. Early approaches relied on 3D-CNNs [65, 14, 32] and optical flow guidance [74, 38, 41, 22, 87, 43] for spatio-temporal coherence, while Transformer-based models [50, 49, 12, 85] later improved long-range correlation capture. Recently, diffusion models [75] and video foundation models like Wan [63] and Flux [39] have been extensively applied to inpainting [27, 82, 26, 86, 8, 42, 40, 64]. Closely related to our sparse-control setting, VideoRepainter [28] performs keyframe-guided creative video inpainting by editing a keyframe and propagating the change through the video. However, inpainting-based formulations still operate on masked-region completion and typically require dense masks or keyframe edits rather than explicit sparse instance RGB/depth controls.

Reference-Guided Video Object Insertion and Editing. Reference-guided video editing leverages multimodal references, such as images or text, to guide synthesis. Large-scale frameworks like UniVideo [71] and VACE [35] unify generation and editing tasks, while VideoDirector [69] and ContextFlow [18] utilize text-to-video models for scene modification. Recent task-specific methods further study high-fidelity object insertion or replacement, including VideoAnydoor [62], which uses explicit motion controls such as boxes and keypoints, and OmniInsert [15], which targets mask-free insertion and subject-scene harmonization. However, these methods struggle to jointly control instance appearance and spatial-temporal placement. In contrast, PISCO uses sparse keyframe conditions to anchor both appearance and location, and propagates the instance through the video.

Image-to-Video Generation and Edit Propagation. The "generate-then-animate" paradigm involves editing a keyframe using tools like Step1x-edit [52], Z-Image [11], or Qwen-Image [72], and then animating it via I2V models [31, 56, 68, 37, 45] or transition generation [77, 67]. Specialized architectures such as MotionStone [60], MotionPro [83], and Through-The-Mask [78] attempt to decouple motion from appearance for better controllability. GenProp [51] further formulates video editing as generative propagation, where a first-frame edit is propagated through an I2V model for tasks such as insertion, removal, and replacement. However, such propagation pipelines are typically anchored on one or a few edited full frames and can suffer from background hallucination or temporal drift because the model regenerates full-frame motion rather than explicitly preserving the original video context. PISCO instead injects sparse instance-side conditions into a conditional video model while using the clean video as persistent context.

4D Reconstruction and Manipulation. To ensure geometric consistency, some works lift 2D video into 3D or 4D representations. Methods like "Place Anything into Any Video" [53] and "Anything in Any Scene" [7] use depth and flow for physical placement, while InsertAnywhere [36] bridges 4D scene geometry with diffusion models to handle complex occlusions. However, these geometry-heavy approaches are limited by data scarcity and poor generalization across diverse scenes. Furthermore, they struggle with dynamic object insertion, as they often require pre-existing dynamic 4D models. Our method bypasses heavy reconstruction requirements, offering a data-efficient 2D generative solution that supports natural dynamic object insertion and interaction.

B Limitations and Responsible Research

Limitations. PISCO uses instance masks and depth maps in addition to the RGB input video, which introduces extra conditioning signals compared with purely text- or image-guided editors. In practice, this requirement is substantially eased by our dedicated editing interface: masks and depth can be obtained automatically with off-the-shelf segmentation and depth estimators such as SAM and Depth Anything, and our user study indicates that these automatically generated conditions are sufficient for high-quality instance insertion. For fair comparison with existing baselines, our main benchmark uses 49-frame videos at 832×480 resolution, matching the operating range of several compared models; after post-training, however, PISCO can generate 120-frame videos at 720p resolution, as shown in the supplementary results. The main remaining challenge is handling extremely fast motion, severe occlusion, and other highly complex dynamics, where insertion quality can degrade. We view this as a difficult open problem for future sparse-control video editing systems.

Broader impact and safeguards. Precise video instance insertion can benefit film production, simulation, data augmentation, and accessibility-oriented content creation, but it also carries dual-use risks such as deceptive media manipulation or unauthorized insertion of people or objects into videos. We mitigate these risks by positioning the method for research and professional editing workflows, evaluating on curated non-sensitive benchmark clips, and planning any release with clear usage terms that prohibit impersonation, harassment, non-consensual manipulation, and misleading deployment. We do not release private user data or personally identifying participant information.

Assets and licenses. Our benchmark is derived from publicly available research datasets and models cited in the main paper and appendix. We use these assets for research evaluation under their respective terms, and we document the asset role, source, license or access terms, and redistribution policy in Table 7. For any released benchmark or code, we will release only the derived annotations and metadata that we are permitted to redistribute, and users will be directed to obtain restricted source videos or model weights from the original providers.

Table 7: **Asset provenance and redistribution plan.** Licenses and terms were checked from the public project or model pages in May 2026. "No raw redistribution" means that our release will contain derived annotations, prompts, masks, or evaluation metadata when permitted, while source assets must be obtained from the original provider.

Asset	Role	Source	License / terms	Redistribution
BURST [6]	`PISCO-Bench` source videos and masks	https://github.com/Ali2500/BURST-benchmark	Original dataset terms	No raw redistribution; release derived benchmark metadata where permitted
Progressive Video Inpainting [87]	Training video source	Cited project release	Original research terms	No raw redistribution
VPData [8]	Training video/mask source	https://huggingface.co/datasets/TencentARC/VPData	Provider dataset terms; raw videos retain original source terms	No raw redistribution
MOSE/MOSEv2 [20, 21]	Training masks/videos	https://mose.video/	CC BY-NC-SA 4.0, non-commercial research	No raw redistribution beyond allowed metadata
DAVIS [10, 58]	Training masks/videos	https://davischallenge.org/	CC BY 4.0 for challenge annotations	No raw redistribution beyond allowed metadata
ROSE [54]	Side-effect-aware removal model	https://github.com/Kunbyte-AI/ROSE	Apache 2.0 for public code/model release	Release only generated clean-video metadata where source terms allow
Wan/VACE [63, 35]	Backbone and baseline	Public model/project pages	Apache 2.0 for Wan/VACE releases used here	No model redistribution; cite original providers
Depth Anything V3 [46]	Depth preprocessing	Public code/model pages	Apache 2.0 code; model weights under provider terms	No model redistribution
IC-Light [80], Qwen-Image-Edit [72]	Relighting and image-edit augmentation	Public model/project pages	Original model licenses/terms, including Apache 2.0 for Qwen-Image-Edit	No model redistribution; derived training data released only if permitted

B.1 Human Preference Study Details

Interface, consent, and risk. We conduct a blind pairwise preference study following the evaluation spirit of Chatbot Arena / LM Arena [19]. For each comparison, a participant is shown the original video, the inserted instance, and two anonymized generated videos from the same input condition, with the left/right order randomized. Participants are instructed to select the video with better overall insertion quality, considering instance identity preservation, spatial placement, temporal coherence, background preservation, and scene-consistent realism. If neither result is clearly better, participants may select a tie. The interface provides three choices: *Choose Video A*, *Choose Video B*, and *About the same*. Participants were informed that the task involved viewing anonymized generated videos for research evaluation, that no personally identifying information would be collected, and that they could stop participation at any time. We collect 265 pairwise annotations from 16 users. The study is minimal risk because participants only view non-sensitive generated videos and provide pairwise quality judgments; no biometric, demographic, or contact information is collected. Participants were not recruited through a paid crowdsourcing marketplace.

Bradley–Terry model. We aggregate pairwise preferences with a Bradley–Terry (BT) model [9], which is also the statistical basis of Arena-style human-preference ranking [19, 2]. Let $\mathcal{M}$ be the set of evaluated methods and let each method $m \in \mathcal{M}$ have a latent strength $s_m > 0$, equivalently $\theta_m = \log s_m$. For a comparison between methods i and j, the BT win probability is

$$p_{ij} = \Pr(i \succ j) = \frac{s_i}{s_i + s_j} = \sigma(\theta_i - \theta_j), \tag{3}$$

where $\sigma(\cdot)$ denotes the sigmoid function. For each annotation $b = (i_b, j_b, y_b)$, we set $y_b = 1$ if i_b is preferred, $y_b = 0$ if j_b is preferred, and $y_b = 0.5$ for a tie. The BT parameters are estimated by maximizing the log-likelihood

$$\hat{\theta} = \arg\max_{\theta} \sum_{b=1}^{B} \left[y_b \log p_{i_b j_b} + (1 - y_b) \log(1 - p_{i_b j_b}) \right], \tag{4}$$

which makes ties equivalent to assigning a half win and a half loss.

Arena-style score. Following the Arena convention of reporting BT-derived coefficients on an Elo-like scale [19, 2, 5], we first center the fitted log-strengths by subtracting their mean, $\tilde{\theta}_m = \hat{\theta}_m - \frac{1}{|\mathcal{M}|} \sum_{k \in \mathcal{M}} \hat{\theta}_k$, and define the normalized BT strength $\tilde{s}_m = \exp(\tilde{\theta}_m)$. The reported Arena-style score is

$$\mathrm{ArenaScore}(m) = 1000 + 400 \log_{10}(\tilde{s}_m). \tag{5}$$

This transformation is monotonic in the BT strength and only changes the display scale; it does not affect pairwise win-probability estimates or the method ranking.

Confidence intervals. To quantify uncertainty, we bootstrap the collected pairwise annotations 300 times. In each bootstrap round, we sample $B = 265$ annotations with replacement, refit the BT model, recompute Arena-style scores, and take the empirical 2.5th and 97.5th percentiles as the 95% confidence interval. This follows the Arena practice of reporting uncertainty together with model scores [5].

C Detailed Implementation

C.1 Progressive Training Schedule

Our models are built upon two pretrained backbones: `Wan2.1-VACE-1.3B` and `Wan2.2-VACE-Fun-A14B`, denoted as `PISCO-1.3B` and `PISCO-14B`, respectively. To ensure stable adaptation to our multi-condition setting while preserving pretrained generative priors, we adopt a staged training strategy with progressively increased model flexibility.

All experiments are trained using AdamW on NVIDIA H100 GPUs. As detailed in Table 8, stages I through IV are conducted at a spatial resolution of 832×480 with a temporal length of 49 frames to stabilize optimization and reduce computational cost.

- **Stage I: Adapter input warm-up.** We replace the input projection of the VACE context adapter to support multi-channel conditioning and train only this newly introduced layer while freezing all other modules. This stabilizes condition injection before finetuning deeper components.
- **Stage II: Adapter finetuning.** We finetune the full VACE context adapter while keeping the diffusion backbone frozen, allowing the adapter to better align pretrained representations with task-specific conditioning.
- **Stage III: Joint finetuning.** We jointly finetune the context adapter and the diffusion backbone to improve the coupling between conditioning signals and generation dynamics.
- **Stage IV: Augmented training.** We introduce occlusion-aware completion and relighting augmentations and continue joint training using parameter-efficient LoRA finetuning to enhance robustness under challenging visibility and appearance variations.
- **Stage V: Resolution and temporal extension.** We further extend the model to generate 120-frame videos at a 1280×720 resolution.

Because the Wan2.2 backbone adopts a dual-denoiser architecture (with separate high-noise and low-noise denoising stages), the two denoisers for `PISCO-14B` are trained independently using this exact same schedule and hyperparameters.

Table 8: **Detailed Training Hyperparameters.** Configuration for each progressive training stage.

Stage	Trainable Modules	Learning Rate	Steps	Batch Size
I	Adapter input layer	1e−4	1k	16
II	Full adapter	2e−5	5k	16
III	Adapter + DiT Backbone	1e−5	2k	32
IV	Adapter + DiT (LoRA, $r = 64$)	1e−4	2k	16
V	Adapter + DiT Backbone	2e−5	1k	16

C.2 Amodal Completion Data Construction

To support the amodal instance augmentation used during Stage IV, we train a dedicated completion model $\mathcal{F}$ that recovers the full amodal instance appearance $\tilde{I} = \mathcal{F}(I)$ from an occluded modal instance segment I.

To construct paired training data, we first crop instance regions from our collected dataset, resizing them to 256×256 to yield over 70K candidates. From this pool, we manually select 5,122 fully visible instances to serve as the amodal ground truth. We then synthesize corresponding modal inputs by randomly overlaying 1 to 5 additional instance cutouts onto each amodal sample, yielding paired (occluded, amodal) training examples.

We employ `Qwen-Image-Edit` [72] as the base generator, fine-tuning it with LoRA [30] (rank 64) for 5,000 steps with a batch size of 8. We apply $\mathcal{F}$ frame-wise to generate pseudo-amodal instance clips $\tilde{I}$. The corresponding amodal mask $\tilde{M}$ is derived by thresholding the completed frames, while the amodal instance depth $\tilde{D}_I$ is synthesized via nearest-neighbor interpolation from the available depth values.

D Video Instance Insertion Baselines

Since there are no directly comparable open-sourced models that support our exact sparse conditioning interface, we compare against representative approaches capable of achieving similar functionality via alternative pipelines.

(1) Agentic Pipeline (Image Editing + I2V Generation). This pipeline first edits the keyframe(s) using the reference instance and mask, then generates the full video conditioned on these edited frames with an Image-to-Video (I2V) model. This is a commonly adopted recipe in recent editing data generation workflows [29, 17]. We use `Nano-banana-Pro` [25] for image editing, taking the unedited frame(s) along with the segmented reference instance and mask as input. We then use `Wan2.2-Fun-A14B-InP` [63] for I2V generation under both first-frame and first&last-frame control.

(2) Video Inpainting Models. Video inpainting methods strictly require dense masks over all frames, along with a text prompt describing the inserted object and target scene. We use `Qwen3-VL-32B-Instruct` to generate a detailed textual description of the segmented reference instance image to serve as this prompt. We adopt CoCoCo [86] and VideoPainter [8] as our strong inpainting baselines. Both take the text prompt and the masked video as input.

(3) Reference-Guided Video-to-Video Editing. This category edits the entire video end-to-end using multiple conditions (reference image, text prompt, and masks). We evaluate VACE [35] and UniVideo [71]. VACE takes the unedited video, segmented reference image, text prompt, and the full mask video as input. Because UniVideo does not support mask conditioning, it relies solely on the unedited video, segmented reference image, and text prompt.

E Practical Workflow and Comparison Scope

Sparse motion control. A single instance keyframe specifies appearance, pose, and placement but does not uniquely determine future motion; `PISCO` samples a plausible continuation conditioned on the background video. First/last anchors constrain the endpoints and overall displacement, while intermediate anchors further constrain the path and timing.

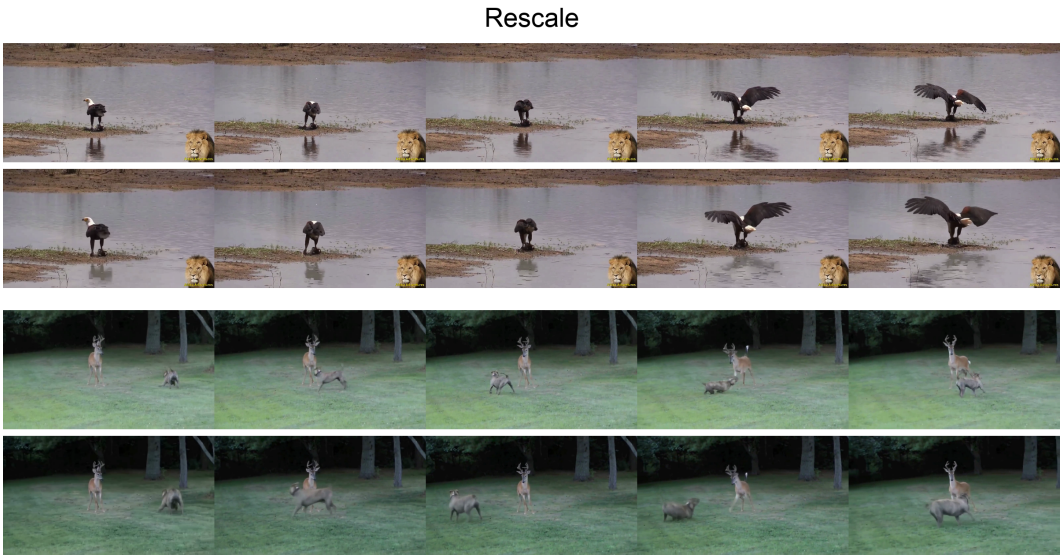

Figure 9: **Instance rescaling.** Adjusting the scale condition allows the framework to accurately resize dynamic target instances while preserving their original identities and environmental interactions.

Public editing workflow. Our development tools combine image-based asset creation, automatic mask/depth estimation, and drag/resize placement of sparse keyframes. The reported setup time of under one minute refers to active user operation and excludes model generation time. The tools, code, model weights, and released training data are publicly linked at https://xiangbogaobarry.github.io/PISCO/.

Comparison scope. Comparative performance claims refer only to the evaluated baselines under the stated input protocols. VideoAnydoor [62] uses dense motion controls, GenProp [51] and VideoRepainter [28] propagate an edited keyframe, and OmniInsert [15] uses a mask-free reference interface. At rebuttal time, their public releases lacked complete runnable implementations or checkpoints for a controlled comparison. We therefore do not claim superiority over these unevaluated methods.

F Broader Applications of PISCO

To demonstrate the versatility of the framework in handling complex visual dynamics, we showcase its performance across a broad spectrum of downstream video editing tasks. Beyond standard precise **Instance Insertion** (Figure 10), the model's robust temporal and spatial control mechanisms naturally extend to **Instance Repositioning** (Figure 11) and **Instance Rescaling** (Figure 9), where target objects are manipulated in size and location while maintaining physically plausible interactions like reflections and dynamic shadows. Furthermore, the framework proves highly effective for **Dynamics Simulation** (Figure 12), enabling the generation of counterfactual trajectories for autonomous driving scenarios, as well as **Creative Editing** (Figure 13), which allows for high-fidelity modifications in stylized and cinematic footage. Together, these diverse applications highlight PISCO's capability to serve as a comprehensive tool for both rigorous physical simulation and artistic video production.



Figure 10: **Qualitative results for precise instance insertion.** The framework seamlessly integrates various target instances—ranging from static objects to dynamic wildlife—into original clean background videos while maintaining strict temporal coherence and physical realism.



Figure 11: **Instance repositioning capabilities.** By modifying the spatial coordinates of existing objects, the model naturally adapts the surrounding visual context, accurately hallucinating scene-consistent interactions such as shadows and water reflections.

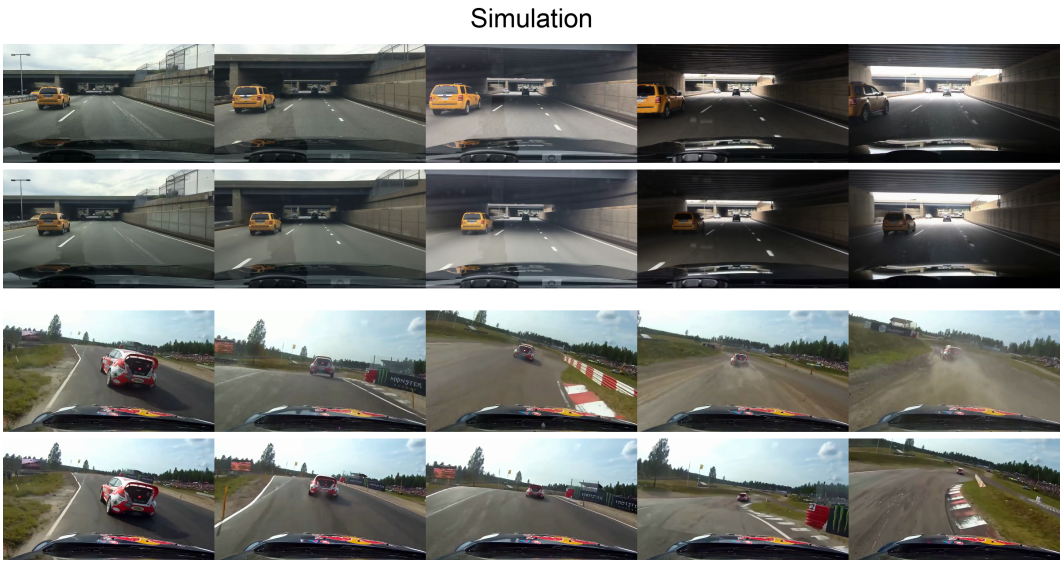

Figure 12: **Dynamics simulation.** The framework can generate counterfactual physical trajectories and realistic driving scenarios, demonstrating its potential for stress-testing perception and autonomous driving systems.



Figure 13: **Creative video editing.** Beyond strict physical simulation, the model effectively handles stylized, in-the-wild video footage and creative cinematic manipulations, preserving the aesthetic integrity of the original media.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the task, sparse-control assumptions, main technical components, benchmark, and empirical claims. The scope is further clarified by the evaluation protocol in Sec. 5 and limitations in Appendix B.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix B discusses dependence on sparse masks/depth, challenging cases such as long videos, severe occlusion and transparent objects, and the computational cost inherited from large video diffusion backbones.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper is empirical and does not present formal theoretical results, theorems, or proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper describes the model inputs, sparse-control protocol, benchmark construction, metrics, baselines, ablations, and user-study fitting procedure in Secs. 3–5 and Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code, development tools, model weights, and the released training data are publicly linked at https://xiangbogaobarry.github.io/PISCO/; the linked repositories provide training and usage instructions. Benchmark construction and preprocessing details are documented in Sec. 5 and Appendix C.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 5 specifies evaluation data, sparse-control settings, inference resolution and frame count, baselines, metrics, and user-study protocol; Appendix C provides training stages, optimizer, batch sizes, learning rates, and data construction details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The human preference study reports 95% bootstrap confidence intervals and explains the Bradley–Terry fitting, tie handling, and 300-round bootstrap procedure in Sec. 5.1. Automated video metrics are reported as aggregate benchmark scores over the curated evaluation set.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix C reports the GPU type, optimizer, training stages, steps, batch sizes, and resolutions. The supplementary material includes the corresponding training and inference configurations needed to estimate per-experiment compute.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research is conducted for controllable video editing and simulation, uses cited public research assets, avoids releasing personally identifying participant data, and discusses dual-use risks and safeguards in Appendix B.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix B discusses positive applications, dual-use risks such as deceptive media manipulation, and mitigation through curated evaluation, research/professional-use framing, and planned responsible-release terms.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Appendix B describes safeguards for responsible release, including curated non-sensitive benchmark clips, no release of private user data, and intended usage restrictions against impersonation, harassment, non-consensual manipulation, and misleading deployment.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Existing datasets, models, and baseline systems are cited throughout the paper and appendix. Appendix B documents asset roles, sources, licenses or access terms, provenance, preprocessing, and redistribution constraints, including that restricted raw source videos or model weights should be obtained from the original providers.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper introduces PISCO-Bench and documents its source data, mask refinement, removal-based clean-video construction, and evaluation protocol in Sec. 5 and Appendix B. Public code, tools, weights, and released training data are linked from the project page at `https://xiangbogaobarry.github.io/PISCO/`.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [Yes]

Justification: Sec. 5.1 reports the number of users, annotations, tie handling, Bradley–Terry fitting, and confidence intervals. Appendix B.1 provides the study instructions, interface details, consent and stopping information, risk characterization, and notes that participants were not recruited through a paid crowdsourcing marketplace; the supplementary material includes a screenshot of the study interface.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No]

    Justification: The user study did not collect personally identifiable information (PII) or expose participants to offensive content.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not an important, original, or non-standard component of the core method. A vision-language model is used only to generate text prompts for certain baselines, as described in the experimental setup.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.