CoFrame: Learning Real-World Video Retaking via Synchronized Reframing

Anonymous authors
Paper under double-blind review

Abstract

Video retaking aims to synthesize a new video that follows an input camera trajectory while preserving the appearance and motion depicted in the input video. Enabled by advances in conditional video generation models, together with synthetic multi-camera supervision, recent works have shown strong retaking performance, especially in terms of camera accuracy. However, faithfully preserving subject appearance and dynamics in real-world videos remains challenging, particularly for rapid movements and large displacements. A key challenge is still the lack of real synchronized multi-view recordings for training. In this work, we address this bottleneck from the views of complementary supervision. Specifically, we introduce CoFrame, a framework that constructs synchronized reframings from real monocular videos to complement synthetic multi-camera supervision for conditional re-rendering. Our key insight is that synthetic multi-camera pairs supervise extrinsic viewpoint changes, while synchronized real-video reframings provide complementary supervision for preserving real appearance and dynamics, with camera intrinsics as the bridge for expressing crop-and-resize transformations as explicit camera conditions. We further explore relative camera conditioning and paired flow matching to encode source-target camera relationships and jointly supervise synchronized target predictions. Experiments demonstrate improved motion preservation and camera accuracy over ReCamMaster, ReDirector, and Geo-Align, together with improved data efficiency under limited synthetic multi-camera supervision. Project page: https://coframe-gallery.pages.dev.

1 Introduction

Recent advances in video generation models have motivated growing interest in video retaking (Bai et al., 2025; Yu et al., 2025; Park et al., 2026). Given a monocular video, the aim of video retaking is to synthesize the same dynamic scene from requested viewpoints while preserving both the appearance and the events. This capability enables broad applications in cinematic production (Jiang et al., 2024), video stabilization (Zhu et al., 2026), virtual cinematography (Zhang et al., 2025), and 4D reconstruction (Wu et al., 2025) without requiring a physical reshoot.

Obtaining synchronized multi-view recordings of real-world dynamic scenes is challenging, limiting paired training data for video retaking. Therefore, *inpainting-based* methods (Yu et al., 2025; Paliwal et al., 2026) try to circumvent this need by formulating retaking as video completion and using geometric warping to construct training pairs from monocular videos, *i.e.*, with warped, incomplete videos as inputs and clean frames as supervision. At inference, the source video is similarly warped to the target viewpoints with offline geometric models, then completed with a video generation model. However, since these methods delegate viewpoint transformation to offline geometric reconstruction and rendering, their performance is sensitive to the warping quality, and errors in the warped inputs can propagate to the output as distortions of scene structure and subject appearance.

A recent study (Huang et al., 2026) shows that video generation models can learn strong 3D representations from video pretraining without explicit 3D supervision. These geometric representations, together with strong video synthesis capabilities, motivate a more direct formulation in which the generative model jointly learns viewpoint transformation and video synthesis. The MultiCamVideo dataset introduced by ReCamMaster (Bai et al., 2025) supports this formulation with accurate cam-

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

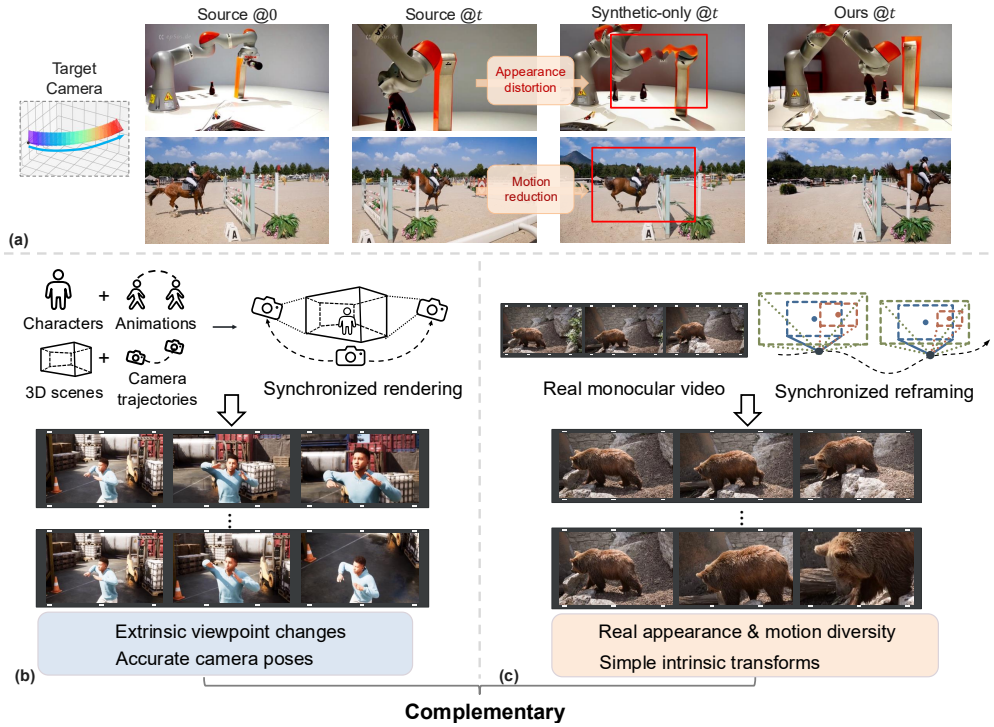

Figure 1: Motivation of CoFrame. **(a)** Models trained only on synthetic pairs can exhibit appearance distortion and motion reduction under viewpoint changes (red boxes). CoFrame better preserves the robotic arm's structure and the horse's progression across the obstacle. **(b)** Synthetic multi-camera rendering provides synchronized views with accurate camera poses and extrinsic viewpoint changes. **(c)** Synchronized reframings from real monocular videos preserve diverse appearance and dynamics under known intrinsic transformations. CoFrame combines both forms of supervision.

era annotations and synchronized multi-camera training pairs rendered from animated human characters in virtual environments. With such supervision, *conditional re-rendering* methods such as ReCamMaster (Bai et al., 2025) and ReDirector (Park et al., 2026) learn to generate target videos directly from source videos and camera conditions, allowing the model to exploit its geometric representations and synthesis capabilities. However, these methods' heavy reliance on human-character-based synthetic pairs leaves the rich diversity of appearance and motion in real-world videos underutilized, which can lead to poor appearance generalization and motion preservation, as illustrated in Figure 1(a). Therefore, a central question remains: can we derive synchronized supervision from diverse real-world videos to *complement* synthetic multi-camera data for conditional re-rendering?

To address this question, we introduce **CoFrame**, a framework for conditional re-rendering that supplements synthetic multi-camera supervision with synchronized real-video *reframings*, *i.e.*, videos extracted from local regions of the same recording. Our key insight is that synthetic multi-camera pairs supply accurately annotated extrinsic viewpoint changes, while real-video reframings provide complementary supervision for preserving diverse real-world appearance and dynamics under intrinsic changes, as illustrated in Figure 1(b,c). Camera intrinsics provide the bridge: reframing a single real recording creates synchronized source-target pairs with known relative intrinsic changes, making real appearance and motion available for camera-conditioned training without real multi-camera capture. We further explore a relative camera adapter to represent source-target camera relationships and paired flow matching to supervise flow differences across synchronized targets.

Experiments show that CoFrame improves motion preservation, camera accuracy, and cross-view fidelity over conditional re-rendering baselines, including ReCamMaster (Bai et al., 2025), ReDirector (Park et al., 2026), and Geo-Align (Li et al., 2026). CoFrame also demonstrates improved ground-truth data efficiency under limited synthetic multi-camera supervision. Our main contributions are as follows:

- We identify the motion preservation issue and supervision gap in conditional re-rendering, where real-world appearance and motion diversity remains underutilized.

- We introduce CoFrame, a framework for conditional re-rendering that combines synchronized real-video reframings with synthetic multi-camera pairs as complementary supervision for preserving real-world appearance and dynamics under viewpoint changes. We further explore paired flow matching and relative camera conditioning to model relationships across synchronized views.
- Our framework demonstrates improved motion preservation and camera accuracy on real-world videos and shows improved ground-truth data efficiency when using limited synthetic multi-camera supervision.

2 Related Work

Video Generation and 3D Priors. Video diffusion models (Blattmann et al., 2023; Wan et al., 2025) can learn appearance and motion priors from large-scale video data and show powerful generative capability. A recent probing study (Huang et al., 2026) further reveals strong 3D capabilities in pretrained video generation models without explicit 3D supervision. Multi-view video generation models (Bai et al., 2024; Watson et al., 2025; Wu et al., 2025) can synthesize videos across viewpoints and time. World models such as Genie (Bruce et al., 2024) and WorldDirector (Wang et al., 2026a) further explore interactive simulation and controllable scene evolution.

Inpainting-based video retaking methods (Yu et al., 2025; Ren et al., 2025) use estimated geometry to warp source videos into target viewpoints and complete the renderings with video diffusion. Subsequent methods extend geometric guidance to rendered depth in DepthDirector (Chen et al., 2026) or reconstruction features (Xie et al., 2026). However, reconstruction errors can propagate through this guidance. Reshoot-Anything (Paliwal et al., 2026) also constructs training pairs by cropping videos and combines them with synthetic multi-camera pairs, but still requires warping and anchor-guided synthesis. CoFrame instead uses crop-induced intrinsic variations as explicit camera supervision for direct conditional re-rendering.

Conditional Re-Rendering for Video Retaking. GCD (Van Hoorick et al., 2024) introduces a conditional generation pipeline with source videos and relative cameras. Recently, MultiCamVideo, introduced by ReCamMaster (Bai et al., 2025), provides synchronized videos and camera annotations for direct conditional re-rendering and significantly benefits the training of conditional generation models for this task. ReDirector (Park et al., 2026) introduces rotary camera encoding to model multi-view relationships within and across source and target videos, while supporting any-length generation. AnyView (Van Hoorick et al., 2026) combines dynamic multi-view videos with monocular videos of static scenes. Geo-Align (Li et al., 2026) uses geometric rewards for post-training without paired targets. TARS (Liu et al., 2026) uses different temporal segments of monocular videos and adds synchronized pairs during high-noise training. CoFrame instead constructs synchronized local views through simple deterministic transformations of estimated camera intrinsics.

3 Method

3.1 Problem Formulation and Preliminaries

Video Retaking. Given a source video $\mathbf{V}^s = \{\mathbf{I}_i^s\}_{i=1}^T$ with T frames and its source camera trajectory $\mathcal{C}^s$, video retaking aims to synthesize a target video $\mathbf{V}^t = \{\mathbf{I}_i^t\}_{i=1}^T$ depicting the same scene along a requested target camera trajectory $\mathcal{C}^t$ while preserving scene appearance, subject actions, and event timing. Under the pinhole camera model, a camera trajectory can be represented as $\mathcal{C} = \{(\mathbf{K}_i, \mathbf{P}_i)\}_{i=1}^T$, where $\mathbf{K}_i \in \mathbb{R}^{3\times3}$ is the intrinsic matrix and $\mathbf{P}_i \in \mathrm{SE}(3)$ is the camera-to-world pose in a shared coordinate system.

Conditional Re-rendering methods (Bai et al., 2025; Park et al., 2026) directly exploit the power of video diffusion models to jointly perform viewpoint transformation and video synthesis, with synchronized source-target pairs constructed for adapting the pretrained models. For training, using the rectified-flow formulation (Liu et al., 2022), a noisy latent is the linear interpolation between Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and the clean target latent: $\mathbf{z}_\tau^t = (1-\tau)\mathbf{z}^t + \tau\boldsymbol{\epsilon}$, where $\mathbf{z}^t$ is the clean target latent and $\tau \in [0, 1]$ is the noising time step. Let $\mathbf{c}$ denote the source-video, camera, and text conditions. A conditional model $v_\theta(\mathbf{z}_\tau^t, \tau; \mathbf{c})$ is therefore trained using the flow-matching objective:

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{(\mathbf{z}^t, \mathbf{c}), \tau, \boldsymbol{\epsilon}} \left[\left\| v_\theta(\mathbf{z}_\tau^t, \tau; \mathbf{c}) - (\boldsymbol{\epsilon} - \mathbf{z}^t) \right\|_2^2 \right]. \tag{1}$$

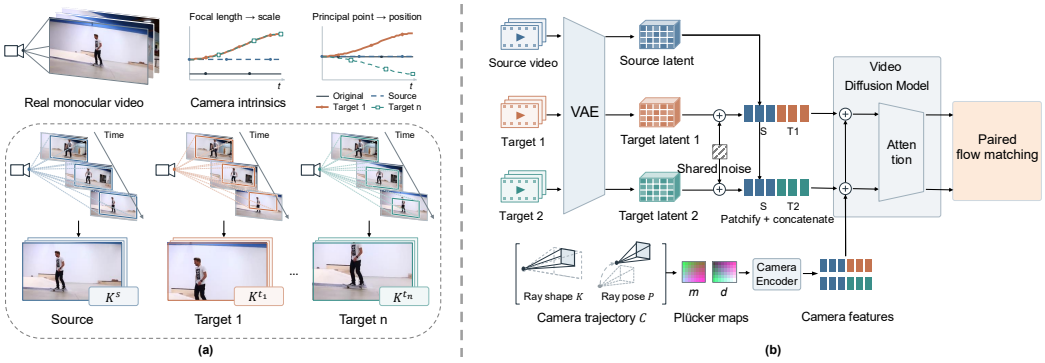

Figure 2: The proposed CoFrame framework. (a) CoFrame data construction. A real monocular video is reframed into one source with a fixed crop window and multiple synchronized targets following smooth crop trajectories. Focal length and principal point control crop scale and position, respectively. All reframings preserve recorded dynamics and share the original camera pose at each timestamp. (b) Paired camera-conditional training. Source and target camera trajectories are converted to Plücker ray maps and encoded into camera features that condition a shared video diffusion model. With a shared source, noise sample, and timestep across targets, paired flow matching supervises the difference between camera-conditioned predictions.

In this work, we directly adopt the design of ReDirector (Park et al., 2026) as our baseline.

3.2 CoFrame: Learning from Synchronized Reframings

To enable conditional re-rendering models to better preserve real subject dynamics, as shown in Figure 2, we introduce CoFrame, which turns real monocular recordings into synchronized retaking supervision through video reframing. As a complement to the synthetic multi-camera supervision, real-world videos can be collected from diverse sources. To make better use of these training sources, we introduce the following strategies.

Synchronized Reframing. As shown in Figure 2(a), for each monocular video, a source window is first selected and kept fixed in image coordinates, providing a stable view of source video. Then several targets start from the same first window but different trajectories are introduced over the frame sequence, changing the framing while preserving temporal correspondence with the source.

Specifically, for a video recording $\mathbf{V}^o = \{\mathbf{I}_i^o\}_{i=1}^T$, let $v \in \{s, t_1, \ldots, t_M\}$ be the source and M target views. For each view, we introduce a window $\mathbf{b}_i^v = (x_i^v, y_i^v, w_i^v, h_i^v)$, which specifies the crop's top-left corner, width, and height in the original frame. The reframed frame would be $\mathbf{I}_i^v = \mathcal{R}(\mathbf{I}_i^o; \mathbf{b}_i^v)$, where $\mathcal{R}$ crops this window from the input frame and resizes it to a common resolution $W \times H$. In our implementation, the source window is constant, *i.e.*, $\mathbf{b}_i^s = \mathbf{b}_1^s$, and all targets begin at $\mathbf{b}_1^{t_j} = \mathbf{b}_1^s$ and then smoothly interpolate along a sampled trajectory, jointly varying crop position and size to compose a new video. More details are shown in Appendix A.1.

From Reframing to Camera Conditions. To use these synchronized videos to complement synthetic data in conditional re-rendering, we need to connect their framing changes to the model's camera conditions. We use camera intrinsics as the bridge: each reframing is deterministically represented by an intrinsic trajectory, then using a shared camera encoder maps to conditioning features. As illustrated by the two plots in Figure 2(a), focal length controls image scale, while the principal point controls framing position. The target reframings start from the same first frame as the source, with crop-induced intrinsic transformations varying smoothly over time. Under the pinhole model, these intrinsic trajectories provide an equivalent parameterization of the crop-and-resize trajectories, while all reframings retain the original camera pose at each timestamp. Given the estimated intrinsics $\hat{\mathbf{K}}_i$ and camera-to-world pose $\hat{\mathbf{P}}_i$ of the original video, we define crop-and-resize transformation $\mathbf{A}_i^v$ and have

$$\mathbf{K}_i^v = \mathbf{A}_i^v \hat{\mathbf{K}}_i = \begin{bmatrix} \frac{W}{w_i^v}\hat{f}_{x,i} & 0 & \frac{W}{w_i^v}(\hat{c}_{x,i} - x_i^v) \\ 0 & \frac{H}{h_i^v}\hat{f}_{y,i} & \frac{H}{h_i^v}(\hat{c}_{y,i} - y_i^v) \\ 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{P}_i^v = \hat{\mathbf{P}}_i. \tag{2}$$

where $(\hat{f}_{x,i}, \hat{f}_{y,i})$ and $(\hat{c}_{x,i}, \hat{c}_{y,i})$ are the original-frame focal lengths and principal point in pixels.

This construction provides synchronized source-target videos with deterministic relative intrinsic transformations and brings *no additional camera estimation error*. All views are cropped and resized from the same recorded frame, preserving the underlying appearance, motion and action. As the construction requires neither depth-based warping nor 3D rendering, it also avoids warping-induced holes and geometric distortions caused by depth or reconstruction errors.

Model training. By expressing reframing transformations as changes in camera intrinsics, we can directly incorporate the constructed video pairs into conditional re-rendering training alongside synthetic multi-camera data. The reframings retain real-world appearance and actions under intrinsic variations, while synthetic multi-camera videos (Bai et al., 2025) supply extrinsic viewpoint changes, including parallax and visibility changes. This *complementary supervision* trains a shared model through the same flow-matching objective. The conditional training is shown in Figure 2(b). Given sampled source and target videos from either dataset, we encode the shared source and individual targets into video latents, add noise to the target latents, and concatenate source tokens with the noisy tokens of targets. The conditional video DiT predicts a separate flow for each source-target pair using shared model parameters, with source and target camera conditions supplied in the form of camera embeddings and rotary camera encoding Park et al. (2026). For training with our constructed data, we fix the source for each scene and vary only the target for stable optimization, while for training with synthetic data, we follow Bai et al. (2025); Park et al. (2026) and randomly select the source and target from a scene.

Paired Flow Matching. To better capture camera-dependent variations of the same dynamic scene, we introduce paired flow matching across synchronized targets sharing the same source. Alongside the standard flow-matching objective for each source-target pair, we supervise differences between the predicted flows of these targets. Specifically, for a group of $M \geq 2$ targets, let $\mathbf{z}^{t_j}$ denote the clean latent of target j and $\mathbf{c}_j$ denote its conditions. We corrupt all target latents using the same noise $\boldsymbol{\epsilon}$ and timestep τ and predict their flows separately as $\hat{\mathbf{v}}_j = v_\theta(\mathbf{z}_\tau^{t_j}, \tau; \mathbf{c}_j)$. Taking the first target as a reference, we compute the ground-truth flow differences as:

$$(\boldsymbol{\epsilon} - \mathbf{z}^{t_1}) - (\boldsymbol{\epsilon} - \mathbf{z}^{t_j}) = \mathbf{z}^{t_j} - \mathbf{z}^{t_1}, \tag{3}$$

where $\mathbf{z}_\tau^{t_j} = (1-\tau)\mathbf{z}^{t_j} + \tau\boldsymbol{\epsilon}$. Therefore, we can match the predicted flow differences to the corresponding clean-latent differences using

$$\ell_{\text{pair}} = \frac{1}{M-1} \sum_{j=2}^{M} \left\| (\hat{\mathbf{v}}_1 - \hat{\mathbf{v}}_j) - (\mathbf{z}^{t_j} - \mathbf{z}^{t_1}) \right\|_2^2 . \tag{4}$$

The overall training objective combines the two losses $\mathcal{L} = \mathcal{L}_{\text{flow}} + \lambda_{\text{pair}}\mathcal{L}_{\text{pair}}$, where $\mathcal{L}_{\text{pair}}$ averages ℓ_{pair} over active training groups. The paired training steps are activated with probability p_{pair} and apply the paired term only at high noise levels (*i.e.*, $\tau \geq \tau_{\min}$), providing auxiliary gradients under stronger corruption. More training details are provided in Appendix A.1.

Relative Camera Conditioning. Following ReDirector (Park et al., 2026), we use a camera encoder to encode source and target camera trajectories as Plücker ray fields and map them to camera features $\mathbf{h}^s$ and $\mathbf{h}^{t_j}$. Besides, to reduce sensitivity to noise and variability in camera estimates from real-world videos, we further introduce a relative camera adapter that explicitly models the source-target camera relationship when injecting camera conditions. The target camera features are updated as:

$$\widetilde{\mathbf{h}}^{t_j} = \mathbf{h}^{t_j} + g\big([\mathbf{W}\mathbf{h}^{t_j}, \mathbf{W}\mathbf{h}^{s}, \mathbf{W}(\mathbf{h}^{t_j} - \mathbf{h}^{s})]\big), \tag{5}$$

where $\mathbf{W}$ is a shared projection layer and g is a small MLP whose output projection is initialized to zero. The updated target camera features and source camera features are added to their corresponding video tokens. More details can be found in Appendix A.2.

4 Experiments

4.1 Experimental Setup

Training Data. To construct a training dataset with dynamic real-world videos, following DynPose-100K (Rockwell et al., 2025), we collect 6,525 video clips from YouTube and use estimated camera

Table 1: Camera accuracy, source fidelity, and video quality on DAVIS. We compare with TrajectoryCrafter (Yu et al., 2025), DepthDirector (Chen et al., 2026), ReCamMaster (Bai et al., 2025), ReDirector (Park et al., 2026), and Geo-Align (Li et al., 2026). Temporal Quality and Frame Quality are group means of VBench metrics. TrajectoryCrafter uses CogVideoX-V1.1-5B base model; DepthDirector uses Wan2.2-5B; ReCamMaster, ReDirector, Geo-Align, and ours use Wan2.1-1.3B.

	Camera Accuracy		Content & Motion		VBench		
Method	TransErr↓	RotErr↓	Mat.Pix↑	MME↓	Dynamic↑	Temporal Quality↑	Frame Quality↑
Source video	-	-	-	-	1.000	0.952	0.622
Inpainting-based methods							
TrajectoryCrafter	0.0386	2.16	408	0.714	1.000	0.965	0.577
DepthDirector	0.0660	11.73	305	0.457	0.998	0.962	0.514
Conditional re-rendering methods							
ReCamMaster	0.0198	2.11	385	0.610	0.962	0.978	0.582
ReDirector	0.0143	1.69	381	0.521	0.966	0.970	0.589
Geo-Align	0.0155	1.48	405	0.544	0.925	0.971	0.591
Ours	**0.0124**	**1.32**	**466**	**0.387**	0.996	0.970	0.584

parameters from DynPose-100K. One source video and up to seven target videos are constructed for training. Figure 4 shows two constructed training examples of outdoor locomotion and fine-grained hand-object interaction. The target reframings vary in crop position and scale while retaining the synchronized progression of the recorded actions. For synthetic multi-camera videos, we directly use data from MultiCamVideo (Bai et al., 2025), which contains 13,600 dynamic scenes with ten synchronized camera views per scene. During training, we sample data items from these two datasets with equal probability. More data construction details are given in Appendix A.1.

Implementation Details. Our model follows ReDirector (Park et al., 2026), which uses pretrained Wan-I2V-CamCtrl-1.3B (Wan et al., 2025) as initialization. We train for 20,000 steps with AdamW using a learning rate of 10^{-4} and bfloat16 mixed precision on eight RTX Pro 6000 Blackwell GPUs. We update the self-attention layers, rotary camera encoding, and the relative camera adapter, with the remaining pretrained weights frozen. The paired flow training is activated with a probability of $p_{\text{pair}} = 0.3$, loss weight of $\lambda_{pair} = 0.1$ and is applied for noise levels $\tau \geq 0.5$. The relative adapter has an intermediate 64-channel bottleneck. More training and inference settings are provided in Appendix A.1 and Appendix A.2.

Baselines. The baselines include both conditional re-rendering and inpainting-based methods: ReDirector (Park et al., 2026), ReCamMaster (Bai et al., 2025), Geo-Align (Li et al., 2026), TrajectoryCrafter (Yu et al., 2025), and DepthDirector (Chen et al., 2026). We reproduce ReDirector under the same training setup as CoFrame and use publicly released checkpoints for the other baselines. Each method receives the same source video and requested target trajectory, with inputs converted to its camera convention, retaining each baseline's inference recipe. For inpainting-based methods, we obtain the source poses and intrinsics from Pi3X (Wang et al., 2026b) and perform warping and rendering after lifting the source video to 3D space. DepthDirector is run at 81-frame training length using the Wan2.2-TI2V-5B (Wan et al., 2025) base model, while TrajectoryCrafter generates 49 frames with CogVideoX (Yang et al., 2024). All comparisons use fixed generation seeds.

Evaluation Data. Following ReDirector (Park et al., 2026), we use the DAVIS (Perazzi et al., 2016) test dataset as our main evaluation dataset. We retain 42 of 50 sequences whose estimated rotation changes by less than 3° between consecutive frames to ensure camera pose can be reliably estimated for evaluation. To test the camera variability, following ReCamMaster (Bai et al., 2025), each test case is asked to generate new videos following ten pre-defined target trajectories. The pre-defined trajectories cover camera rotation, translation, and arcs. Besides, we also use iPhone (Gao et al., 2022) dataset, which contains synchronized recordings from different cameras, to validate our method. We evaluate on seven scenes, using an 81-frame moving-camera recording as the source and a held-out static camera as the target, with dataset-provided camera parameters.

Camera Accuracy and Content Preservation Evaluation. Translation error (TransErr) and rotation error (RotErr) (He et al., 2024) are used to measure the camera accuracy, where TransErr measures the difference between requested and estimated camera paths after alignment of position, orientation, and scale, and RotErr measures the mean angular discrepancy, in degrees, between rel-

ative rotations over frame pairs. ViPE (Huang et al., 2025) is used to estimate camera pose from generated videos. To assess how much source content is retained, Mat.Pix measures local correspondences through geometrically verified image matches using GIM-LightGlue (Shen et al., 2024; Lindenberger et al., 2023). Appendix A.3 provides more detailed calculation of each metric.

Motion Magnitude Error (MME). Subject motion preservation is important for dynamic video retaking. However, camera accuracy and appearance similarity metrics used by previous methods do not directly measure whether subject motion is weakened or exaggerated. Besides, the dynamic degree metric in VBench only detects motion in generated videos without comparing its magnitude with that in the source video. Therefore, inspired by the trajectory-based motion fidelity metric (Yatim et al., 2024), we introduce Motion Magnitude Error (MME) to quantify motion-magnitude preservation using the source video as a reference. Specifically, we track points with CoTracker3 (Karaev et al., 2025) and use the median displacement between consecutive frames to approximate camera-induced image motion and subtract it from each track's displacement. Each source subject track is then matched to an output track based on residual displacement directions. Specifically, for test sample i, let m_i^s and m_i^g denote the mean residual image-plane displacement magnitudes over source subject tracks and their matched output tracks, MME is defined as:

$$\mathrm{MME} = \frac{1}{N}\sum_{i=1}^{N}|1 - r_i|, \qquad r_i = \frac{m_i^g}{m_i^s + \varepsilon}, \tag{6}$$

where N is the number of test samples and ε is for numerical stabilization. Lower MME would indicate closer agreement in average residual motion magnitude. See Appendix A.4 for more details.

Visual Quality Evaluation. We use VBench (Huang et al., 2024) to measure detected motion, temporal quality, and frame quality. We report Dynamic for "dynamic degree", Temporal Quality for the average of "temporal flickering" and "motion smoothness", and Frame Quality for the average of "aesthetic quality" and "imaging quality". The full metric results are provided in Appendix A.3. Besides, on the iPhone dataset, we report full-frame PSNR, which measures pixel agreement in dB, and LPIPS (Zhang et al., 2018), which measures perceptual distance to the recorded target view.

4.2 Main Results

Camera accuracy and source-target consistency. Table 1 compares camera accuracy and source-target consistency across all methods. CoFrame achieves the lowest TransErr of 0.0124 and RotErr of 1.32°, together with the highest matched-pixel count of 466, indicating more accurate camera control and stronger local source-target correspondences. Importantly, CoFrame also achieves the lowest MME, scoring 0.387 compared with 0.610 for ReCamMaster, 0.521 for ReDirector, and 0.544 for Geo-Align. For inpainting-based methods, TrajectoryCrafter has a TransErr of 0.0386, a RotErr of 2.16° and an MME of 0.714, while DepthDirector exhibits larger camera errors on these dynamic DAVIS sequences.

Novel-view synthesis on DyCheck iPhone. DyCheck iPhone (Gao et al., 2022) provides a recorded target view, allowing us to evaluate whether a retaken video matches the real target video. The results are shown in Table 2. CoFrame achieves the highest PSNR and lowest LPIPS. Specifically, relative to ReDirector, PSNR increases from 11.12 to 11.41 dB and LPIPS decreases from 0.6286 to 0.6008, indicating better agreement with the recorded target views. More details can be found in Appendix A.5.

Table 2: Novel-view quality on DyCheck iPhone.

Method	PSNR↑	LPIPS↓
TrajectoryCrafter	10.82	0.6559
ReCamMaster	10.71	0.6787
ReDirector	11.12	0.6286
Ours	**11.41**	**0.6008**

VBench evaluation. We compare all methods using VBench metrics in Table 1, with scores for the input video included for reference. CoFrame retains a high dynamic degree of 0.996, compared with 0.966 for ReDirector and 0.962 for ReCamMaster. For the visual quality metrics, most methods perform similarly, with ReCamMaster achieving the highest Temporal Quality and Geo-Align the highest Frame Quality. The full VBench results are reported in Appendix A.3.

Qualitative comparison. Figure 3 shows three examples depicting subjects undergoing rapid motion or large displacements, each with a different type of target camera trajectory. In the car-drifting example, the source video shows a fast-moving car and substantial camera rotation. TrajectoryCrafter preserves the car but introduces substantial distortion. ReCamMaster fails to preserve

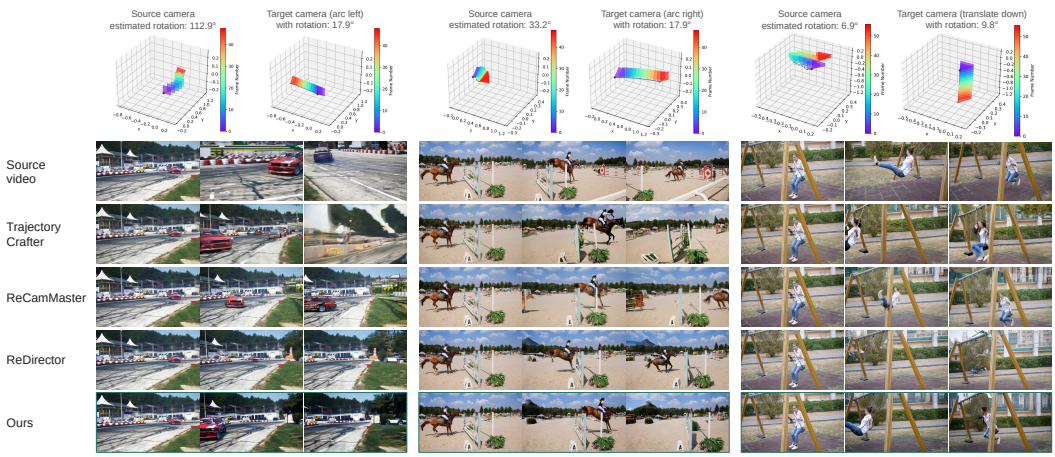

Figure 3: Qualitative comparisons, from left to right: car drifting, horse jumping, and swinging. Source inputs are shown in the first row, and the camera plots show the source and requested target trajectories. Frames are sampled at 0%, 50%, and 100% of each clip. More qualitative results can be found at https://coframe-gallery.pages.dev.

Table 3: Component ablation on DAVIS. Components are added cumulatively and metrics and VBench group means follow Table 1.

	Camera Accuracy		Content & Motion		VBench		
Configuration	TransErr↓	RotErr↓	Mat.Pix↑	MME↓	Dynamic↑	Temporal Quality↑	Frame Quality↑
Baseline	0.0143	1.69	381	0.521	0.966	0.970	0.589
+ reframings	0.0155	1.68	**469**	0.423	0.994	0.969	0.587
+ paired training	0.0129	1.58	443	0.424	1.000	0.969	0.591
+ adapter (**Ours**)	**0.0124**	**1.32**	466	**0.387**	0.996	0.970	0.584

the car's motion correctly, while ReDirector fails to retain the car. CoFrame, in contrast, retains the car's passage through the scene with the same motion. In the horse-jumping example, CoFrame maintains good visual quality as the horse and rider move through the jump under the requested arc-right camera motion. In the swing example, CoFrame preserves the rider's appearance and swinging back-and-forth motion, whereas ReCamMaster produces degraded motion and ReDirector loses much of the rider's body.

4.3 Ablation Studies

We report ablation studies of CoFrame's key designs in Table 3, including real-video reframing supervision, paired flow matching, and relative camera conditioning. We use our reproduced ReDirector as the baseline and train each ablation configuration separately with the same initialization and training budget, adding components cumulatively.

Effect of real-video reframing supervision. First, we add real-video reframings as complementary supervision during training. As shown in Table 3, compared with synthetic-only training, this raises dynamic degree from 0.966 to 0.994, increases the matched-pixel count from 381 to 469, and reduces MME from 0.521 to 0.423, while Temporal Quality, Frame Quality and camera accuracy remain close to the baseline. Although camera accuracy does not improve at this stage, these results highlight reframing supervision as a simple way to introduce complementary real-world appearance and dynamics for stronger motion preservation.

Effect of paired flow training. Adding paired flow training reduces TransErr from 0.0155 to 0.0129 and RotErr from 1.68° to 1.58° in Table 3, while maintaining the high dynamic degree and low MME achieved by real-video reframing supervision. Since synchronized targets depict the same scene as the source video, supervising flow differences explicitly couples predictions across target

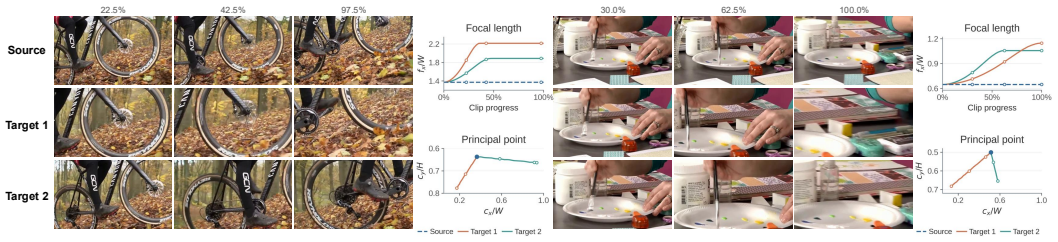

Figure 4: Examples of constructed real-video reframing data: forest cycling (left) and handcraft (right). The adjacent plots show focal-length and principal-point trajectories, with hollow markers corresponding to the displayed frames.

views and emphasizes relative changes between them, complementing the flow-matching objective for individual views to improve camera accuracy.

Effect of relative camera conditioning. Finally, adding the relative camera adapter further reduces TransErr to 0.0124 and RotErr to $1.32°$. By explicitly encoding the source-target camera relationship, the adapter provides a direct representation of the requested viewpoint change, supporting more accurate camera control. The full model also achieves a lower MME of 0.387, indicating closer agreement with the source motion magnitude alongside the gains in camera accuracy.

4.4 Ground-Truth Camera Supervision Efficiency

Obtaining diverse ground-truth multi-view videos with accurate camera annotations is costly, even when using synthetic engines. We therefore evaluate whether CoFrame can reduce the amount of such supervision by supplementing it with cheaper real-video reframings. To this end, we explore training with subsets of the MultiCamVideo dataset and our reframing data. Specifically, we subsample MultiCamVideo by scene, retaining all ten camera views of each selected scene. The resulting 1% and 10% subsets contain 136 and 1,360 scenes, respectively. We also train ReDirector with the same subset of synthetic data for comparison.

Table 4: Ground-truth camera supervision efficiency. Scene percentages refer to MultiCamVideo. MME is computed as in Table 1.

Method	Scenes	TransErr↓	RotErr↓	MME↓
ReDirector	1%	0.0183	1.70	0.653
	10%	0.0186	1.67	0.512
Ours	0%	0.0379	3.26	0.339
	1%	0.0164	1.84	0.436
	10%	0.0160	1.54	0.487

As shown in Table 4, the model trained without synthetic multi-camera data has much larger camera errors of 0.0379 and $3.26°$, together with the lowest motion magnitude error of 0.339. Such behavior is as expected as real-video reframings vary camera intrinsics but do not provide extrinsic viewpoint changes between synchronized views. With 1% of the ground-truth data, CoFrame achieves lower translation error and MME than ReDirector, with slightly higher rotation error. With 10% of the ground-truth data, CoFrame improves both camera errors, from 0.0186 to 0.0160 in translation error and from $1.67°$ to $1.54°$ in rotation error. These results support CoFrame's effectiveness with limited synthetic multi-camera supervision.

5 Conclusion and Limitations

In this work, we present CoFrame, a framework for dynamic video retaking that introduces synchronized real-video reframings to complement synthetic multi-camera supervision. As a result, CoFrame improves motion and appearance preservation over conditional re-rendering methods such as ReCamMaster and ReDirector. It also makes more effective use of limited synthetic multi-camera supervision. Relative camera modeling and paired flow matching training further improve camera accuracy while maintaining the gains in motion and appearance preservation. These results show that reframing real monocular videos is a simple but practical way to bring real-world motion diversity into conditional re-rendering while supporting accurate camera control. Future work could scale real-video reframing data and examine how data diversity influences conditional re-rendering. It could also explore more varied reframing trajectories and sampling strategies for combining real-video and synthetic multi-camera supervision.

AI USE STATEMENT

In this work, we used generative AI tools to polish the writing process, discover relevant papers and code review. We have read and checked all edits and content, and all scientific work was done and verified by at least 2 authors. We have reviewed all AI-assisted work and take responsibility for the content, claims, text and research artifacts in this paper.

REPRODUCIBILITY STATEMENT

We provide a self-contained description of the dataset sources, data processing procedures, baseline implementation details, training hardware, and evaluation metrics to support reproducibility. We will also make our code publicly available.

REFERENCES

Jianhong Bai, Menghan Xia, Xintao Wang, Ziyang Yuan, Xiao Fu, Zuozhu Liu, Haoji Hu, Pengfei Wan, and Di Zhang. Syncammaster: Synchronizing multi-camera video generation from diverse viewpoints. *arXiv preprint arXiv:2412.07760*, 2024.

Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14834–14844. IEEE, 2025.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first international conference on machine learning*, 2024.

Dong-Yu Chen, Yixin Guo, Shuojin Yang, Tai-Jiang Mu, and Shi-Min Hu. Beyond inpainting: Unleash 3d understanding for precise camera-controlled video generation. *arXiv preprint arXiv:2601.10214*, 2026.

Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. *Advances in Neural Information Processing Systems*, 35: 33768–33780, 2022.

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

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

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024.

Zixuan Huang, Xiang Li, Zhaoyang Lv, and James M Rehg. How much 3d do video foundation models encode? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 384–394, 2026.

Xuekun Jiang, Anyi Rao, Jingbo Wang, Dahua Lin, and Bo Dai. Cinematic behavior transfer via nerf-based differentiable filming. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6723–6732. IEEE, 2024.

Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labeling real videos. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2025.

Zizun Li, Haoyu Guo, Runzhe Teng, Chunhua Shen, and Tong He. Geo-align: Video generation alignment via metric geometry reward. *arXiv preprint arXiv:2605.23903*, 2026.

Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 17581–17592. IEEE, 2023.

Jiwen Liu, Shujuan Li, Xiaohan Li, Zijie Meng, Xinyue Liu, Yulong Xu, Yan Zhou, and Guoxin Zhang. Tars: Timestep-aware data scaling for 3d-free video re-shooting. *arXiv preprint arXiv:2607.28261*, 2026.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Avinash Paliwal, Adithya Iyer, Shivin Yadav, Muhammad Afridi, and Midhun Harikumar. Reshoot-anything: A self-supervised model for in-the-wild video reshooting. In *CVPR Workshops*, pp. 11596–11606, June 2026.

Byeongjun Park, Byung-Hoon Kim, Hyungjin Chung, and Jong Chul Ye. Redirector: Creating any-length video retakes with rotary camera encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11163–11173, 2026.

Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 724–732, 2016.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6121–6132. IEEE, 2025.

Chris Rockwell, Joseph Tung, Tsung-Yi Lin, Ming-Yu Liu, David F Fouhey, and Chen-Hsuan Lin. Dynamic camera poses and where to find them. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12444–12455. IEEE, 2025.

Xuelun Shen, Zhipeng Cai, Wei Yin, Matthias Müller, Zijun Li, Kaixuan Wang, Xiaozhi Chen, and Cheng Wang. Gim: Learning generalizable image matcher from internet videos. In *The Twelfth International Conference on Learning Representations*, 2024.

Basile Van Hoorick, Rundi Wu, Ege Ozguroglu, Kyle Sargent, Ruoshi Liu, Pavel Tokmakov, Achal Dave, Changxi Zheng, and Carl Vondrick. Generative camera dolly: Extreme monocular dynamic novel view synthesis. In *European Conference on Computer Vision*, pp. 313–331. Springer, 2024.

Basile Van Hoorick, Dian Chen, Shun Iwase, Pavel Tokmakov, Muhammad Zubair Irshad, Igor Vasiljevic, Swati Gupta, Fangzhou Cheng, Sergey Zakharov, and Vitor Campagnolo Guizilini. Anyview: Synthesizing any novel view in dynamic scenes. *arXiv preprint arXiv:2601.16982*, 2026.

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

Hanlin Wang, Hao Ouyang, Qiuyu Wang, Wen Wang, Qingyan Bai, Ka Leong Cheng, Yue Yu, Yixuan Li, Yihao Meng, Zichen Liu, et al. Worlddirector: Building controllable world simulators with persistent dynamic memory. *arXiv preprint arXiv:2607.02517*, 2026a.

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

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

Daniel Watson, Saurabh Saxena, Lala Li, Andrea Tagliasacchi, and David Fleet. Controlling space and time with diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 95989–96012, 2025.

Rundi Wu, Ruiqi Gao, Ben Poole, Alex Trevithick, Changxi Zheng, Jonathan T Barron, and Aleksander Holynski. Cat4d: Create anything in 4d with multi-view video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26057–26068, 2025.

Mingyang Xie, Numair Khan, Tianfu Wang, Naina Dhingra, Seonghyeon Nam, Haitao Yang, Zhuo Hui, Christopher Metzler, Andrea Vedaldi, Hamed Pirsiavash, et al. Lavr: Scene latent conditioned generative video trajectory re-rendering using large 4d reconstruction models. *arXiv preprint arXiv:2601.14674*, 2026.

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.

Danah Yatim, Rafail Fridman, Omer Bar-Tal, Yoni Kasten, and Tali Dekel. Space-time diffusion features for zero-shot text-driven motion transfer. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8466–8476. IEEE, 2024.

Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 100–111. IEEE, 2025.

Mengchen Zhang, Tong Wu, Jing Tan, Ziwei Liu, Gordon Wetzstein, and Dahua Lin. Gendop: Auto-regressive camera trajectory generation as a director of photography. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18229–18239. IEEE, 2025.

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

Muhua Zhu, Xinhao Jin, Xinping Wang, Yu Zhang, Yifei Xue, Tie Ji, and Yizhen Lao. Vs3r: Robust full-frame video stabilization via deep 3d reconstruction. *arXiv preprint arXiv:2603.05851*, 2026.

A Experimental Details

A.1 Data Construction and Training

Real-Video Construction. We extract 81-frame windows from each collected video at its native frame rate, sampling the start frame uniformly among valid positions. For each window, we retain a fixed source crop and construct up to seven targets with different crop trajectories, resizing all crops to 480×832 (height $\times$ width). We use the original-video camera parameters released with DynPose (Rockwell et al., 2025). Note that this construction is not specific to DynPose videos. It can be applied to any Internet videos, while camera parameters can be readily estimated using recent tools (Huang et al., 2025; Wang et al., 2025). To avoid low-resolution crops, we cap source crop enlargement at $1.3\times$ and target crop enlargement at $1.5\times$. The final training set contains 6,525 clips, approximately half the number of scenes in MultiCamVideo.

Crop Sampling. For each video clip, we first randomly sample the source window's size and position within the original frame and keep the window fixed over time. Each target starts from the same first-frame window and ends at a crop with a randomly sampled position and scale. All crops preserve the output aspect ratio and remain inside the original frame without padding. To encourage distinct target framings, the IoU between target endpoints is capped at 0.88.

Crop Trajectories. Each target moves from the source window to its sampled endpoint over a randomly sampled duration. The crop window follows $(1 - s(u))\mathbf{b}_1^s + s(u)\mathbf{b}_T^t$, where $s(u) = 3u^2 - 2u^3$ and $u \in [0, 1]$ denotes normalized motion progress. The endpoint is held fixed for the remaining frames.

Camera Trajectories. The original camera intrinsic matrix provided by DynPose is constant within each clip, so the fixed source crop also has constant intrinsics, while target intrinsics follow the crop transformations in Eq. 2. We interpolate the original camera trajectory to the sampled frame timestamps, using spherical linear interpolation for rotations and linear interpolation for camera centers. All poses are expressed relative to the first sampled pose. Source and target reframings share these poses at each timestamp and require no separate camera estimation.

Sampling and Optimization. Real-video and synthetic samples are drawn with equal probability. Following Park et al. (2026), synthetic clips are sampled in forward or reversed time with equal probability. Video latents and text embeddings are precomputed before training, with video autoencoder and text encoder kept frozen. AdamW uses a constant learning rate of 10^{-4}, and gradient checkpointing is used to reduce training memory. The relative camera adapter's output projection is zero-initialized.

Paired Training. Each paired training group contains one source and $M = 2$ distinct synchronized targets. Paired groups are used with probability $p_{\text{pair}} = 0.3$, while the remaining steps use one uniformly sampled target. Standard flow matching supervises every target and targets in a camera-conditioned group share a Gaussian noise sample and timestep, with the auxiliary loss in Eq. 4 applied at weight $\lambda_{\text{pair}} = 0.1$ for $\tau \geq 0.5$. When camera conditions are dropped, targets use independent noise and timesteps and receive only standard flow supervision. The high-noise condition only restricts the auxiliary loss rather than changing the timestep sampling distribution.

Noise Sampling and Loss Reduction. Following the ReDirector (Park et al., 2026) implementation, we sample noise levels uniformly from the shifted grid $\tau_k = 5u_k/(1 + 4u_k)$, where $u_k = k/1000$ for $k \in \{1, \ldots, 1000\}$. Both objectives use timestep weights

$$w_k = \frac{q_k - q_{\min}}{\frac{1}{1000}\sum_{l=1}^{1000}(q_l - q_{\min})}, \qquad q_k = \exp[-2(\tau_k - 0.5)^2],$$

where $q_{\min} = \min_l q_l$. Each objective is a weighted mean squared error over all latent elements. The flow objective averages over all targets, while the paired objective averages over active groups in each local minibatch and is zero when no group is active.

A.2 Implementation Details

Camera Encoding and Relative Adapter. Following ReDirector (Park et al., 2026), a shared camera encoder maps source and target Plücker ray fields to features aligned with the video token grid. The features are added to the corresponding source and target tokens. Each pixel's ray is represented by $(\mathbf{o} \times \mathbf{d}, \mathbf{d})$, where $\mathbf{o}$ is the camera center and $\mathbf{d}$ is the unit ray direction in world coordinates, determined by the camera intrinsics and pose. Groups of four ray maps form 24-channel inputs aligned with the video latent's temporal stride, with four copies of the first ray map at the initial timestep. Spatial encoding uses pixel unshuffle by a factor of eight, a patch-strided convolution, and one residual block. ReDirector's rotary camera encoding on the query/key and value/output paths is retained in every self-attention block.

For our relative adapter, as indicated by Eq. 5, a shared pointwise 3D convolution projects each branch to 64 channels. A pointwise convolution, SiLU activation, and output projection then process the concatenated target, source, and target-minus-source features to produce a residual for the target branch, while the source branch is unchanged.

Conditioning and Camera Augmentation. During training, camera and text conditions are independently dropped with probability 0.1. Text dropout replaces the caption embedding with the empty-prompt embedding. Camera dropout removes source and target camera features while retaining the source video. Since we cannot ideally estimate camera parameters from real-world videos, source poses are perturbed with a probability of 0.5. We keep the first source pose unchanged and sample independent Gaussian rotation and translation offsets at one-third, two-thirds, and the end of the clip. Offsets for intermediate frames are obtained by linear interpolation. The sampled offsets

Table 5: Full VBench scores on DAVIS.

Method	Dynamic↑	Subject↑	Backgr.↑	Aesth.↑	Imaging↑	Flicker↑	Smooth↑
Source video	1.000	0.916	0.921	0.559	0.684	0.930	0.975
TrajectoryCrafter	1.000	0.902	0.920	0.504	0.649	0.948	0.981
DepthDirector	0.998	0.795	0.870	0.424	0.604	0.949	0.975
ReCamMaster	0.962	0.926	0.924	0.502	0.662	0.966	0.989
ReDirector	0.966	0.905	0.914	0.511	0.667	0.955	0.985
Geo-Align	0.925	0.908	0.915	0.512	0.671	0.956	0.986
Ours	0.996	0.892	0.907	0.500	0.669	0.954	0.986

have standard deviations of 1.5° for rotation and 3% of the source path length for translation. The same source-pose perturbation is shared across targets during paired training. Target poses receive no additional pose noise.

Inference. CoFrame uses a fixed random seed of 0, running with 50 denoising steps. We use a camera-based classifier-free guidance (CFG) as $\hat{v} = \hat{v}_{\text{no camera}} + \lambda_{cfg}(\hat{v}_{\text{camera}} - \hat{v}_{\text{no camera}})$, with $\lambda_{cfg} = 3$. This additional guidance is omitted for the first latent frame for inference on DAVIS. The input source-camera translations are normalized to unit path length, and the pre-defined target trajectory (Bai et al., 2025) retains its pre-defined scale.

A.3 DAVIS Evaluation

Target Camera Trajectories. Following previous works (Park et al., 2026; Bai et al., 2025), we use the ten pre-defined trajectories defined by ReCamMaster (Bai et al., 2025) for our evaluation, which include: (1) Camera 1–2: pan right and left by about 19.9°; (2) Camera 3–4: tilt up and down by about 9.9°. (3) Camera 5–6: translate forward and backward without rotation, each with a path length of about 1.99 in the camera units used for evaluation. (4) Camera 7–8: translate up and down while rotating by about 14.0°, with a path length of about 0.99; (5) Camera 9–10: left and right arcs with about 29.8° of rotation and a path length of about 2.06. When the output length differs from the 81-frame preset, we resample its frame-to-frame rotation and camera-local translation increments.

Camera Errors. Our camera errors implementation follows ReDirector (Park et al., 2026): relative rotations are compared over all ordered pairs of distinct frames, and translation error is computed after a similarity alignment of the estimated camera centers to the requested camera centers. TransErr therefore measures normalized path agreement after scale and coordinate alignment, rather than absolute translation in physical units.

Frame Correspondences. Outputs are resized to the source resolution before frame-based source comparisons. Mat.Pix averages the number of GIM-LightGlue (Shen et al., 2024; Lindenberger et al., 2023) matches with confidence above 0.5 that survive fundamental-matrix RANSAC with a one-pixel threshold.

A.4 Motion Magnitude Error

We use CoTracker3 (Karaev et al., 2025) to track a 55×55 grid of points initialized in the first frame of each video. Source tracks whose initial positions fall inside the first-frame ViPE dynamic mask are treated as subject tracks. We approximately remove camera-induced image motion by subtracting the median displacement of source background tracks and the median displacement of all output tracks from output tracks. Each source subject track is matched to the output track with the highest time-averaged cosine similarity between these residual displacement directions, adapting the trajectory-matching idea of Yatim et al. (2024). Matches are selected independently for each source track. In details, for clip i with K_i source subject tracks, let $\pi_i(k)$ denote the output track matched to source track k, and let $\mathbf{d}^s_{i,k,t}$ and $\mathbf{d}^g_{i,l,t}$ denote the residual displacements of source track k and

output track l between frames t and $t+1$. The mean magnitudes used in Eq. 6 are

$$m_i^s = \frac{1}{K_i(T_i-1)} \sum_{k=1}^{K_i} \sum_{t=1}^{T_i-1} \|\mathbf{d}_{i,k,t}^s\|_2,$$
$$m_i^g = \frac{1}{K_i(T_i-1)} \sum_{k=1}^{K_i} \sum_{t=1}^{T_i-1} \|\mathbf{d}_{i,\pi_i(k),t}^g\|_2. \tag{7}$$

A.5 DyCheck iPhone Evaluation

We use seven scenes: apple, block, paper-windmill, space-out, spin, teddy, and wheel. Given the camera parameters from the dataset orientations are converted consistently to camera-to-world poses. The target trajectory follows the held-out static camera. Source and target translations are scaled together so that the 81-frame source path has unit length, preserving the relative camera configuration. Full-frame PSNR and LPIPS compare generated views with the held-out images at corresponding timestamps.

A.6 Additional Constructed Data Examples

Figures 5, 6, and 7 show six examples of constructed real-video reframings. Within each scene, rows show the source or target reframings, while eight columns show synchronized frames beginning with the shared initial frame.

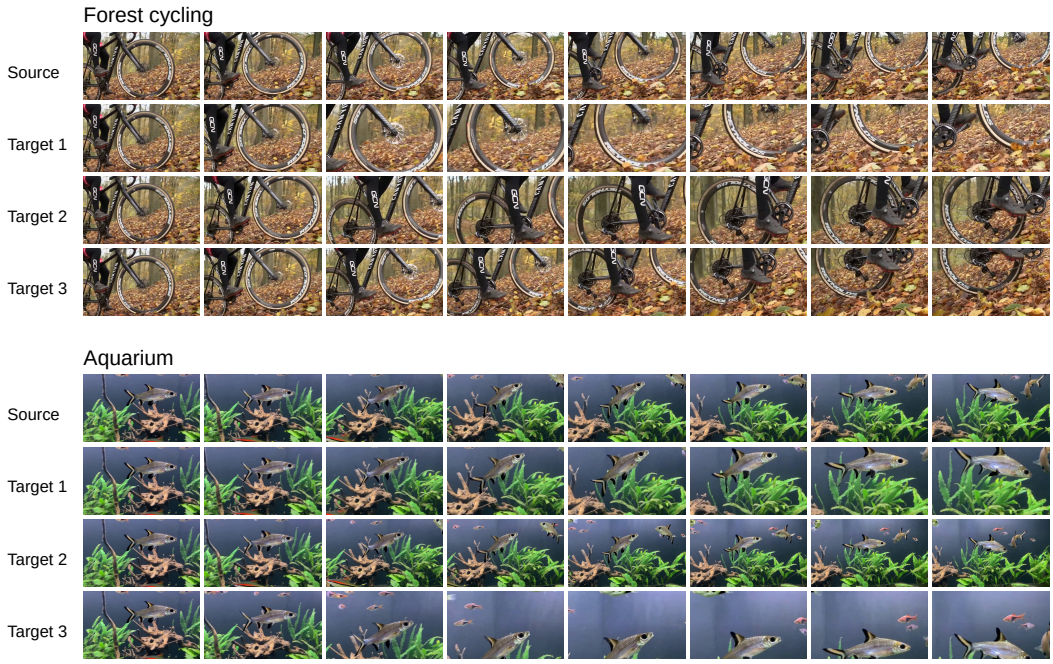

Figure 5: Constructed data examples: forest cycling and aquarium.

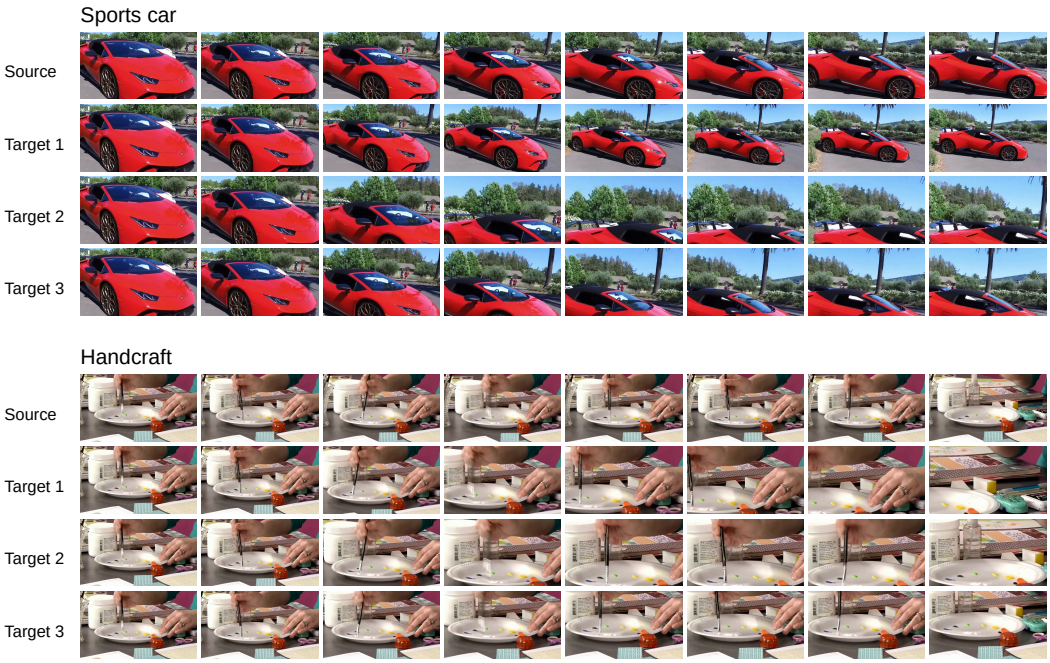

Figure 6: Constructed data examples: sports car and handcraft.

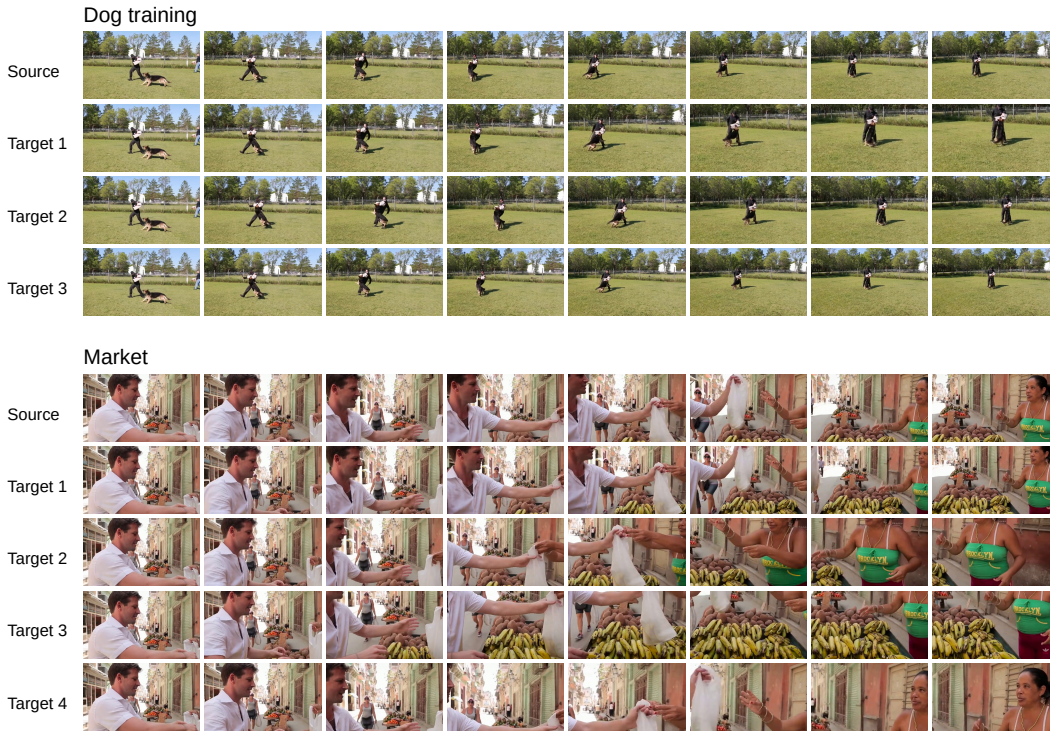

Figure 7: Constructed data examples: dog training and market.