Learning to Correct Geometry in Generated Videos

Weijie Lyu[1] **Tiancheng Shen**[1] **Xiangtai Li**[2] **Yujing Wang**[3]
Xiaoyuan Wang[4] **Haobo Yuan**[1] **Yizhou Zhao**[4] **Ming-Hsuan Yang**[1]

[1]University of California, Merced [2]Nanyang Technological University
[3]Peking University [4]Carnegie Mellon University

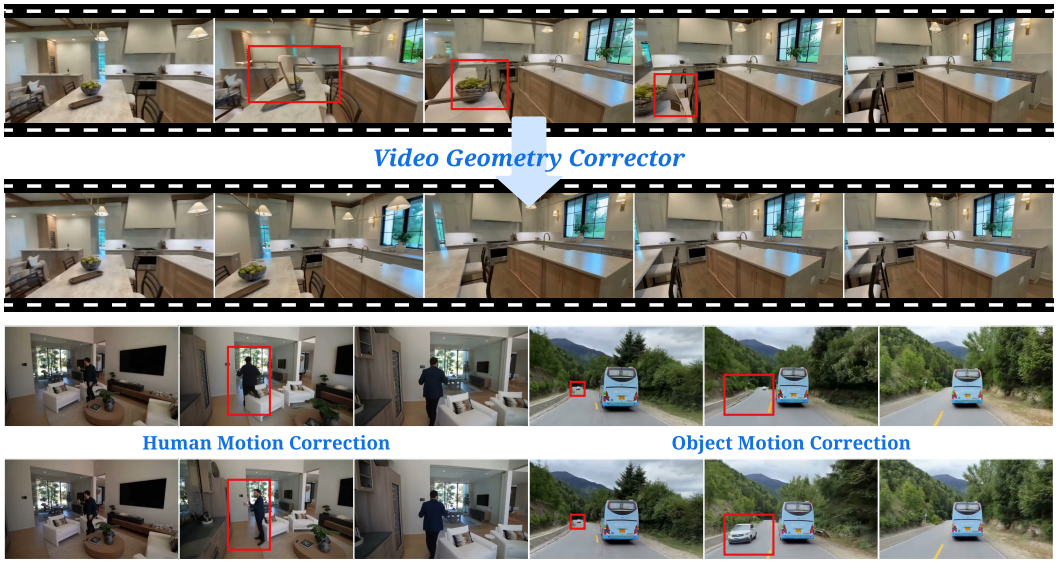

Figure 1: **Video Geometry Corrector** (*VGC*) corrects geometry distortions in generated videos. Under large camera motion, modern video generators often produce warped structures, inconsistent parallax, and drifting scene layouts (red boxes). *VGC* refines these outputs into geometry-consistent videos. When scaled to a larger backbone, it further extends to general video refinement, including dynamic scenes with distorted human or object motion. Project page: https://weijielyu.github.io/VGC.

Abstract

Generated videos often lose geometric consistency under camera motion: structures warp under viewpoint change, parallax becomes inconsistent, and smooth camera movement may collapse into shot-like transitions. We argue that this failure is largely a data problem. In web-scale video corpora, most visible motion comes from dynamic objects captured by static or weakly moving cameras, leaving only sparse supervision for viewpoint-induced appearance changes. To address this, we propose Video Geometry Corrector (*VGC*), a post-hoc video-to-video model trained on synthetic paired supervision. Starting from real videos with diverse camera motion and pose annotations, we construct *VG-Pairs* with a synthetic paired-supervision data generation pipeline. Each video clip is treated as a geometry-consistent target and paired with a distorted counterpart synthesized by a pretrained image-to-video inpainting model using trajectory-adaptive keyframes. *VGC* learns to map the distorted input to the geometry-consistent target without modifying the source generator, and uses optical flow as an auxiliary cue to help disentangle camera motion from object motion. Experiments on pose-conditioned and prompt-driven

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

benchmarks show that *VGC* improves geometric consistency across multiple source generators, and scaling to a larger backbone further extends the framework toward more general video refinement.

1 Introduction

Recent video generation models can synthesize visually compelling clips from text, images, or video conditions [1, 2, 3, 4, 5, 6, 7, 8]. However, their apparent ability to behave as world simulators often breaks down once the camera moves. Rigid structures may warp under viewpoint changes, parallax may become inconsistent, scene layout may drift across frames, and continuous camera motion may collapse into shot-like transitions. These failures are especially severe under large viewpoint changes, showing that current models often generate plausible frame sequences but fail to maintain a stable 3D scene layout. We argue that camera-motion-induced geometric distortion is largely a data problem. In web-scale video corpora, most visible motion comes from dynamic objects captured by static or weakly moving cameras, leaving only sparse supervision for viewpoint-induced appearance changes. As a result, camera motion and object motion are easily entangled: when asked to generate a moving-camera shot, a model may explain the visual change by deforming scene structure rather than maintaining a stable 3D layout.

To address this problem, existing works have largely followed two directions. The first makes the generator more geometry-aware by introducing 3D supervision or spatial priors during training [9, 10, 11, 12, 13]. However, as foundation models evolve rapidly and are often accessible only through checkpoints or inference APIs, modifying the source model is not always feasible. The second direction applies post-hoc video refinement [7, 14, 15]. These methods primarily target pixel-level restoration, such as super-resolution, denoising, and temporal smoothing, thereby improving appearance without explicitly correcting the underlying geometric distortion. Post-hoc geometry correction, which requires reasoning about 3D structure under camera motion, remains comparatively underexplored.

The key bottleneck is the training data. Training a geometry corrector requires paired supervision: a geometry-distorted video and a geometry-consistent target that follows the same camera trajectory. Such pairs are *nearly impossible* to collect in the real world. To address this challenge, we construct the *VG-Pairs* dataset using a paired-supervision data generation pipeline. Starting from real videos with diverse camera motion and annotated poses, we synthesize distorted counterparts using Wan 2.1-Fun-InP, a pretrained image-to-video inpainting model based on Wan [4]. Instead of conditioning only on the first and last frames, we adopt a trajectory-adaptive keyframe strategy that inserts additional keyframes at abrupt pose changes, enabling stable generation of shorter subsegments that are later stitched together. This process produces paired videos with identical semantics and similar camera motion, but different geometric fidelity.

Built on *VG-Pairs*, we propose Video Geometry Corrector (*VGC*), a video-to-video diffusion model that transforms geometry-distorted videos into structurally coherent outputs. *VGC* restores stable scene geometry without regenerating videos from scratch, preserving scene content and object appearance. To further disentangle camera motion from object motion, we incorporate optical flow as an auxiliary structural cue. Optical flow separates globally coherent camera-induced motion from local object dynamics, helping the model determine what to correct and what to preserve.

Although our main focus is correcting geometry in static scenes under camera motion, scaling *VGC* to larger-capacity models naturally extends its capability to dynamic videos containing human and object motion, moving toward general video refinement. This flexibility distinguishes *VGC* from prior geometry-focused approaches [16, 10], which typically focus on static environments.

The main contributions of this work are:

- We identify a critical gap in post-hoc video refinement and attribute camera-induced geometric distortions to insufficient supervision for viewpoint-induced appearance changes.
- We address the paired-data bottleneck with *VG-Pairs*, a synthetic paired-supervision dataset built from real videos with camera motion and their trajectory-aware geometric distortions.
- We propose *VGC*, a post-hoc video-to-video geometry corrector augmented with optical flow cues, demonstrating strong structural restoration, cross-generator transferability, and scalability to dynamic videos.

2 Related Work

Geometry-consistent video generation. Recent video diffusion models such as Sora [1], CogVideoX [2], HunyuanVideo [3], and Wan [4] achieve strong visual quality, yet maintaining geometric consistency under camera motion remains challenging. Most existing approaches address this issue at the generator or training level. Some methods introduce explicit geometric supervision during training. GeoVideo [9] incorporates depth prediction and cross-frame consistency losses. Geometry Forcing [10] aligns intermediate features with pretrained 3D foundation models [17], and Epipolar Geometry Improves Video Generation Models [11] introduces epipolar constraints during fine-tuning. Other approaches improve geometry through post-training alignment or reward optimization. VideoGPA [16] derives preference signals from reprojection consistency, VideoREPA [18] transfers physical reasoning from video foundation models, and VGGRPO [19] extends geometry-aware rewards into latent space for dynamic scenes. A third direction incorporates explicit 3D structure into the generation pipeline. World-consistent Video Diffusion [12] maintains a point-cloud memory for spatial consistency, while GEN3C [13] uses geometric priors for camera-controlled video generation. In contrast, *VGC* frames geometry correction as a post-hoc video-to-video task, avoiding modification of the source generator and enabling transfer across model families.

Video refinement and enhancement. A separate line of work improves generated or degraded videos through post-processing. Upscale-A-Video [7], SeedVR [14], and SeedVR2 [20] focus on restoration and super-resolution, progressing from multi-step diffusion upscaling to single-step arbitrary-resolution enhancement. Meanwhile, several works [8, 21] introduce new datasets to refine high-resolution video generation. VEnhancer [15] jointly improves spatial and temporal quality using a space-time controller on pretrained diffusion priors. Enhance-A-Video [22] enhances temporal consistency in DiT-based models without retraining. ReVision [23] refines coarse videos using 3D object-centric representations and physical motion priors. Despite improving visual quality, temporal smoothness, and motion plausibility, these methods largely overlook post-hoc correction of scene-level geometric consistency under camera motion. *VGC* differs by targeting geometry correction, using paired supervision constructed from real videos to repair structural inconsistency while preserving scene content and motion.

3 Method

Our goal is to restore geometric consistency in generated videos under camera motion. We attribute these failures to insufficient supervision for viewpoint-induced appearance change in existing video data. Instead of modifying each video generation model to learn stronger geometric priors, we formulate geometry correction as a post-hoc video-to-video task and train a separate corrector with paired supervision. This decouples geometry correction from video generator design, making our method applicable across different model families.

Our framework consists of two stages. First, we build *VG-Pairs* by synthesizing paired distorted inputs and geometry-consistent targets from real videos with diverse camera motion and pose annotations. Second, we train a video-to-video diffusion model, *VGC*, on these pairs to correct geometry. Optical flow extracted from the distorted input serves as an auxiliary cue for disentangling camera motion from scene dynamics.

3.1 Paired Supervision from Videos with Camera Motion

The central design choice is to use videos with diverse camera motion not merely as additional training data for a generator, but as a source of correction pairs for a video-to-video task. Hence, we construct *VG-Pairs*, a dataset of geometry-distorted videos paired with geometry-consistent supervision, using the following data generation pipeline. Let $\mathbf{x} = (x_1, \ldots, x_T)$ denote a real video clip with camera poses $\mathbf{p} = (p_1, \ldots, p_T)$, where each pose is represented as $p_i = (\mathbf{t}_i, \mathbf{q}_i)$ with camera position $\mathbf{t}_i \in \mathbb{R}^3$ and rotation $\mathbf{q}_i \in S^3$. We treat $\mathbf{x}$ as the geometry-consistent target and synthesize a distorted counterpart, $\mathbf{y}$, using Wan 2.1-Fun-InP, a pretrained image-to-video inpainting model based on Wan [4]. Given a start frame and an end frame, this model generates the intermediate video that connects them.

To make $\mathbf{y}$ useful for supervision, it should preserve the scene identity and approximately follow the camera trajectory of $\mathbf{x}$ while still exhibiting geometric errors. We therefore select a trajectory-adaptive

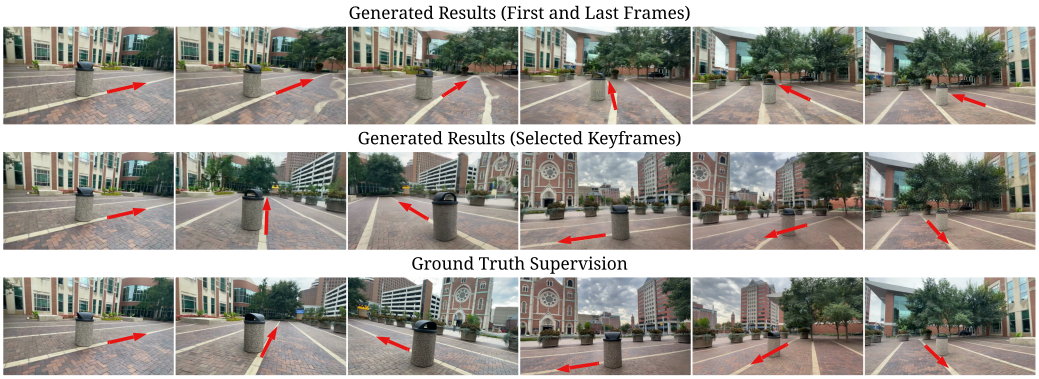

Figure 2: **Effect of adaptive keyframe selection.** The red arrows indicate a constant 3D direction while the camera is moving. For clips with large, nonlinear camera motion, conditioning the image-to-video model only on the first and last frames often produces a shortcut trajectory that deviates substantially from the ground-truth path (top; about 45° rotation versus about 225° in the ground truth). Our adaptive keyframe selection inserts intermediate keyframes, generates shorter segments conditioned on their boundary keyframes, and stitches them together. This produces geometry-distorted videos that remain close to the target camera trajectory and are therefore suitable as paired training data (middle), compared with the ground-truth supervision video (bottom).

set of keyframes $K = \{k_1, \dots, k_{m+1}\}$ with $1 = k_1 < \cdots < k_{m+1} = T$, and generate one video segment between each consecutive keyframe pair. Figure 2 illustrates the motivation: for clips with large, nonlinear camera motion, conditioning only on the first and last frames can lead the inpainting model to produce a shortcut trajectory that deviates from the ground-truth path, whereas adaptive keyframe selection yields distorted videos that remain close to the target trajectory.

We first estimate the total amount of camera motion in the clip. For each adjacent frame pair, we measure rotational change by the geodesic distance between quaternions, $\Delta\phi_i = 2\arccos(|\langle \mathbf{q}_i, \mathbf{q}_{i+1}\rangle|)$, and translational change by the Euclidean displacement $\|\mathbf{t}_{i+1} - \mathbf{t}_i\|_2$. Summing these local changes over the full clip gives the total rotation and translation:

$$R_{\text{tot}} = \sum_{i=1}^{T-1} \Delta\phi_i, \qquad D_{\text{tot}} = \sum_{i=1}^{T-1} \|\mathbf{t}_{i+1} - \mathbf{t}_i\|_2. \tag{1}$$

We then choose how many trajectory segments are needed so that each segment covers only a manageable amount of motion. Concretely, R_{seg} and D_{seg} denote the target rotation and translation budget for a single segment. We compare the total motion of the clip against these budgets along both axes and let the larger normalized motion determine the number of segments, capped at $M_{\max}$:

$$m = \min\left(M_{\max}, \max\left(1, \left\lceil \max\left(\frac{R_{\text{tot}}}{R_{\text{seg}}}, \frac{D_{\text{tot}}}{D_{\text{seg}}}\right)\right\rceil\right)\right). \tag{2}$$

If necessary, we further reduce m so that each segment contains at least $L_{\min}$ frames. As a result, short or nearly linear trajectories are handled with a single segment, whereas clips with larger accumulated motion are divided into shorter intervals that are easier for the inpainting model to follow.

Once m is fixed, we place the $m - 1$ interior keyframes at moments of peak trajectory inflection, the exact points where the generator is most prone to geometric drift. To capture such changes, we compute the local translation and rotation increments, $\mathbf{v}_i = \mathbf{t}_{i+1} - \mathbf{t}_i$ and $\boldsymbol{\omega}_i = \log(\mathbf{q}_{i+1}\mathbf{q}_i^{-1})$, and assign each interior frame a motion-change score:

$$s(i) = \theta(\mathbf{v}_{i-1}, \mathbf{v}_i) + \alpha\,\theta(\boldsymbol{\omega}_{i-1}, \boldsymbol{\omega}_i), \tag{3}$$

where $\theta(\cdot, \cdot)$ denotes the angle between two vectors, and each term is suppressed when the corresponding motion magnitude is below a small threshold. A large score indicates a sharp change in translation direction, rotation direction, or both. We select up to $m - 1$ interior keyframes by greedy non-maximum suppression over $s(i)$, enforcing a minimum temporal separation of $L_{\min}$. If fewer than $m - 1$ reliable peaks are selected, we place the remaining keyframes by approximately uniform partition of the largest uncovered gaps.

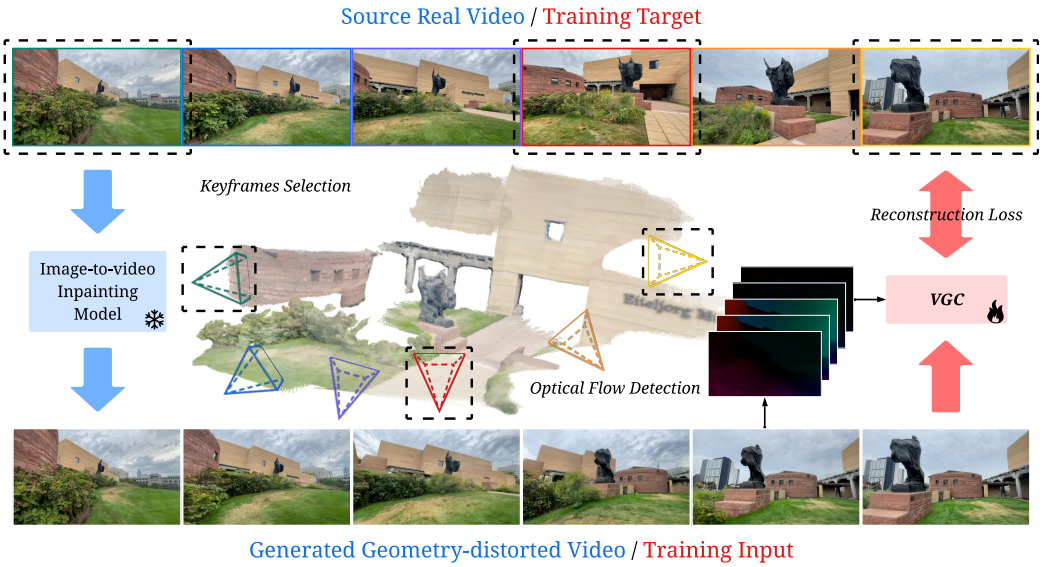

Figure 3: **Overview of Video Geometry Corrector (*VGC*).** Starting from a real video with diverse camera motion, we select trajectory-adaptive keyframes and use them to condition an image-to-video inpainting model, which produces a geometry-distorted video that serves as the training input. We then train *VGC* to map this distorted input to the geometry-consistent target video with a flow-matching objective. Optical flow is computed from the distorted video and used as an additional conditioning signal for *VGC*, providing motion cues that help disentangle camera motion from scene dynamics. *The 3D reconstruction is shown here only for visualization and is not used in the pipeline.*

Given the final keyframe set K, we generate each segment independently by conditioning the inpainting model on the boundary frames x_{k_j} and $x_{k_{j+1}}$. Concatenating the resulting subclips and removing duplicated boundary frames yields the distorted counterpart $\mathbf{y}$. This adaptive construction keeps the synthesized video close enough to the target that the learning problem remains correction rather than regeneration, while still inducing the geometric failures of interest, such as structural warping, inconsistent parallax, depth-order violations, and shot-like transitions.

3.2 Motion Disentanglement with Optical Flow

Geometry distortions under camera motion arise in part because current video training data contains rich human and object motion, but relatively sparse supervision for viewpoint-induced appearance change. As a result, video models can confuse camera motion with scene dynamics, and may explain viewpoint change by deforming scene structure instead of preserving a stable 3D layout. Geometry correction is therefore not only a restoration problem, but also a motion disentanglement problem.

Optical flow provides a natural auxiliary representation for this purpose. Unlike RGB frames, which entangle appearance and motion, optical flow explicitly represents motion through its location, direction, and magnitude. Camera motion tends to induce a globally coherent motion field, whereas object motion appears as local deviations from that field. We therefore compute forward optical flow between consecutive frames of the distorted input video and use it as an additional conditioning signal for the correction model. Because the flow is computed from the distorted input rather than the target, the same signal is available during both training and inference. Optical flow does not supervise the model directly; instead, it provides motion structure that helps distinguish what should be corrected from what should be preserved.

3.3 Video Geometry Corrector

Figure 3 summarizes the overall data generation and training pipeline. *VGC* is built on a pretrained video diffusion backbone [4] and trained to map a distorted input video $\mathbf{y}$ to its geometry-consistent target $\mathbf{x}$. Let $E(\cdot)$ denote the video encoder, and let $\mathbf{z} = E(\mathbf{x})$ be the latent representation of the target video. Following the flow-matching formulation of Wan [4], during training we sample a

Table 1: **Quantitative results on RealEstate10K [24].** Each source generator is followed by its *VGC*-corrected counterpart, with changes relative to the source shown in parentheses. *VGC* improves PSNR, SSIM, and revisit error (RVE) over DFoT [25], Geometry Forcing (GF) [10], and GF + REPA [10, 26], showing that it can be flexibly combined with geometry-aware generators. Best results are in bold. Comparing *VGC* with and without optical flow shows that flow yields modest gains in image quality while maintaining similar long-range geometric consistency.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	RVE↓
DFoT [25]	13.61	0.4006	0.4487	184.2
DFoT + *VGC* (w/o Flow)	14.76 (+1.15)	0.4509 (+0.0503)	0.4433 (-0.0054)	169.4 (-14.8)
DFoT + *VGC* (Ours)	14.87 (+1.26)	0.4545 (+0.0539)	0.4395 (-0.0092)	168.9 (-15.3)
GF [10]	14.76	0.4484	**0.4127**	181.0
GF + *VGC*	**15.50** (+0.74)	**0.4715** (+0.0232)	0.4150 (+0.0022)	**158.8** (-22.2)
GF [10] + REPA [26]	14.16	0.4217	0.4218	171.8
GF + REPA + *VGC*	15.25 (+1.09)	0.4675 (+0.0458)	0.4226 (+0.0008)	160.9 (-10.9)

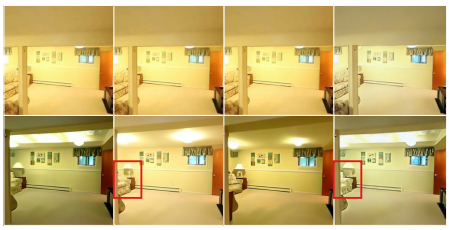

Figure 4: **Visual results on RealEstate10K [24].** DFoT [25] and GF + REPA [10, 26] produce inconsistent lighting across frames, while GF [10] generates an incomplete sofa structure (red box). *VGC* restores more consistent lighting and more coherent scene geometry.

timestep $t \in [0, 1]$ and Gaussian noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and construct the noisy latent $\mathbf{z}_t = (1-t)\,\mathbf{z} + t\,\epsilon$. The network is trained to predict the velocity $\mathbf{u} = \epsilon - \mathbf{z}$ conditioned on the distorted input video, the clip text prompt, and the optical flow maps:

$$\mathcal{L} = \mathbb{E}_{t,\mathbf{z},\epsilon}\left[\|\mathbf{u} - \mathbf{u}_\theta(\mathbf{z}_t, t, \mathbf{c})\|_2^2\right], \tag{4}$$

where $\mathbf{c}$ denotes the distorted-video latent, the text condition, and the optical flow condition. We inject the video and flow conditions through channel concatenation, while the text prompt is provided through cross-attention. The model is thus trained to preserve scene content and camera intent while denoising toward a geometry-consistent target. At inference, *VGC* takes a distorted input video, computes optical flow from it, and performs conditional denoising to produce a geometry-corrected output. Because the model is trained as a separate video-to-video corrector, it can refine videos from different generators without modifying the source model itself.

4 Experiments

4.1 Experimental Setup

Implementation details. We construct *VG-Pairs* from two publicly available datasets, DL3DV [27] and SpatialVID [28]. We select 9K videos from DL3DV, reserve the first 1K for evaluation, and filter 63K videos from SpatialVID that contain large camera motion. To construct paired supervision, we use Wan 2.1-Fun-InP to synthesize geometry-distorted videos from trajectory-adaptive keyframes. Given a start frame and an end frame, Wan 2.1-Fun-InP generates the intermediate video connecting them. For clips with complex camera motion, we apply this model independently to multiple keyframe-bounded subsegments and stitch the generated segments together to form the distorted training input. Because Wan's VAE temporally compresses the video, keyframes are selected so that each generated subsegment has a frame count compatible with the temporal compression schedule. We use Wan 2.1-1.3B [4] as the backbone of *VGC*. The model is trained for 10K steps on 32 H100 GPUs with AdamW, using a learning rate of 1×10^{-5}, weight decay 0.01, and a global batch size of 64. Unless otherwise specified, all training videos contain 81 frames at a resolution of 832×480. Optical flow is estimated with SEA-RAFT [29] and used as an additional conditioning signal. We use $M_{\max} = 4$, $R_{\text{seg}} = 120°$, $D_{\text{seg}} = 10\,\text{m}$, $L_{\min} = 15$, and $\alpha = 1$. Additional details on adaptive keyframe selection, segment construction, and the training recipe are provided in the supplementary material.

Evaluation datasets and metrics. To evaluate *VGC*'s performance, we study two scenarios in which camera motion is introduced. In the first setting, camera motion is specified by camera poses. Following Geometry Forcing [10], we evaluate on RealEstate10K [24] using 100 randomly sampled test videos and DFoT [25] as the source generator. We compare against Geometry Forcing and Geometry Forcing + REPA [26], and report PSNR, SSIM, LPIPS [30], and revisit error (RVE). Following [10], RVE measures long-range consistency under a full camera rotation by comparing the generated view at the end of the trajectory with the corresponding reference view. In the second

Table 2: **Quantitative results on DL3DV [27].** We evaluate *VGC* on three CogVideoX [2] source generators in both text-to-video and image-to-video settings. *VGC* improves PSNR, SSIM, and MSE over the corresponding baselines, and further improves VideoGPA [16] when applied as a post-hoc corrector. These results show that *VGC* transfers across different generation settings and can be flexibly combined with other geometry-aware refinement methods.

Method	*CogVideoX-5B T2V*					*CogVideoX 1.5-5B T2V*					*CogVideoX-5B I2V*				
	PSNR↑	SSIM↑	LPIPS↓	MSE↓	3DCS↓	PSNR↑	SSIM↑	LPIPS↓	MSE↓	3DCS↓	PSNR↑	SSIM↑	LPIPS↓	MSE↓	3DCS↓
Baseline	15.6668	0.5389	0.5285	0.0333	0.8617	15.9062	0.4644	0.5449	0.0289	0.8342	17.3189	0.5543	0.5760	0.0218	0.7944
Baseline + *VGC*	16.6940 (+1.0272)	0.5519 (+0.0130)	0.5733 (+0.0448)	0.0264 (-0.0069)	0.8376 (-0.0241)	17.7342 (+1.8280)	0.5284 (+0.0640)	0.5936 (+0.0487)	0.0192 (-0.0097)	0.7859 (-0.0483)	17.9989 (+0.6800)	0.5456 (-0.0087)	0.6284 (+0.0524)	0.0184 (-0.0034)	0.8121 (+0.0177)
VideoGPA [16]	14.6318	0.5210	0.5065	0.0428	0.9346	14.4002	0.4175	0.5531	0.0408	0.9614	15.7203	0.5240	0.5749	0.0320	0.8952
VideoGPA [16] + *VGC*	15.9092 (+1.2774)	0.5413 (+0.0203)	0.5546 (+0.0481)	0.0319 (-0.0109)	0.8737 (-0.0609)	16.8430 (+2.4428)	0.5039 (+0.0864)	0.5907 (+0.0376)	0.0235 (-0.0173)	0.8258 (-0.1356)	17.3053 (+1.5850)	0.5379 (+0.0139)	0.6216 (+0.0467)	0.0225 (-0.0095)	0.8463 (-0.0489)

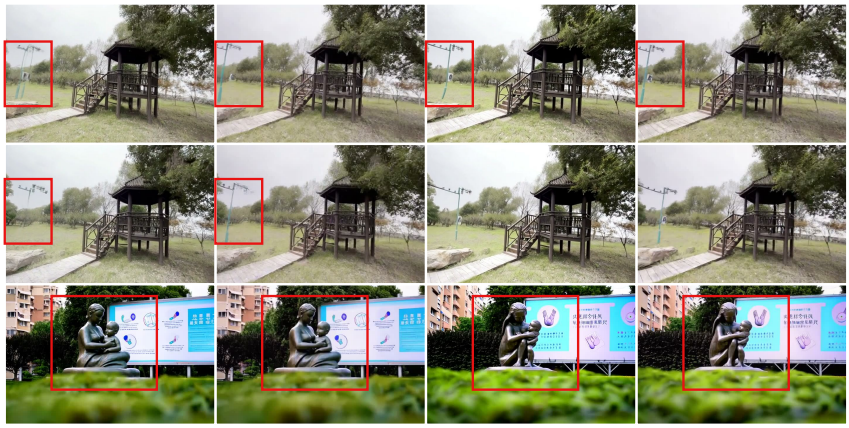

Figure 5: **Visual results on DL3DV [27].** The first two rows show image-to-video results. In the baseline CogVideoX [2] outputs, the streetlight is duplicated, while VideoGPA [16] produces a distorted and tilted streetlight (red boxes). *VGC* corrects these geometric errors and restores a more plausible structure. The last row shows a text-to-video example, where *VGC* better preserves the original appearance of the sculpture, while VideoGPA changes its structure and other scene details.

setting, camera motion is induced by the text prompt. We use three source generators in this setting: CogVideoX-5B I2V, CogVideoX-5B T2V, and CogVideoX 1.5-5B T2V. We randomly sample 200 prompts from the held-out DL3DV test set. Following VideoGPA [16], we reconstruct each generated video independently using Depth Anything 3 [31] and compare its original frames with frames reprojected from the reconstructed 3D geometry to compute PSNR, SSIM, LPIPS [30], MSE, and 3D consistency score (3DCS). Unlike the pose-conditioned setting, these metrics do not require ground-truth video frames. In both settings, *VGC* is applied as a post-hoc corrector to the outputs of the source generator, demonstrating its flexibility across different video model families.

4.2 Experimental Results

Results on RealEstate10K. Table 1 shows that *VGC* improves geometric consistency in the pose-conditioned setting and achieves stronger PSNR, SSIM, and RVE than the DFoT baseline [25]. When applied as a post-hoc corrector to the outputs of Geometry Forcing [10] and Geometry Forcing + REPA [10, 26], *VGC* further improves their results, demonstrating that it can be flexibly combined with geometry-aware generators. Figure 4 provides qualitative examples of the improved temporal consistency and scene structure. Overall, these results support our claim that *VGC* improves both local visual quality and cross-frame geometric consistency under camera motion.

We also conduct an ablation study in this setting to evaluate the effect of optical flow as an auxiliary conditioning signal. Comparing the full model with the variant without flow in Table 1, we find that optical flow improves PSNR, SSIM, and RVE. These results suggest that flow helps improve local structural consistency, and that *VGC*'s main gains come from the paired supervision itself.

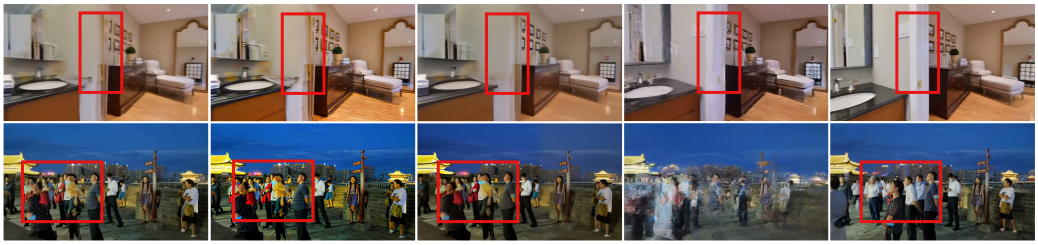

Figure 6: **Comparison of model sizes.** In static scenes (top), both *VGC* 1.3B and 14B correct geometric distortions produced by the baseline model that SeedVR2 [20] and VEnhancer [15] fail to restore (red boxes). The 14B model further improves the geometry of moving humans and objects in dynamic scenes (bottom).

Table 3: **General video refinement results on SpatialVID [28].** We compare *VGC* with general video refinement and enhancement methods [20, 15] on dynamic videos generated by CogVideoX [2], HunyuanVideo [3], and Wan [4]. Although *VGC* is designed for geometry correction under camera motion rather than generic enhancement, it achieves competitive performance across multiple perceptual and temporal metrics, and is strong in dynamic degree, imaging quality, and DOVER [32]. These results suggest that the proposed correction framework extends beyond static-scene geometry and remains effective in more general settings.

Method	Subject Consistency	Background Consistency	Temporal Flickering	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Imaging Quality	DOVER [32]
			CogVideoX-I2V [2]					
Baseline	0.9480	0.9455	0.9742	0.9918	0.30	0.5095	69.6945	9.54
SeedVR2 [20]	0.9447	0.9476	0.9603	0.9829	0.30	**0.5313**	71.1767	8.31
VEnhancer v2 [15]	**0.9506**	**0.9602**	0.9749	**0.9920**	0.31	0.5237	69.3742	10.31
VGC (Ours)	0.9483	0.9450	**0.9772**	0.9891	**0.66**	0.5011	**72.7654**	**10.50**
			HunyuanVideo 1.5-I2V [3]					
Baseline	0.8970	0.9231	0.9592	0.9829	**1.00**	0.5133	73.7167	**10.10**
SeedVR2 [20]	0.8967	0.9167	0.9415	0.9714	**1.00**	**0.5539**	72.2762	7.89
VEnhancer v2 [15]	**0.9061**	**0.9294**	**0.9596**	**0.9860**	0.99	0.5301	71.9688	9.85
VGC (Ours)	0.8995	0.9282	0.9482	0.9799	0.96	0.4849	**75.1367**	9.30
			Wan 2.1-Fun-InP [4]					
Baseline	0.9247	0.9239	**0.9712**	**0.9914**	0.87	0.5043	69.5165	10.87
SeedVR2 [20]	0.9217	0.9231	0.9547	0.9809	0.87	**0.5431**	**72.9463**	8.23
VEnhancer v2 [15]	0.9242	**0.9371**	0.9702	0.9909	0.85	0.5254	70.6322	10.78
VGC (Ours)	**0.9268**	0.9281	0.9641	0.9907	**0.89**	0.5141	72.6499	**11.47**

Results on DL3DV. Table 2 shows that *VGC* improves reconstruction quality and geometric consistency across all three CogVideoX source generators in both text-to-video and image-to-video settings. Compared with the original baselines, *VGC* improves PSNR, SSIM, and MSE, indicating better frame fidelity under prompt-driven camera motion. When applied on top of VideoGPA [16], *VGC* further improves its results, showing that the proposed corrector can be flexibly combined with geometry-aware refinement methods. Figure 5 provides qualitative examples in which *VGC* corrects geometric distortions while better preserving scene appearance.

4.3 Extension to General Video Refinement

We further investigate whether the proposed correction framework can extend beyond geometry correction in static scenes. To this end, we train a larger version of *VGC* based on Wan 2.2-14B and evaluate it in a more general video refinement setting. With increased model capacity, *VGC* not only corrects geometric distortions under camera motion, but also improves dynamic videos by recovering more plausible human and object geometry. As shown in Figure 6, the smaller 1.3B version does not exhibit the same ability to recover dynamic motion. Because the 14B model takes 480p videos as input and produces 720p outputs, it also performs super-resolution as part of the refinement process.

For evaluation, we randomly sample 100 held-out videos from SpatialVID [28] and compare against general video refinement methods, including SeedVR2 [20] and VEnhancer v2 [15]. To demonstrate the flexibility of *VGC*, we apply it to videos generated by three different source models:

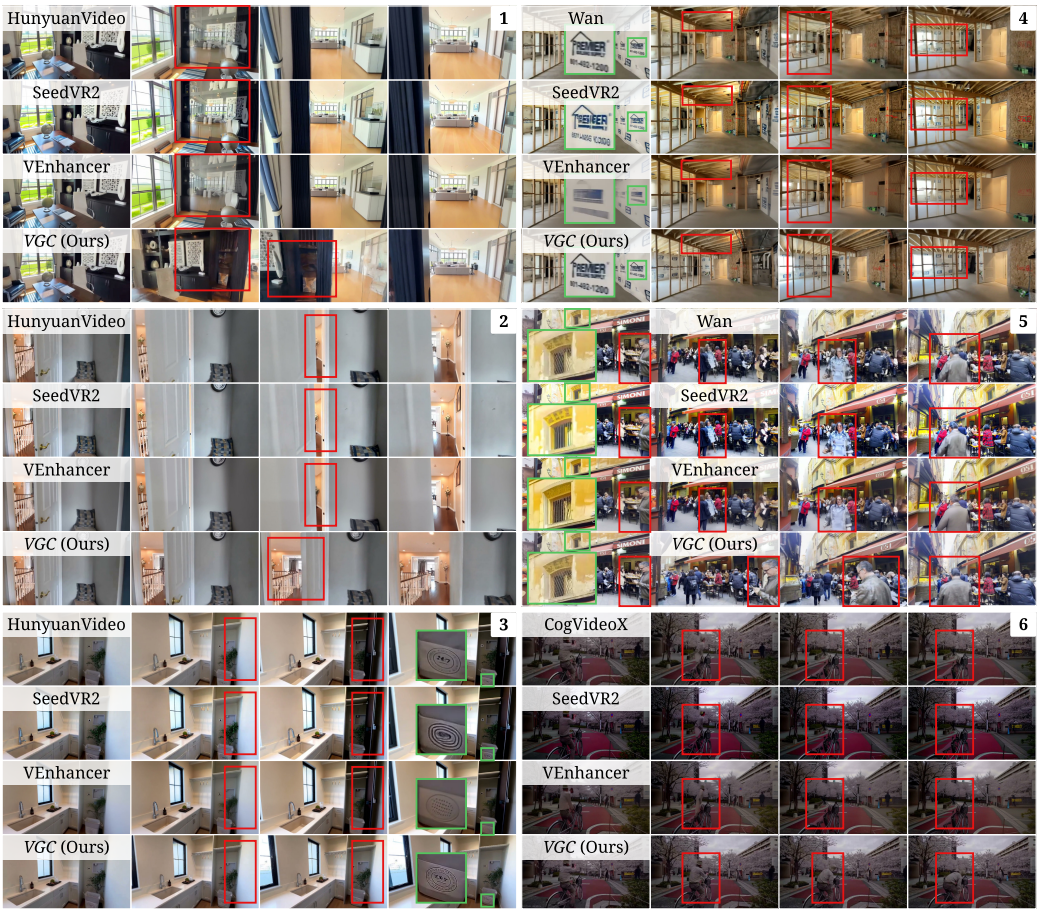

Figure 7: **General video refinement results.** *VGC* corrects several geometric failures produced by the baseline models. In scene 1, HunyuanVideo [3] changes the cabinet into a glass door, while *VGC* restores the correct scene structure and camera motion, placing the door on the right side of the cabinet. In scene 2, HunyuanVideo generates a camera trajectory that passes through the wrong side of the doorway, whereas *VGC* recovers a plausible path. In scenes 3 and 4, *VGC* restores a consistent appearance and recovers fine geometric structure, such as the wooden supports. In the dynamic scenes, *VGC* better preserves identity and motion continuity: in scene 5, it maintains the same man in the leather jacket across frames and recovers a plausible motion trajectory, whereas the Wan baseline fails to infer coherent human motion; in scene 6, it restores the missing cyclist produced by CogVideoX [2]. The green boxes further show that *VGC* enhances local detail while remaining more faithful to the source video than SeedVR2 [20] and VEnhancer [15]. *Please zoom in for details.*

CogVideoX [2], HunyuanVideo 1.5 [3], and Wan 2.1 [4]. CogVideoX and HunyuanVideo 1.5 are evaluated in the image-to-video setting, while Wan 2.1 is evaluated in the inpainting setting. Since these videos contain dynamic content and do not admit a geometry-consistent reconstruction target, we evaluate them with no-reference video-quality metrics rather than reconstruction metrics. Specifically, we report VBench [33] metrics together with DOVER [32], a holistic no-reference video-quality metric that summarizes technical and aesthetic quality.

The quantitative results in Table 3 show that *VGC* achieves performance comparable to strong video refinement baselines across a range of perceptual and temporal metrics, despite not being explicitly designed for generic enhancement. In particular, it remains competitive in subject consistency, background consistency, and motion quality, while achieving strong results in dynamic degree, imaging quality, and DOVER. These results suggest that, when scaled to larger backbones, the proposed correction framework naturally generalizes to broader video refinement settings.

Figure 7 provides qualitative examples. *VGC* corrects implausible geometry changes, recovers more plausible camera motion, restores distorted structural details, and better preserves human identity and

motion continuity. It also improves local detail through super-resolution while remaining faithful to the source video.

5 Conclusion

We presented Video Geometry Corrector (*VGC*), a post-hoc video-to-video model for correcting geometric distortions in generated videos under camera motion. We argued that these failures are largely a data problem: current video training data provides limited supervision for viewpoint-induced appearance change, causing camera motion to be entangled with scene dynamics. To address this, we constructed *VG-Pairs* from real videos with diverse camera motion and trained *VGC* to map geometry-distorted videos to geometry-consistent targets. We further incorporated optical flow as an auxiliary cue to help distinguish what should be corrected from what should be preserved. Experiments on pose-conditioned and prompt-driven benchmarks show that *VGC* improves geometric consistency across different source generators and can be flexibly combined with existing geometry-aware methods. With increased model capacity, the same correction framework also extends beyond static-scene geometry toward more general video refinement.

Acknowledgments and Disclosure of Funding

This work was supported in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean Government (MSIT) (No. RS-2024-00457882, National AI Research Lab Project) and a Sony Focused Research Award. The authors declare no competing interests.

References

[1] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Leo Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 2024. 2, 3

[2] 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. In *ICLR*, 2025. 2, 3, 7, 8, 9, 15, 16, 17, 20

[3] Tencent Hunyuan Foundation Model Team. Hunyuanvideo 1.5 technical report, 2025. 2, 3, 8, 9, 20

[4] 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. 2, 3, 5, 6, 8, 9, 14, 20

[5] Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026. 2

[6] Jianzong Wu, Xiangtai Li, Yanhong Zeng, Jiangning Zhang, Qianyu Zhou, Yining Li, Kai Chen, and Yunhai Tong. Motionbooth: Motion-aware customized text-to-video generation. In *NeurIPS*, 2024. 2

[7] Shangchen Zhou, Peiqing Yang, Jianyi Wang, Yihang Luo, and Chen Change Loy. Upscale-A-Video: Temporal-consistent diffusion model for real-world video super-resolution. In *CVPR*, 2024. 2, 3

[8] Jiangning Zhang, Junwei Zhu, Teng Hu, Yabiao Wang, Donghao Luo, Weijian Cao, Zhenye Gan, Xiaobin Hu, Zhucun Xue, Xiangtai Li, Chengjie Wang, and Yong Liu. Transform trained transformer: Accelerating naive 4k video generation over 10 times. In *ICML*, 2026. 2, 3

[9] Yunpeng Bai, Shaoheng Fang, Chaohui Yu, Fan Wang, and Qixing Huang. Geovideo: Introducing geometric regularization into video generation model. *arXiv preprint arXiv:2512.03453*, 2025. 2, 3

[10] Haoyu Wu, Diankun Wu, Tianyu He, Junliang Guo, Yang Ye, Yueqi Duan, and Jiang Bian. Geometry Forcing: Marrying video diffusion and 3D representation for consistent world modeling. In *ICLR*, 2026. 2, 3, 6, 7, 15, 16, 20

[11] Orest Kupyn, Fabian Manhardt, Federico Tombari, and Christian Rupprecht. Epipolar geometry improves video generation models. *arXiv preprint arXiv:2510.21615*, 2025. 2, 3

[12] Qihang Zhang, Shuangfei Zhai, Miguel Ángel Bautista, Kevin Miao, Alexander Toshev, Joshua Susskind, and Jiatao Gu. World-consistent video diffusion with explicit 3d modeling. In *CVPR*, 2025. 2, 3

[13] 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 *CVPR*, 2025. 2, 3

[14] Jianyi Wang, Zhijie Lin, Meng Wei, Yang Zhao, Ceyuan Yang, Chen Change Loy, and Lu Jiang. Seedvr: Seeding infinity in diffusion transformer towards generic video restoration. In *CVPR*, 2025. 2, 3

[15] Jingwen He, Tianfan Xue, Dongyang Liu, Xinqi Lin, Peng Gao, Dahua Lin, Yu Qiao, Wanli Ouyang, and Ziwei Liu. VEnhancer: Generative space-time enhancement for video generation. *arXiv preprint arXiv:2407.07667*, 2024. 2, 3, 8, 9, 18, 20

[16] Hongyang Du, Junjie Ye, Xiaoyan Cong, Runhao Li, Jingcheng Ni, Aman Agarwal, Zeqi Zhou, Zekun Li, Randall Balestriero, and Yue Wang. VideoGPA: Distilling geometry priors for 3D-consistent video generation. *arXiv preprint arXiv:2601.23286*, 2026. 2, 3, 7, 8, 15, 16, 20

[17] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *CVPR*, 2025. 3

[18] Xiangdong Zhang, Jiaqi Liao, Shaofeng Zhang, Fanqing Meng, Xiangpeng Wan, Junchi Yan, and Yu Cheng. VideoREPA: Learning physics for video generation through relational alignment with foundation models. *arXiv preprint arXiv:2505.23656*, 2025. 3

[19] Zhaochong An, Orest Kupyn, Théo Uscidda, Andrea Colaco, Karan Ahuja, Serge J. Belongie, Mar González-Franco, and Marta Tintore Gazulla. VGGRPO: towards world-consistent video generation with 4d latent reward. In Paolo Favaro, Zuzana Kukelova, Atsuto Maki, Anna Rohrbach, Konrad Schindler, and Federico Tombari, editors, *ECCV*, 2026. 3

[20] Jianyi Wang, Shanchuan Lin, Zhijie Lin, Yuxi Ren, Meng Wei, Zongsheng Yue, Shangchen Zhou, Hao Chen, Yang Zhao, Ceyuan Yang, et al. Seedvr2: One-step video restoration via diffusion adversarial post-training. *arXiv preprint arXiv:2506.05301*, 2025. 3, 8, 9, 18, 20

[21] Zhucun Xue, Jiangning Zhang, Teng Hu, Haoyang He, Yinan Chen, Yuxuan Cai, Yabiao Wang, Chengjie Wang, Yong Liu, Xiangtai Li, et al. Ultravideo: High-quality uhd video dataset with comprehensive captions. In *NeurIPS*, 2025. 3

[22] Yang Luo, Xuanlei Zhao, Mengzhao Chen, Kaipeng Zhang, Wenqi Shao, Kai Wang, Zhangyang Wang, and Yang You. Enhance-A-Video: Better generated video for free. *arXiv preprint arXiv:2502.07508*, 2025. 3

[23] Qihao Liu, Ju He, Qihang Yu, Liang-Chieh Chen, and Alan Yuille. ReVision: Refining video diffusion with explicit 3D motion modeling. *arXiv preprint arXiv:2504.21855*, 2025. 3

[24] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 6, 15, 20

[25] Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. In *ICML*, 2025. 6, 7, 15, 20

[26] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *ICLR*, 2025. 6, 7, 20

[27] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. DL3DV-10K: A large-scale scene dataset for deep learning-based 3D vision. In *CVPR*, 2024. 6, 7, 14, 15, 17, 20

[28] Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Yi Zhang, Chang Zeng, Yanxi Zhou, et al. Spatialvid: A large-scale video dataset with spatial annotations. *arXiv preprint arXiv:2509.09676*, 2025. 6, 8, 14, 15, 20

[29] Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *ECCV*, 2024. 6, 14, 20

[30] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6, 7, 16

[31] Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In *ICLR*, 2026. 7, 17

[32] Haoning Wu, Erli Zhang, Liang Liao, Chaofeng Chen, Jingwen Hou, Annan Wang, Wenxiu Sun, Qiong Yan, and Weisi Lin. Exploring video quality assessment on user generated contents from aesthetic and technical perspectives. In *ICCV*, 2023. 8, 9, 19

[33] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024. 9, 19

[34] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 14

[35] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. 16

Supplementary Material

A Overview

In this supplementary material, we provide additional visualizations of *VG-Pairs* and additional experimental results for *VGC* in Section B. We then present implementation details in Section C, compare *VGC* with direct generator fine-tuning in Section D, discuss limitations and broader impacts in Sections E and F, and summarize asset licenses and terms of use in Section G.

B Additional Visualizations

B.1 Samples of *VG-Pairs*

Figure 8 shows examples from *VG-Pairs*, including two samples constructed from DL3DV [27] and two from SpatialVID [28]. Each example contains a geometry-distorted video and its paired ground-truth target. DL3DV provides videos with diverse camera trajectories in mostly static scenes, while SpatialVID contributes dynamic videos with human and object motion. Together, they enable *VGC* to learn both geometry correction under camera motion and refinement of dynamic structure.

B.2 Experimental Results

We show additional qualitative results on general video refinement. Figure 9 and Figure 10 show examples where *VGC* refines videos with geometry distortions, inconsistent appearance, and dynamic motion errors caused by camera movement. These examples further illustrate that, when scaled to a larger model, *VGC* can improve both static-scene geometry and dynamic human or object motion. Figure 11 shows additional video super-resolution results. Because the 14B version of *VGC* takes 480p videos as input and produces 720p outputs, it can also enhance fine details through super-resolution while remaining faithful to the source video and avoiding hallucinated structures.

C Implementation Details

Training details. *VGC* is built on Wan 2.1-1.3B and Wan 2.2-14B [4]. We fully fine-tune the DiT [34] backbone while keeping the VAE and text encoder frozen. Both models are trained using AdamW with a learning rate of 1×10^{-5} and a global batch size of 64. Optical flow is estimated using SEA-RAFT [29] from the distorted input videos and used during both training and inference. Additional training configurations are summarized in Table 4.

Paired-data construction. We construct *VG-Pairs*, a synthetic paired-supervision dataset built from DL3DV [27] and SpatialVID [28]. For each real video clip, we treat the original clip as the geometry-consistent target and synthesize a geometry-distorted counterpart with Wan 2.1-Fun-InP. Given two boundary keyframes, Wan 2.1-Fun-InP generates the intermediate video segment connecting them. For clips with simple camera motion, this naturally reduces to a single segment conditioned on the first and last frames. For more complex trajectories, we insert additional intermediate keyframes and

Table 4: Training configurations of *VGC* with different video diffusion backbones.

Hyperparameter	**Wan 2.1-1.3B**	**Wan 2.2-14B**
Training strategy	Full fine-tuning	Full fine-tuning
Trainable module	DiT	DiT
Training steps	10K	3.5K
Video frames	81	81
Resolution	832×480	1248×720
Learning rate	1×10^{-5}	1×10^{-5}
Global batch size	64	64
Optimizer	AdamW	AdamW
Weight decay	0.01	0.01



Figure 8: **Examples from *VG-Pairs*.** The first two examples are constructed from DL3DV [27], and the last two are from SpatialVID [28]. In each case, the first row shows the synthesized geometry-distorted video used as training input, while the second row shows the corresponding ground-truth video used as supervision. The red boxes highlight typical geometric errors that the model learns to correct toward the target. The fourth example additionally illustrates a dynamic scene, in which the synthesized video contains inconsistent human identity and distorted geometry, providing supervision for correcting human motion and structure.

generate multiple shorter subsegments, then stitch them together after removing duplicated boundary frames.

We classify a trajectory as complex if its total camera rotation exceeds $R_{\text{thr}} = 60^\circ$ or its total translation exceeds $D_{\text{thr}} = 5\,\text{m}$. For such clips, we target a per-segment budget of $R_{\text{seg}} = 120^\circ$ rotation and $D_{\text{seg}} = 10\,\text{m}$ translation. Interior keyframes are selected by greedy non-maximum suppression over the motion-change score defined in Section 3.1, which combines the angular change of the camera bearing direction and the change of the rotation axis between consecutive frames, with mixing weight $\alpha = 1.0$. We require an absolute score threshold of $s_{\min} = 10^\circ$ and a relative threshold of at least $\beta = 0.5$ times the maximum score within the clip. To avoid overly short segments, we enforce a minimum temporal gap of $L_{\min} = 15$ frames between selected keyframes and cap the number of generated segments at $M_{\max} = 4$, corresponding to at most five keyframes. Frames with negligible motion, defined as translation below $10^{-3}\,\text{m}$ or rotation below 0.1°, are excluded when computing the score. Because Wan's VAE applies temporal compression, selected keyframes must also satisfy the corresponding frame-count compatibility constraint for each generated subsegment.

Algorithm 1 summarizes the full paired-data construction pipeline used in training.

Evaluation protocol. For the pose-conditioned benchmark, we follow Geometry Forcing [10] and evaluate on RealEstate10K [24] using 100 randomly sampled test videos and DFoT [25] as the source generator. We compare against Geometry Forcing and Geometry Forcing + REPA. For the prompt-driven benchmark, we follow VideoGPA [16] and evaluate both image-to-video and text-to-video generation using CogVideoX source models [2]. We randomly sample 200 prompts from the held-out DL3DV test set. In all cases, *VGC* is applied post hoc to the source generator outputs.

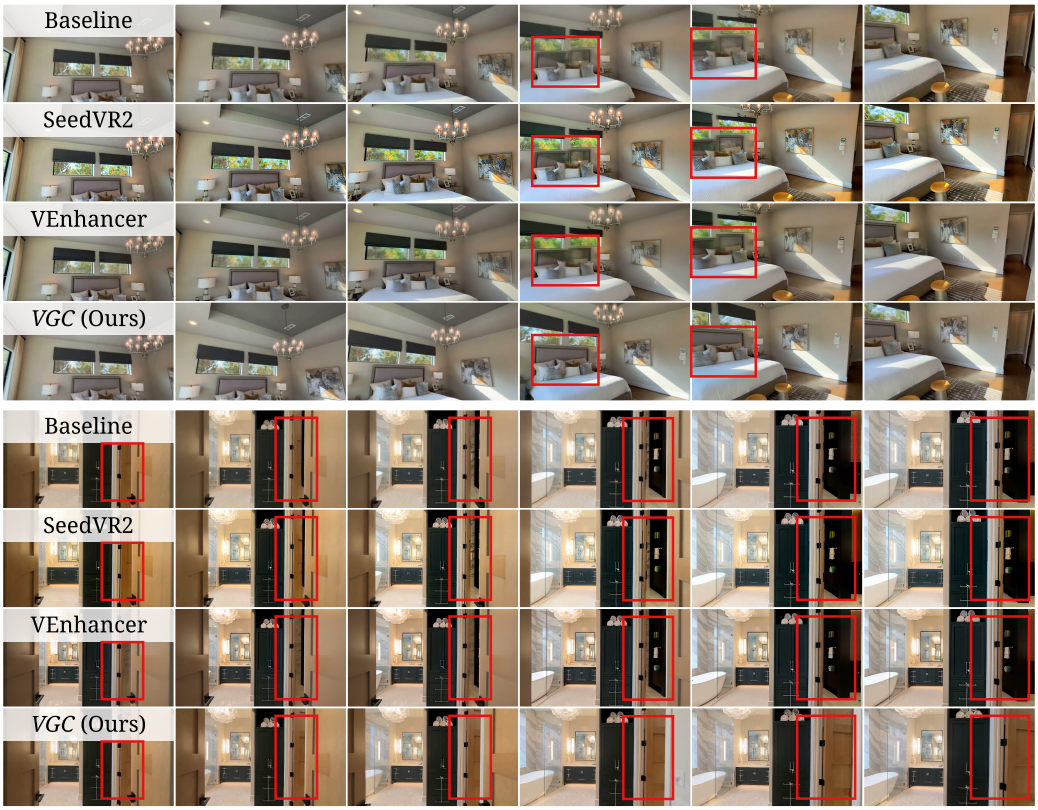

Figure 9: **Video refinement results in static scenes.** *VGC* corrects geometry distortions caused by camera motion, such as the warped bed headboard in the first example, and restores a consistent appearance across frames. In the second example, it resolves the baseline's inconsistent door state over time.

Metric definitions. For the pose-conditioned benchmark, we report PSNR, SSIM, LPIPS [30], and revisit error (RVE). Following Geometry Forcing [10], RVE measures long-range consistency under a full camera rotation by comparing the generated view at the end of the trajectory with the corresponding reference view. Lower RVE indicates better geometric consistency over long trajectories.

For the prompt-driven benchmark, we report PSNR, SSIM, LPIPS [30], MSE, and 3D consistency score (3DCS). Following VideoGPA [16], 3DCS is a reprojection-based geometry metric computed from the discrepancy between original frames and frames reprojected from reconstructed 3D geometry. In the implementation of VideoGPA, $\mathrm{3DCS} = 10 \cdot \mathrm{MSE} + \mathrm{LPIPS}$, where lower values indicate better geometric consistency.

Compute resources. Unless otherwise specified, the following numbers refer to the 1.3B model used in the main experiments. We train *VGC* on 4 nodes with 8 NVIDIA H100-80GB GPUs each (32 GPUs total). With bf16 mixed precision and gradient checkpointing, each training step takes approximately 7.75 seconds at global batch size 64, corresponding to about 21 hours for 10K steps and roughly 670 GPU hours in total. Peak GPU memory during training is approximately 45–55 GB per device. At inference, a hot-loaded model refines a single 81-frame 832×480 clip in about 90 seconds on one H100-80GB GPU.

D Comparison with Direct Generator Fine-tuning

A natural alternative to a separate corrector is to fine-tune the source generator directly on the geometry-consistent videos in *VG-Pairs*. We therefore fine-tune CogVideoX-5B T2V [2] with LoRA [35] on the target videos of *VG-Pairs*, using the official CogVideoX training code with rank 128, $\alpha = 64$, and adapters on the query, key, value, and output projections of the attention layers (`to_q`, `to_k`, `to_v`, and `to_out.0`). A rank-64 configuration produced slightly worse results, so we



Figure 10: **Video refinement results in dynamic scenes.** *VGC* improves motion consistency by recovering more plausible trajectories and structure for moving objects. In the first example, it restores a coherent trajectory for the car, which incorrectly fades or disappears in the baseline. In the second example, it recovers the cyclist's motion as riding away toward the left side of the frame, instead of drifting downward, shrinking unnaturally, and moving out of view.

Table 5: **Comparison with direct generator fine-tuning** on DL3DV [27], with CogVideoX-5B T2V [2] as the source generator. Best results are in bold.

Method	PSNR↑	SSIM↑	LPIPS↓	MSE↓	3DCS↓	Depth Error↓	Normal Error (°)↓
CogVideoX-5B T2V	15.6668	0.5389	0.5285	0.0333	0.8617	0.0375	18.71
+ LoRA fine-tuning	16.2128	**0.5552**	**0.5284**	0.0301	0.8676	0.0359	18.75
+ *VGC* (Ours)	**16.6940**	0.5519	0.5733	**0.0264**	**0.8376**	**0.0332**	**18.46**

report the rank-128 model. Both methods are evaluated on the same 200 DL3DV prompts with the protocol of Table 2. In addition to the reprojection metrics, we report two geometry-specific metrics computed with Depth Anything 3 [31]. For each video, we uniformly sample 10 frames and let Depth Anything 3 jointly predict per-frame depth, camera intrinsics, and camera extrinsics. For frame pairs separated by three sampling intervals, Depth Error is the scale-invariant AbsRel between the depth reprojected from the first frame into the second view and the depth predicted for the second frame, and Normal Error is the mean angular difference between corresponding surface normals derived from the back-projected points. Points that fall outside the target view, lie behind the camera, are occluded under forward-backward reprojection, or have undefined normals are excluded. As shown in Table 5, direct LoRA fine-tuning improves PSNR, SSIM, MSE, and Depth Error over the original generator and leaves LPIPS essentially unchanged, but slightly worsens 3DCS and Normal Error. *VGC* achieves the best PSNR, MSE, 3DCS, Depth Error, and Normal Error, whereas LoRA fine-tuning attains the best SSIM and LPIPS. These results indicate that paired video-to-video supervision yields a stronger gain in geometric consistency, while the comparison on appearance-oriented metrics is mixed (Section E). Beyond accuracy, direct fine-tuning must be repeated for every generator and

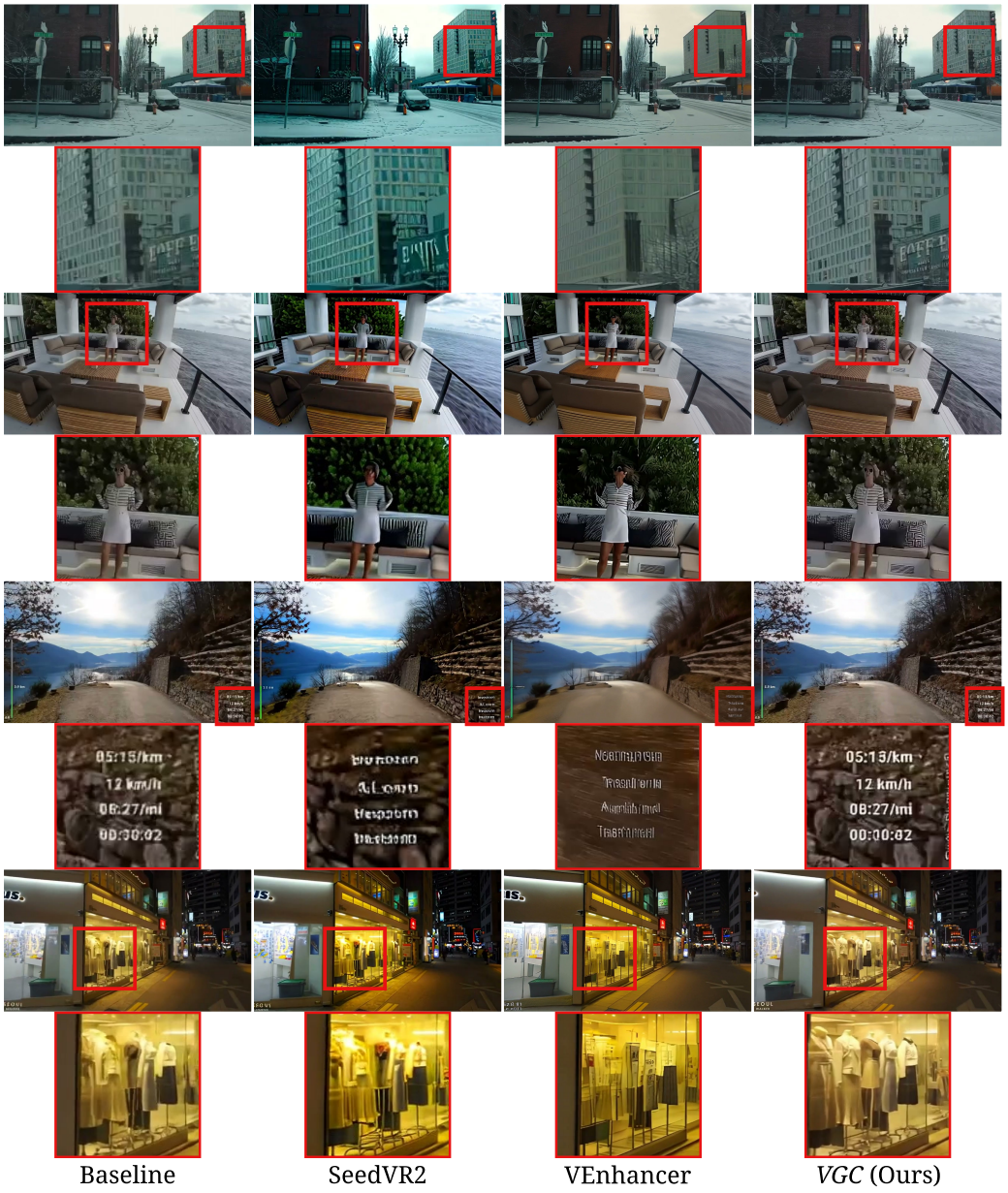

Figure 11: **Video super-resolution results.** The 14B version of *VGC* takes 480p videos as input and generates 720p outputs, naturally extending the framework to video super-resolution. Compared with SeedVR2 [20] and VEnhancer [15], *VGC* produces more detailed and refined results while remaining more faithful to the source video and avoiding hallucinated structures. It better preserves fine details such as the patterns on the building facade and sofa cushions, and also recovers text and clothing details more clearly.

requires access to its weights and training pipeline, whereas *VGC* is trained once and applied post hoc across model families.

E Failure Cases and Limitations

Figure 12 shows a representative failure case of *VGC*. When the distortion is extremely severe and introduces large structural inconsistencies in the source video, *VGC* may struggle to recover a coherent result. For example, if a door suddenly appears between a few consecutive frames, the missing geometric information cannot be reliably inferred from the corrupted input alone. Although

Algorithm 1 VG-Pairs construction from a real video clip

Require: real video clip $\mathbf{x} = (x_1, \ldots, x_T)$ with camera poses $\mathbf{p} = (p_1, \ldots, p_T)$
Require: image-to-video inpainting model G
Ensure: geometry-distorted video $\mathbf{y}$ paired with target $\mathbf{x}$

compute per-frame rotation increments $\Delta\phi_i$ and translation increments Δd_i
compute total rotation $R_{\text{tot}} = \sum_{i=1}^{T-1} \Delta\phi_i$ and total translation $D_{\text{tot}} = \sum_{i=1}^{T-1} \Delta d_i$
if $R_{\text{tot}} \le R_{\text{thr}}$ **and** $D_{\text{tot}} \le D_{\text{thr}}$ **then**
    $K \leftarrow \{1, T\}$
else
    $m \leftarrow \min\Big(M_{\max}, \max\Big(1, \Big\lceil \max\Big(\frac{R_{\text{tot}}}{R_{\text{seg}}}, \frac{D_{\text{tot}}}{D_{\text{seg}}}\Big)\Big\rceil\Big)\Big)$
    compute motion-change score $s(i)$ for each interior frame $i \in \{2, \ldots, T-1\}$
    discard frames with translation $< 10^{-3}$ m or rotation $< 0.1^\circ$
    $s_{\max} \leftarrow \max_i s(i)$
    keep candidate peaks satisfying $s(i) \ge s_{\min}$ and $s(i) \ge \beta s_{\max}$
    select up to $m-1$ interior keyframes by greedy non-maximum suppression with minimum gap $L_{\min}$
    if fewer than $m-1$ interior keyframes are selected **then**
        insert remaining keyframes by approximately uniform partition of the largest uncovered gaps
    end if
    $K \leftarrow \{1\} \cup \{\text{selected interior keyframes}\} \cup \{T\}$
end if
adjust K so that every subsegment length is compatible with Wan VAE temporal compression
sort K in ascending order
for $j = 1$ to $|K| - 1$ **do**
    generate subsegment $\mathbf{y}^{(j)} \leftarrow G(x_{k_j}, x_{k_{j+1}})$
end for
concatenate $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(|K|-1)}$ and remove duplicated boundary frames
return $(\mathbf{y}, \mathbf{x})$



Figure 12: **Failure case.** *VGC* recovers the correct geometric relation between the hallway and dining room and restores the disappearing cabinet, but produces blurry results for the plant and cabinet, while also introducing artifacts near the entrance to the dining room.

the model recovers the correct geometric relation between the two rooms, it still introduces artifacts near the dining room entrance in the last column. The refined result can also become blurry in regions such as the plant and the cabinet.

More broadly, *VGC* has several limitations. First, while it improves geometric consistency under camera motion, it does not always preserve fine visual details, as reflected in the fact that LPIPS does not consistently improve over the baseline. This trade-off also appears in the comparison with direct LoRA fine-tuning of the source generator (Section D): *VGC* achieves better PSNR, MSE, 3DCS, and depth and normal consistency, whereas LoRA fine-tuning attains better SSIM and LPIPS. Since PSNR and MSE can favor blurrier outputs, the appearance comparison between the two approaches is mixed, and the advantage of *VGC* is clearest on geometry-specific metrics. Second, the broader general-video-refinement behavior discussed in Section 4.3 relies on a much larger Wan 2.2-14B backbone rather than the 1.3B model used in our main setting. This larger variant is substantially more expensive to train and deploy, and we do not view it as the primary setting of the paper. Third, our quantitative evaluation of general video refinement relies on VBench [33] and DOVER [32],

which mainly assess global video quality and temporal consistency. These metrics do not reliably capture object-level temporal failures, such as objects disappearing or reappearing, identity changes, or abrupt structural changes, which are among the errors that *VGC* corrects most visibly. We therefore demonstrate these improvements through qualitative comparisons (Figure 7, Figure 9, and Figure 10). Important directions for future work include lighter-weight refinement models that retain strong geometry correction while improving sharpness and efficiency, and benchmarks with object-level annotations of identity, state, and motion continuity.

F Broader Impacts

Methods that improve the geometric consistency of generated videos may benefit video editing, post-production, controllable video generation, and other applications that require more stable scene structure under camera motion. In particular, a post-hoc corrector can improve existing generators without modifying the source model, which may reduce manual cleanup and broaden access to higher-quality video tools.

At the same time, improving the realism and coherence of synthetic videos may also increase the risk of misuse. A geometry-aware refinement model can make manipulated or misleading videos appear more convincing, especially when combined with super-resolution or other enhancement capabilities. We therefore view provenance, watermarking, and other content-authentication tools as important complementary safeguards for practical deployment.

G Asset Licenses and Terms of Use

Unless otherwise noted, all external assets are used for non-commercial academic research under their original licenses or access conditions.

- SpatialVID [28]: released under CC BY-NC-SA 4.0.
- DL3DV-10K [27]: released under the DL3DV-10K Terms of Use.
- RealEstate10K [24]: the official camera-trajectory release is licensed under CC BY 4.0.
- Wan 2.1 / Wan 2.2 [4]: released under Apache 2.0.
- CogVideoX [2]: the official codebase is released under Apache 2.0; the 2B model is Apache 2.0, while the 5B models are distributed under the CogVideoX model license.
- HunyuanVideo [3]: released under the Tencent Hunyuan Community License Agreement.
- SeedVR2 [20]: released under Apache 2.0.
- VideoGPA [16]: released under the MIT License.
- REPA [26]: released under the MIT License.
- DFoT [25]: released under the MIT License.
- Geometry Forcing [10]: the official repository's license states that new Geometry Forcing code is released under Apache 2.0, and that it includes an MIT notice for inherited template code.
- SEA-RAFT [29]: released under the BSD 3-Clause License.
- VEnhancer [15]: we use the authors' officially released repository for research comparison; as of May 5, 2026, the public repository page does not clearly expose a standalone software license.

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 accurately describe the paper's scope and contributions: a post-hoc video-to-video geometry corrector trained from synthetic paired supervision, with experiments on pose-conditioned, prompt-driven, and larger-model refinement settings. See the Abstract, Introduction, Sec. 3, Sec. 4, and Conclusion.

   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: The paper discusses limitations in the supplementary material, including occasional blurry outputs, the computational cost of the 14B extension, and the fact that the larger-model refinement setting is an extension rather than the main evaluation setting.

   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 does not present theoretical results, formal theorems, or proofs. The contribution is empirical and methodological.

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 formulation, training data construction, keyframe-selection procedure, evaluation settings, and metrics in the main text and supplementary material, including additional implementation and metric details. See Sec. 3, Sec. 4, and the supplementary material.

   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: We release the code and model checkpoints of *VGC*, linked from the project page (https://weijielyu.github.io/VGC), together with instructions for inference and training. Implementation, training, and evaluation details are also provided in the main paper and supplementary material.

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: The paper specifies the main training and evaluation settings, including datasets, source generators, metrics, optimizer, learning rate, batch size, training duration, and additional data-construction details in the supplementary material. See Sec. 4 and the supplementary material.

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: [No]

Justification: We do not report error bars or repeated-run statistics. Due to the cost of large-scale video training and evaluation, we report deterministic metrics on held-out test sets and use multiple benchmarks and qualitative comparisons to support the main claims.

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: The supplementary material reports the main training hardware configuration, per-step speed, approximate training duration, total GPU hours, peak memory usage, and inference runtime. In particular, the 1.3B model is trained on 32 NVIDIA H100-80GB GPUs, with approximately 21 hours of training, about 670 GPU hours in total, peak memory around 45–55 GB per GPU, and about 90 seconds per 81-frame clip at inference on one H100.

   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: To the best of our knowledge, the research described in the paper conforms to the NeurIPS Code of Ethics. The work uses existing public datasets and models for research purposes and does not involve human-subject studies.

   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: The supplementary material discusses both positive uses, such as improved video editing and refinement, and negative risks, such as producing more convincing synthetic videos that could be misused for deceptive content.

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: [N/A]

    Justification: The released models are post-hoc correctors that refine videos produced by existing public generators and do not introduce new generation capabilities beyond those models. Potential misuse risks are discussed in the broader impacts section of the supplementary material.

    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: The paper cites all major external datasets, models, and baseline methods, and the supplementary material includes an asset note listing licenses or terms of use for the main datasets and models used in our experiments, including public licenses, custom model licenses, and dataset terms of use where applicable.

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: We release the code and model checkpoints of *VGC*, linked from the project page (`https://weijielyu.github.io/VGC`). The release is documented with installation, inference, and training instructions.

    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: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    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: [N/A]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects, so IRB approval is not applicable.

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. Any language or vision-language models used in preprocessing are auxiliary tools rather than part of the proposed method itself.

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.