

LooseControlVideo: Directorial Video Control using Spatial Blocking

Shariq Farooq Bhat , Niloy J. Mitra , and Kalyan Sunkavalli

Adobe Research

Abstract. Precise 3D spatial orchestration in text-to-video generation remains a significant challenge, particularly for multi-object scenes where semantic layout and temporal dynamics are often entangled. While existing depth-conditioned models achieve good structural fidelity, they necessitate dense, frame-accurate guidance that is labor-intensive to author for dynamic events involving deformable objects. We present LooseControlVideo (LCV), a framework that enables intuitive and expressive control by using sparse, oriented 3D boxes as a “blocking” proxy. This allows users to author high-level layout and trajectory while leveraging a video generative model to generate realistic occlusions, dynamics and interactions. We achieve this by fine-tuning a Wan 2.2 backbone on a video dataset annotated with DNOCS, a novel encoding for 3D size, orientation and depth-ordered occlusions. Furthermore, our method allows for localized refinement—such as adjusting a jump trajectory or adding an interaction—with minimal disruption to the global scene context. Extensive evaluations on the nuScenes, HO-3D, and BEHAVE benchmarks demonstrate that LCV significantly outperforms existing 2D-box and flow-based baselines. Our findings indicate a 1.2-3x improvement in Trajectory Error; 2x improvement in Rigid Motion Consistency; and a 1.5-2x increase in Occlusion Accuracy over current state-of-the-art layout-conditioned models, demonstrating that oriented 3D primitives provide good geometric prior for complex, multi-agent video authoring.

Keywords: video controls · generation/editing · directorial controls

1 Introduction

While modern video diffusion models [5, 8, 37] have achieved remarkable photorealism, choreographing compelling narratives that involve intricate spatio-temporal synchronization and complex multi-object interactions remains an elusive task using current controls. This is because these models often lack the *directorial* controls required to orchestrate specific physical events, leaving a significant gap between generative capability and creative intent.

Existing control modalities present a difficult trade-off: natural language is too imprecise to describe exact spatial trajectories, whereas dense video signals, such as per-frame depth or edge maps, are nearly impossible for a user to manually author for dynamic scenes. This difficulty stems from the fact that

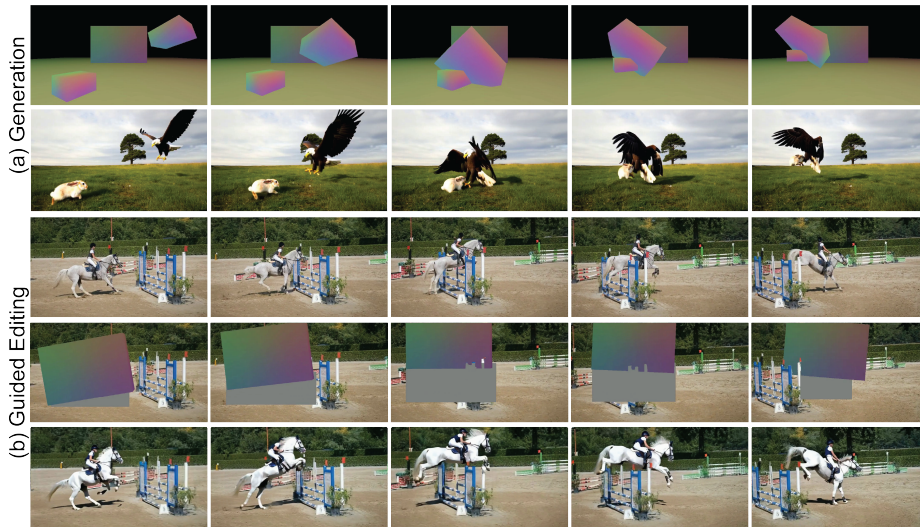


Fig. 1: Cinematic video authoring and editing via oriented 3D blocking. Our method, LooseControlVideo, enables precise choreography of complex multi-object interactions using sparse, oriented 3D bounding boxes. *Top*: From a coarse blocking of *an eagle stooping to catch a rabbit*, our model infers realistic morphological deformations and temporal synchronization. *Bottom*: LooseControlVideo facilitates intuitive motion editing; by lifting the jumping horse video (DAVIS dataset [28]) into our oriented 3D proxy space, a user can easily modify the jump trajectory of the horse while preserving its visual identity and the scene’s temporal consistency. See supplemental.

structure guidance (e.g., depth video) conflates two distinct control axes: (i) the layout, motion and interactions of the camera and objects in the scene, and (ii) the fine-grained object poses and deformations arising from those interactions. For instance, requiring a user to provide a frame-accurate depth sequence of an eagle stooping to catch a rabbit—capturing every wing-beat and skeletal contraction—is an impractical prerequisite.

Drawing inspiration from the “blocking” phase in professional cinematography, where directors orchestrate scene layouts using coarse spatial proxies to define movement, we introduce LooseControlVideo (LCV), a method that empowers users to author complex video events through the intuitive manipulation of oriented 3D primitives. Here, the user focuses solely on the high-level choreography (the layout and motion path and timing), while our generative model infers the secondary dynamics and realistic deformations (the execution). By allowing users to author these trajectories via simple 3D modeling tools or rigid-body physics simulations, LooseControlVideo effectively offloads the burden of geometric modeling as well as complex object deformations to the diffusion process, bridging the gap between casual authoring and cinematic precision.

We demonstrate that a simple-to-author sequence of *oriented 3D boxes*, when rendered as a structured temporal signal, provides a sufficient geometric prior for fine-tuning a large-scale video generator to achieve precise motion control. We expect the users to only provide a (sparse) set of keyframes consisting of oriented

3D boxes rendered from the target camera’s perspective. To resolve ambiguities in complex motion, we propose Depth-modulated Normalized Object Coordinate Space (DNOCS) representation, a novel coloring scheme to jointly encode local orientation as well as global depth.

To bridge the gap between coarse 3D proxies and high-fidelity synthesis, we render our 3D DNOCS representation into a 2D control signal (see Figure 1). We demonstrate that this enriched representation provides a robust conditioning signal for the Wan 2.2 DiT architecture, allowing the model to learn the *execution* of physical motion from sparse *intent* signals. To facilitate training without manual 3D annotation, we propose an automated pipeline that extracts temporally-tracked 3D oriented boxes from large-scale in-the-wild video datasets. By utilizing this system, we create a training corpus of $\sim 10\text{K}$ aligned RGB-DNOCS video pairs, enabling the model to learn the relationship between 3D spatial occupancy and realistic object-level dynamics.

We demonstrate LooseControlVideo through quantitative and qualitative evaluations on datasets including nuScenes [6] for navigation and HO-3D [11] and BEHAVE [2] for articulated interactions. We propose novel metrics for spatial blocking adherence, an area unaddressed in prior work. Our experiments show LooseControlVideo significantly outperforms state-of-the-art baselines; we achieve a $1.2\text{-}3\times$ reduction in Trajectory Error and a $1.5\text{-}2\times$ increase in Occlusion Accuracy while maintaining base DiT photorealism. Our contributions include: (i) a 3D-aware spatial blocking paradigm using oriented boxes to decouple choreography from local deformation; (ii) a specialized control mechanism and encoding for Diffusion Transformers enabling intuitive *loose* control; (iii) an automated pipeline for lifting 2D video datasets into 3D-conditioned training sets; and (iv) metrics to evaluate video model adherence to spatial controls.

2 Related Work

Image edit metaphors and handles. The rapid evolution of generative diffusion models has established a robust foundation for high-quality 2D image synthesis [13, 32, 34], yet the usability of these systems hinges entirely on the specific *modalities of control* exposed to the user. Early efforts in 2D spatial conditioning ranged from sparse inputs like 2D bounding boxes [22] and sketches [39] to dense signals such as segmentation maps [42] and depth gradients [26, 46]. ControlNet [46] and followups successfully unified these dense modalities (e.g., edge maps, human poses) for localized guidance; however, they often force users into *indirect control*—where the complexity of authoring means that signals must be derived from existing imagery—making it difficult to create from scratch. Alternatives emerged by allowing *text/instruction* based editors [3, 4] that offer zero-shot convenience but lack precise compositional steering, while *drag-and-point-based* interfaces [20, 35] enable local geometric manipulation but provide limited understanding of global articulation or temporal continuity. Coarse geometry-driven controls utilizing loose boxes [1] reduced the need for exact shape guidance; they do not support temporal changes, especially arising from

deformation and complex interactions. Our work addresses this critical gap; unlike 2D-centric methods [1, 22] that offer loose control without 3D-awareness, we introduce oriented 3D primitives that combine the intuitive flexibility of dense/sparse keyframes with the loose 3D scene blocking, enabling the choreography of complex, deforming multi-object events.

Video generators and control paradigms The transition from U-Net based inflated attention architectures [14, 36] to Diffusion Transformer (DiT) [27] has catalyzed a surge in high-fidelity video generation. While proprietary models [5, 8, 30] have demonstrated unprecedented temporal consistency and scale, the community has seen a parallel rise in open-weight/source alternatives [37, 45, 48]. Despite these foundational advancements, steering these models remains primarily limited to high-level text instructions or global style descriptors. Efforts to port the spatial precision of ControlNet to the temporal domain, such as ControlVideo [47] or Ctrl-V [25], typically still rely on dense video signals (e.g., depth or Canny maps, or pixel-space boxes). However, as seen in recent work on cinematic steering [12, 43], dense depth is not only difficult for users to author for complex dynamic scenes but also conflates camera ego-motion with local object deformation. While recent modules like CameraCtrl [12] and Direct-a-Video [44] introduce dedicated camera and trajectory parameters, they often struggle to maintain identity and structural integrity during intricate multi-object interactions. We leverage DiT architecture’s capacity for long-context temporal reasoning, conditioning it on sparse oriented primitives that sidestep the need for authoring/sourcing dense, hard-to-obtain structural maps.

3D-Aware Video Synthesis/Editing. Video control has increasingly moved toward 3D-aware scene composition, drawing inspiration from professional cinematography. Recent works have explored this through varied lenses: LLM-as-director frameworks [18, 23, 49] translate high-level scripts into spatial coordinates for rigged shapes, while Gen3C [31] and other world models focus on maintaining global 3D consistency. Parallel efforts like Diffusion-as-Shader [10] treat the diffusion process as a rendering pass over 3D tracked signals, often requiring a pipeline of rigging, animated meshes, and subsequent depth estimation to provide structure. In the realm of user-centric interaction, Boximator [41] introduced intuitive 2D box-guided control for object selection and motion, and recent point-track methods like Edit-by-Track [19] allow for precise motion editing via sparse point trajectories. However, these 2D-centric or track-based approaches face a dual challenge: (i) authoring exhaustive 2D tracks/boxes for dynamic, deforming scenes remains difficult, and (ii) they struggle to represent complex 3D behaviors such as axial spins, viewpoint-consistent deformations, and depth-ordered occlusions. By distinguishing between global rigid motion (encoded in our oriented 3D boxes) and local semantic deformation (inferred by the generator), we introduce a scalable oriented proxy that captures the full 6-DOF intent of an interaction without the overhead of full 3D rigging or dense tracking.

3 Method

3.1 Overview

We address the problem of controllable video generation and editing using light-weight 3D proxy signals. Our goal is to enable intuitive spatial control while preserving the generative flexibility of modern video diffusion models.

Let $y := \{y_t\}_{t=1}^T$ denote a target video of T frames and let p denote a text prompt describing the scene. We introduce control signal $b := \{b_t\}_{t=1}^T$ represented as a sequence of oriented 3D boxes. Each frame-level control map b_t encodes the (time-varying) spatial layout of (multiple) objects in the scene. In addition, we allow the user to specify 3D cameras, static or dynamic, $c := \{c_t\}_{t=1}^T$ for the video to be generated.

Our objective is to model the conditional distribution $p_\theta(y \mid p, b, c)$; this corresponds to video *generation*, controlled by the input text p and user-authored 3D bounding boxes b as seen through the cameras c . In addition, we would also like to support video *editing* scenarios where a (possibly spatially or temporally sparse) input source video, v , may be provided. In the most general setting, our video generator needs to model the distribution $p_\theta(y \mid p, b, c, v)$.

There are many design choices for how the bounding box and camera controls can be provided to a video generator. A box at time t can be parameterized by coordinate representation: center $\mathbf{o}_t \in \mathbb{R}^3$, scale $\mathbf{s}_t \in \mathbb{R}^3$, and rotation $\mathbf{R}_t \in SO(3)$. A naive approach for conditioning a video diffusion model on 3D box and camera parameters is to encode these parameters directly using MLPs and append the resulting embeddings as special tokens into the attention layers of the base DiT. Variants of this idea can be constructed by modifying how the tokens are injected or fused with the backbone. However, such a design suffers from a fundamental flaw: *video diffusion models operate primarily in two spatial dimensions and do not seem to possess an explicit strong understanding of 3D geometry [16]*. As a result, a tokenized coordinate representation of the box and camera parameters provides only implicit geometric cues.

In particular, critical spatial cues, such as depth ordering, occlusion relationships, and perspective projection, are not directly observable from the raw coordinate representation of the boxes and camera parameters. This introduces a significant representation mismatch: the conditioning signal is expressed in abstract 3D parameters, while the base video model operates entirely in the 2D image domain. Bridging this gap requires the model to internally reason about complex 3D scene geometry, effectively reconstructing the rendering pipeline that maps 3D structure to 2D observations. This substantially increases the learning burden and makes the conditioning difficult to utilize. Indeed, in practice, we observe that models trained with such token-based conditioning frequently fail to converge reliably and often learn to ignore the control inputs altogether, producing generations that do not respect the intended 3D structure at all.

The key insight of our work is to render the (time-varying) boxes (b_t) using the given camera (c_t) to a conditioning signal in image space *before* feeding to the video model, aligning the control representation with the domain in which

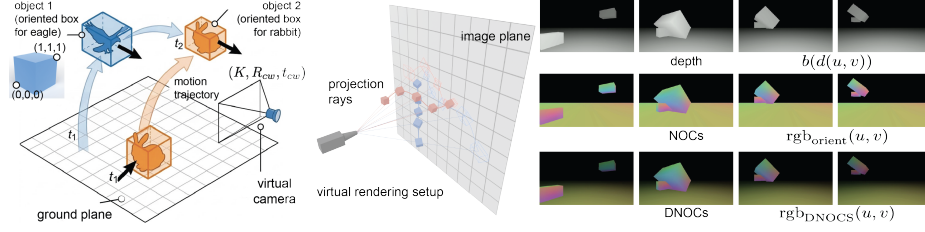


Fig. 2: Overview of the 3D control space and virtual rendering setup. (Left) The control space abstractly defined as 3D oriented boxes and their motion trajectories over time $([t_1, t_2])$ for distinct objects (e.g., eagle, blue and rabbit, orange) on an optional ground plane relative to a virtual camera. (Middle) The virtual rendering setup illustrates the projection of 3D object primitives onto a virtual image plane via projection rays originating from the camera. (Right) Illustrations of the dense maps rendered for a sequence of the moving objects: depth map, NOCs (Normalized Object Coordinates), and DNOCS (our depth+NOCS).

the base video model operates. By rendering beforehand, we effectively amortize the geometric projection step, exposing depth ordering, occlusion, and perspective cues explicitly rather than requiring the model to infer them from abstract parameters. This results in a control signal, v^{ctrl} , that is expressive enough to enable complex spatio-temporal operations without requiring architectural modifications to pre-trained video models. Moreover, this results in a simple yet expressive control interface that can handle *both* video generation and editing.

3.2 DNOCS: Rendered Oriented 3D Box Representation

Our goal is to render boxes such that the control signal can encode all possible object motion trajectories including 3D translations, rotations and scales, while respecting occlusions and relative relationships. We introduce Depth-modulated Normalized Object Coordinate Space (DNOCS), a hybrid color representation that jointly encodes local orientation as well as global depth. DNOCS makes key geometric cues such as object orientation, depth ordering, and occlusion relationships and object interactions directly observable in the conditioning frames. Inspired by NOCS [40] maps, we color each box using its local normalized coordinate frame, allowing the model to infer object orientation and relative pose. We render these colored boxes into the video frame using the specified camera, c_t . In addition, we modulate the color intensity according to the object’s depth with respect to the camera, introducing a global distance cue that standard NOCS encodings lack. See Figure 2 for an illustration.

Given camera intrinsics K and extrinsics (R_{cw}, t_{cw}) (world $\rightarrow$ camera), and a box defined by center C_w , rotation R_{wb} (box $\rightarrow$ world), and half-extents h , we cast a ray for each pixel and compute its intersection with the box. For pixel (u, v) the camera-space ray direction is,

$$d_c(u, v) = \left(\frac{u - c_x}{f_x}, \frac{v - c_y}{f_y}, 1 \right). \quad (1)$$

Transforming to world space using $R_{wc} = R_{cw}^\top$ gives,

$$o_w = -R_{wc}t_{cw}, \quad d_w(u, v) = R_{wc}d_c(u, v). \quad (2)$$

We express the ray in the box-local frame using $R_{bw} = R_{wb}^\top$:

$$o_b = R_{bw}(o_w - C_w), \quad d_b(u, v) = R_{bw}d_w(u, v). \quad (3)$$

We then compute ray-box intersection using the standard slab test for ray-AABB intersection in the local frame $[-h, h]$, yielding a hit mask $m(u, v)$, intersection parameter $\tau(u, v)$, and local surface point,

$$p_b(u, v) = o_b + \tau(u, v)d_b(u, v). \quad (4)$$

Finally, we obtain the corresponding camera-space depth is as,

$$p_w = o_w + \tau d_w, \quad p_c = R_{cw}p_w + t_{cw}, \quad z(u, v) = (p_c)_z, \quad (5)$$

with background pixels assigned $z = +\infty$.

Our DNOCS representation of control image combines a depth-independent orientation hue with a depth-dependent brightness, yielding a single 3-channel conditioning signal that carries both local orientation and global depth ordering. We first convert the local intersection to a unit direction on the sphere,

$$n(u, v) = \frac{p_b(u, v)}{\|p_b(u, v)\| + \epsilon}, \quad (6)$$

and map directions to RGB with approximately constant luminance using a smooth spherical color wheel,

$$\text{rgb}_{\text{orient}}(u, v) = \text{unit_sphere_color}(n(u, v); L, a), \quad (7)$$

where L is a fixed luminance (we use $L = 0.55$) and a controls chroma strength (we use $a = 0.35$). This yields easily distinguishable hues across orientations. Next, we compute a normalized inverse depth signal $d \in [0, 1]$ (closer $\rightarrow 1$, farther $\rightarrow 0$) so nearer surfaces have larger values:

$$d(u, v) = 1 - \text{clip}\left(\frac{z(u, v) - z_{\min}}{z_{\max} - z_{\min}}, 0, 1\right), \quad (8)$$

where $(z_{\min}, z_{\max})$ are 2^{nd} and 98^{th} percentile depths computed per frame from $\{z(u, v) \mid m(u, v) = 1\}$, making the distance-to-intensity mapping scale-invariant. Finally, we turn depth into a multiplicative brightness with an exponential falloff and a nonzero floor:

$$b(d) = \beta_{\min} + (1 - \beta_{\min}) \exp(-k(1 - d)), \quad (9)$$

with $\beta_{\min} = 0.08$ and $k = 2.0$ by default. The final DNOCS rendering is

$$\text{rgb}_{\text{DNOCS}}(u, v) = \text{rgb}_{\text{orient}}(u, v) \odot b(d(u, v)), \quad (10)$$

where $\odot$ denotes channel-wise multiplication (broadcasting b over RGB channels). Background pixels are set to black ($\text{rgb}_{\text{hyb}} = 0$) with $m = 0$ and $z = +\infty$.

Representation of input videos for Editing. Our goal is to enable loose control for both video generation and editing. The latter requires specifying the boxes in the context of the input video. We achieve this by compositing the rendered DNOCS representation into the input video frames in an occlusion-aware manner to construct the complete control, v^{ctrl} , to the video generator. For compositing, we estimate depth of the input video using VideoDepthAnything [7]. We make the authoring of v^{ctrl} as flexible as possible by allowing each frame of this context input to be any one of the following: (i) DNOCS-only frames; (ii) Input video segments to specify frames that must stay the same; (iii) Completely black (empty) frames which video model must fill in; and (iv) Spatially composited mixtures of input video frames and the DNOCS renderings. We also support gray masks over input video frames for regions that are to be removed.

This unified formulation enables the same model to handle both generation and editing. Moreover, it is highly expressive and allows users to choose which form of control should be applied on a per-frame and per-object basis. Figure 1 shows control video examples for generation (top) and editing (bottom).

3.3 Automatic Training Data Pipeline

Training LooseControlVideo requires a dataset of videos annotated with boxes for a wide variety of objects in the scenes. We build these annotations for a curated subset of internal stock videos of approximately 10k real videos using an automated, robust pipeline. For each video, we obtain object tracks using GroundingDINO [24] and SAM-based segmentation [29] to produce per-frame object masks. We then estimate per-object point clouds using monocular depth prediction [7] within the masked regions and fit oriented 3D bounding boxes per frame [21]. We refine the raw box trajectories using 3D Kalman filtering procedure to produce temporally consistent box sequences.

Control Video Construction. During training, we randomly construct control signals by randomly choosing between box-only renders (70%), and a mix of partial real video segments and the spatially composited mixtures (30%).

3.4 Training

We build on a ControlNet [46]-style architecture, WAN-VACE [17], and keep the base video diffusion architecture unchanged. The control video, v^{ctrl} is provided through the standard VACE conditioning pathway, which injects control residuals into the backbone DiT model. We LoRA fine-tune the VACE modules (rank 64) for 10k iterations on 4 H100 80GB GPUs while keeping the base model frozen. This design choice isolates the effect of the proposed control representation and demonstrates that strong controllability can be achieved without architectural modification.

At inference time, the user can switch between generation and editing modes simply by changing how v^{ctrl} is constructed: (i) *Pure Generation*. When v^{ctrl} consists of only DNOCS renderings, the model synthesizes a full video consistent with the specified camera and object layout and motion. (ii) *Editing and*

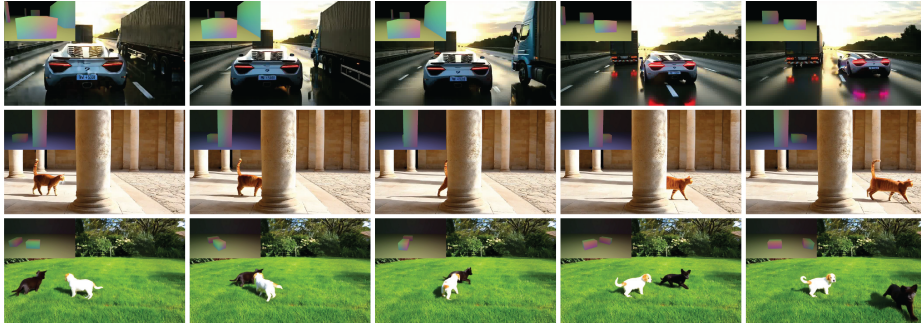


Fig. 3: Video generation results. Insets visualize the input 3D oriented box sequences used as conditioning proxies. *Top*: High-speed weaving and maneuvering. The oriented boxes guide a vehicle through narrow gaps between trucks, capturing subtle 6-DOF rotations and triggering responsive effects like brake light activation. *Middle*: Robust occlusion and shadow consistency. A cat maintains its identity and temporal coherence while walking behind a large pillar; note the preservation of intricate, viewpoint-consistent shadows and the smooth reappearance from total occlusion. *Bottom*: Articulated interaction and rhythmic motion. Two puppies interact with natural deformation and gait as one rotates around the other, demonstrating LCV’s ability to infer complex morphological changes from sparse loose/rigid proxies. In all cases, our easy-to-author oriented boxes anchor the structural orchestration required to drive the video generator. See supplemental webpage for videos and more examples.

Interpolation. When v^{ctrl} is a combination of the input video and the DNOCS renderings, the model preserves the visible regions while generating the remaining content consistent with the box controls and text prompt.

4 Evaluation and Results

While previous work has looked at the problem of 3D camera and 2D/3D bounding box controls for video generation models, we believe that we are the first to present a holistic combination of easy-to-author 3D “blocking” proxies combined with the ability to author complex, multi-object interactions and dynamics. We propose several novel metrics to evaluate the quality of such proxy-controlled video generation models and we evaluate our method against several baselines.

In particular, we measure whether generated videos respect object trajectories, spatial containment, motion dynamics, and occlusion ordering defined by the control signals. We compare against baselines that rely on weaker structural representations such as 2D optical flow or 2D bounding boxes, both common form of control signals adopted in controllable video generation methods.

Datasets. To measure the quality of our method in real-world scenarios, we make use of real-world video datasets that already come with 3D box annotations. We evaluate on three datasets covering both real-world driving scenes and complex human-object interactions. (i) *nuScenes* [6] is a large-scale autonomous driving dataset containing urban driving videos with annotated 3D bounding

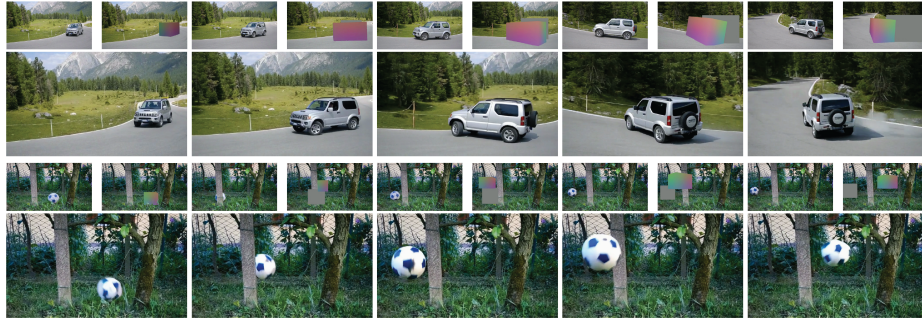


Fig. 4: Video motion editing via oriented 3D proxy manipulation. Insets visualize the (left) original video frame and (right) our editing interface, where the source object is neutralized (gray) and the new target trajectory is specified using our hybrid DNOCs-based oriented boxes. *Top:* Dynamic trajectory retargeting. We modify a jeep’s standard path from the DAVIS dataset [28] into a high-speed drift. Our model successfully infers secondary physics-based effects, including realistic tire skids and volumetric smoke generated by sudden braking. *Bottom:* Complex occlusion and spin editing. A football’s simple linear path is edited into a weaving trajectory. This requires delicate spatio-temporal reasoning to handle intricate occlusions behind trees and foliage, while simultaneously maintaining consistent spin and orientation throughout the new motion. Our method, even with simple-to-author controls, enables these high-fidelity edits through sparse 3D proxies, preserving the original scene’s identity while significantly altering its dynamics. Please see our supplemental webpage for video results.

boxes for multiple dynamic objects such as vehicles, pedestrians, and cyclists. We use the validation split and derive control signals directly from ground-truth 3D bounding boxes. These boxes capture full 3D position, orientation, and scale over time, enabling evaluation of spatial grounding and trajectory fidelity in real-world scenes. *(ii)* *HO-3D* [11] contains sequences of human hand-object interactions with accurate object pose annotations and mesh models. We derive oriented bounding boxes from the ground-truth object poses and meshes. The dataset is challenging due to rapid rotations, articulated hand motion, and frequent occlusions between hands and manipulated objects. *(iii)* *BEHAVE* [2] focuses on full-body human interactions with large objects such as chairs and suitcases. Compared to HO-3D, the dataset contains more significant object motion and complex physical interactions. As in HO-3D, we derive oriented bounding boxes from ground-truth poses and meshes. Across all the datasets, the control signals are rendered into conditioning videos that are provided to the generation model.

Evaluation Metrics. We evaluate structural adherence using metrics based on scene flow and mask tracking via video segmentation. These evaluate spatial grounding, trajectory accuracy, motion consistency, and occlusion reasoning. For each object i at time t , we denote the 2D projected control box mask as $B_{i,t}$ and the generated object mask as $M_{i,t}$, obtained via box-prompted video segmentation and tracking using SAM-2 [29].

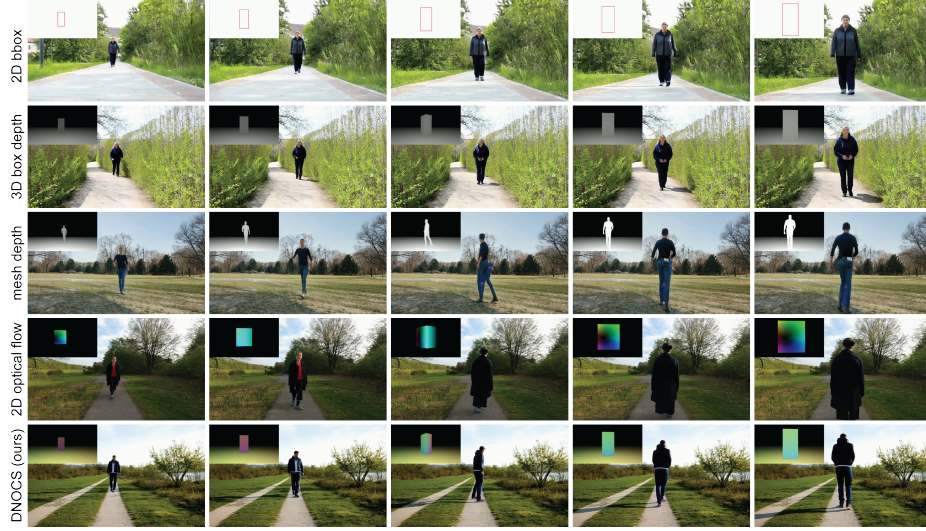


Fig. 5: Qualitative comparison. We compare our DNOCs-based oriented box control against several alternatives, including 2D bounding boxes, 3D box depth, mesh depth, and 2D optical flow. While 2D-centric methods struggle with viewpoint-consistent orientation and temporal grounding, and dense depth/mesh guidance can over-constrain natural dynamics, our method (bottom) excels at preserving precise 6-DOF choreography (note the 360 rotation of the person) while still generating natural object deformations and realistic environmental interactions.

(i) *Containment* ($\text{Contain} \uparrow$) measures how well generated object pixels remain within the prescribed control region:

$$\text{Contain}_{i,t} = \frac{|M_{i,t} \cap B_{i,t}|}{|M_{i,t}| + \varepsilon}. \quad (11)$$

The reported score averages over all objects and frames.

(ii) *Trajectory Error* ($\text{TrajErr} \downarrow$) measures the deviation between the generated object trajectory and the control trajectory. Let $c_{i,t}$ be the centroid of the generated object mask and $b_{i,t}$ the projected center of the control box. The error is

$$\text{TrajErr} = \frac{1}{NT} \sum_{i=1}^N \sum_{t=1}^T \|c_{i,t} - b_{i,t}\|_2. \quad (12)$$

(iii) *Occlusion Accuracy* ($\text{OcclAcc} \uparrow$) measures whether the generated video respects correct depth ordering between overlapping objects. Let $O_{n,f,t}$ denote the overlap region between near object n and far object f . The near dominance ratio is defined as

$$\text{NDR}_t = \frac{|M_{n,t} \cap O_{n,f,t}|}{|O_{n,f,t}| + \varepsilon}. \quad (13)$$

A frame is counted as correct if $\text{NDR}_t > 0.5 + \delta$, where $\delta = 0.05$. Occlusion accuracy is the fraction of correct frames among all valid overlaps.

Table 1: Real-world structural control on nuScenes [6]. We evaluate the ability of various control signals to reconstruct real-world dynamics using ground-truth 3D boxes from the **nuScenes** validation set. Ours, utilizing oriented 3D boxes, significantly outperforms 2D-based baselines on relevant metrics [15] in spatial grounding (TrajErr) and occlusion reasoning (OcclAcc), while preserving high visual quality. Upward arrows ($\uparrow$) indicate higher is better; **best** results are in bold, second best are underlined.

Method	Input	Contain $\uparrow$	GOW $\uparrow$	GMFA $\downarrow$	RMC $\downarrow$	TrajErr $\downarrow$	OcclAcc $\uparrow$	Quality
Control-free Baseline	GT First + Last frames	10.22	40.65	0.828	0.863	90.12	41.45	76.45
VACE	2D Flow	21.23	<u>86.76</u>	0.135	0.566	7.86	73.91	73.90
VACE ft	2D Flow	22.45	85.32	<u>0.093</u>	<u>0.528</u>	<u>6.78</u>	<u>79.32</u>	<u>75.50</u>
VACE ft	2D Boxes	96.33	42.33	0.232	0.735	16.66	42.45	66.34
LooseControlVideo	Rendered Oriented 3D Boxes	<u>87.93</u>	97.32	0.066	0.318	5.79	92.69	74.45

(iv) *Rigid Motion Consistency (RMC $\downarrow$)* evaluates whether the generated scene motion matches the rigid motion implied by the control animation. Scene flow is estimated from predicted optical flow using Raft [38] and depth using VDA [7], producing a 3D motion vector $v_t(x, y)$ for each pixel. Let $\Delta T_{i,t}$ denote the rigid motion of object i between frames. For pixels inside the object region, RMC is defined as the error between specified rigid motion and estimated scene flow:

$$\text{RMC} = \text{median}_{(x,y) \in B_{i,t}} \|X_{t+1}(x, y) - \Delta T_{i,t} X_t(x, y)\|_2. \quad (14)$$

(v) *Global Motion Field Agreement (GMFA $\downarrow$)* measures agreement between the estimated scene flow $v_t(x, y)$ and the motion field predicted from the control animation (combining camera motion and object rigid motion). It is computed as the median motion error over all pixels:

$$\text{GMFA} = \text{median}_{(x,y)} \|v_t(x, y) - v_{\text{pred}}(x, y)\|_2. \quad (15)$$

(vi) *Global Overlap Winner (GOW $\uparrow$)* evaluates occlusion reasoning in overlapping regions. For each pixel in the overlap region O , we measure how well the observed scene flow is explained by the rigid motion of the near and far objects, respectively, and calculate accuracy:

$$\text{GOW} = \frac{1}{|O|} \sum_{(x,y) \in O} \mathbb{1}[e_n(x, y) < e_f(x, y)], \quad (16)$$

where $e_n(x, y) = \|v(x, y) - v_n(x, y)\|_2$ and $e_f(x, y) = \|v(x, y) - v_f(x, y)\|_2$. denote motion reconstruction errors for near and far objects, respectively.

(vii) *Visual Quality* using “overall normalized average quality” from VBench [15].

4.1 Results

Table 1 presents results on the **nuScenes** dataset. Our method significantly improves structural adherence across key metrics. In particular, it achieves the lowest trajectory error and highest occlusion accuracy, indicating that oriented

Table 2: Comparison of object-centric structural control on HO-3D [11] and BEHAVE [2]. We evaluate our method on complex human/object interactions by deriving 3D oriented bounding boxes from ground-truth poses and meshes. Our approach (LooseControlVideo **LCV**) consistently achieves superior motion alignment (GMFA, RMC) and trajectory accuracy (TrajErr), demonstrating that oriented 3D primitives provide a more robust prior for complex/deforming motion (e.g., involving spings and turns) and close-contact interactions than 2D flow or axis-aligned boxes. “ft” implies finetuning on our training set.

Dataset	Method	Input	Contain $\uparrow$	GOW $\uparrow$	GMFA $\downarrow$	RMC $\downarrow$	TrajErr $\downarrow$	OcclAcc $\uparrow$	Quality
HO-3D	Control-free	GT First + Last	46.8	52.1	0.440	0.362	38.5	53.4	76.4
	VACE	2D Flow	63.4	<u>89.7</u>	0.083	0.214	6.3	81.5	73.3
	VACE ft	2D Flow	69.1	88.2	<u>0.071</u>	0.192	<u>5.4</u>	<u>84.2</u>	<u>73.6</u>
	VACE ft	2D Boxes	97.9	56.8	0.126	<u>0.181</u>	9.7	55.1	72.4
	LCV (Ours)	Rendered 3D Boxes	<u>91.3</u>	97.4	0.045	0.122	3.9	94.1	72.9
BEHAVE	Control-free	GT First + Last	42.3	46.9	0.611	0.490	54.8	48.6	76.2
	VACE	2D Flow	57.2	<u>86.3</u>	0.121	<u>0.341</u>	8.9	74.8	75.1
	VACE ft	2D Flow	63.8	84.7	<u>0.098</u>	0.318	<u>7.6</u>	<u>78.5</u>	<u>75.8</u>
	VACE ft	2D Boxes	95.8	48.2	0.238	0.412	14.9	49.7	69.3
	LCV (Ours)	Rendered 3D Boxes	<u>88.6</u>	95.6	0.062	0.207	5.8	90.2	75.0

3D boxes rendered in DNOCS provide strong geometric priors for modeling real-world scene dynamics. Compared to 2D flow-based baselines, which encode motion only in image space, our representation captures full 3D spatial relationships including orientation and depth ordering. The 2D box baseline achieves high containment scores due to tight spatial constraints in image space, but lacks orientation and depth information, resulting in worse motion alignment and trajectory accuracy. In contrast, our method maintains strong containment while significantly improving motion fidelity.

Table 2 reports results on **HO-3D** and **BEHAVE**, datasets involving complex human-object interactions with rotations and close-contact motion. Our method consistently achieves the best performance in motion alignment (GMFA, RMC) and trajectory accuracy (TrajErr). These improvements demonstrate that oriented 3D primitives provide a stronger structural prior for complex object motion than 2D flow or boxes. The control-free baseline achieves the highest perceptual quality by receiving ground-truth frames but fails to reproduce correct motion, resulting in large trajectory errors. Overall, our approach provides the best balance between structural fidelity and visual realism.

Tab. 4 ablates DNOCS: NOCS alone ($k=0$) loses ~ 19 pp of OcclAcc versus DNOCS $k=2$, confirming depth modulation is the key driver of occlusion reasoning. $k=2$ is the sweet spot; over-modulation ($k=4$) washes out the orientation cue. Specifying boxes only on every 8th frame recovers trajectories within 2% TrajErr of dense control on nuScenes, confirming robust sparse-keyframe support.

We present qualitative results for video generation in Figure 3. As can be seen here, LooseControlVideo can be used to author extremely complex video stories with intricate object dynamics. This includes trajectories that converge to create interactions (e.g., the eagle and rabbit in Figure 1 top), trajectories are designed

to *avoid* interactions (the weaving car in Figure 3 top), objects that disappear and re-appear because of occlusions (the cat in Figure 3 middle) and scenes where orientations can convey deeply meaningful narratives (the puppies facing each other and playing in Figure 3 bottom). In all these cases, LooseControlVideo allows for a simple, expressive control authoring interface that leverages video models to generate complex appearance, animations, and secondary dynamics.

Figure 4 presents results for video *editing* and demonstrates that LooseControlVideo can edit the motion of individual objects in the scene while preserving the overall identity and appearance of this scene, generating secondary physics-based effects (the skidding car in Figure 4 top) and staying faithful to scene geometry and occlusions (the soccer ball in Figure 4). This makes LooseControlVideo a powerful tool for post-production video editing.

In Figure 5, we evaluate alternate control signals—specifically 2D bounding boxes, dense 2D depth maps constructed from 3D bounding boxes or 3D meshes, and 2D optical flow. Our spatial blocking paradigm combined with the DNOCS-based control signal does the best job of capturing object motions with view and motion ambiguities (an issue with 2D centric methods) while preserving the flexibility to infer natural motion and secondary effects (a challenge with dense guidance methods that, additionally, are challenging to author). We further validate LCV through a 2-AFC pairwise user study (16 participants, 64 votes per method pair); LCV is preferred over the Depth-only, Optical-Flow, and 2D-Box baselines in 78.1%, 87.5%, and 92.2% of comparisons respectively. Full details are provided in the supplementary material.

4.2 Comparison with 3D-Aware Baseline

We additionally compare with 3DTrajMaster [9], the most directly comparable 3D-aware video generation method. By design, 3DTrajMaster accepts at most 3 entities per clip; we therefore evaluate using three selected objects per clip on HO-3D and BEHAVE and exclude nuScenes (its scenes routinely contain >10 actors). As shown in Tab. 3, LCV outperforms 3DTrajMaster by ~ 10 – $12\times$ in TrajErr and ~ 25 pp in OcclAcc on both datasets. This confirms that our oriented 3D proxy combined with DNOCS provides substantially stronger geometric grounding than trajectory-only approaches.

4.3 Limitations

Identity-to-Box Assignment. Our primary limitation is that we do not *explicitly* bind specific visual identities to individual 3D boxes. This binding problem is particularly challenging in a residual ControlNet-style architecture where global features often blend. In the future, we aim to incorporate multi-view reference images assigned to specific 3D anchors [33], allowing for consistent identity preservation in complex, multi-character scenes.

Manual Timing. Although our method decouples deformation from trajectory, the user is still responsible for authoring the temporal rhythm of interactions. While this aligns with professional animator workflows which prioritize

Table 3: Comparison with 3DTrajMaster [9]. 3DTrajMaster is limited to ≤ 3 entities per clip; we evaluate on three selected objects from HO-3D and BEHAVE (nuScenes excluded: scenes have > 10 actors). LCV outperforms by $\sim 10\text{--}12\times$ in TrajErr and ~ 25 pp in OcclAcc on both datasets.

Method	HO-3D		BEHAVE	
	TrajErr↓	OcclAcc↑	TrajErr↓	OcclAcc↑
3DTrajMaster	47.2	67.5	52.6	64.8
LCV (ours)	3.9	94.1	5.8	90.2

Table 4: DNOCS ablation (nuScenes). Sweeping k ; $k=0$ is plain NOCS.

Variant	TrajErr↓	RMC↓	OcclAcc↑
Depth-only	7.42	0.471	81.34
NOCS ($k=0$)	6.18	0.402	73.56
DNOCS $k=1$	6.05	0.371	88.21
DNOCS $k=2$ (ours)	5.79	0.318	92.69
DNOCS $k=4$	6.34	0.395	89.47

precise control over motion timing, an interesting future direction is to be able to specify spatial trajectories sparsely and let the generative model infer timing.

5 Conclusion

We have introduced LooseControlVideo as a new simple-but-effective control mechanism that bridges the gap between high-level creative intent and low-level video synthesis through the use of *oriented 3D bounding boxes*. By adopting the *blocking metaphor* commonly used in cinematography and game design, we demonstrate that sparse, 3D-aware primitives provide a sufficient yet loose geometric prior to orchestrate complex multi-object events – effectively decoupling global choreography (aka timing) from fine-grained object dynamics (aka deformations). Our experiments across diverse datasets, including driving scenes [6] and articulated human-object interactions [2], show that LooseControlVideo achieves better spatial adherence and temporal consistency compared to traditional 2D-based control signals using video metrics [15]. Ultimately, this approach moves us closer to an intuitive, simple-to-use, director-in-the-loop paradigm for video generation, where complex physical interactions can be authored with ease without sacrificing photorealistic fidelity or structural rigor.

References

1. Bhat, S.F., Mitra, N., Wonka, P.: Loosecontrol: Lifting controlnet for generalized depth conditioning. In: ACM SIGGRAPH 2024 Conference Papers. pp. 1–11 (2024)
2. Bhatnagar, B.L., Xie, X., Petrov, I., Sminchisescu, C., Theobalt, C., Pons-Moll, G.: BEHAVE: dataset and method for tracking human object interactions. In: CVPR (2022)
3. Brack, M., Friedrich, F., Kornmeier, K., Tsaban, L., Schramowski, P., Kersting, K., Passos, A.: Ledits++: Limitless image editing using text-to-image models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8861–8870 (2024)

4. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18392–18402 (2023)
5. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Lecomte, J., Sukhum, A., Senpuru, D., et al.: Video generation models as world simulators. OpenAI Blog (2024)
6. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nusenes: A multimodal dataset for autonomous driving. In: CVPR (2020)
7. Chen, S., Guo, H., Zhu, S., Zhang, F., Huang, Z., Feng, J., Kang, B.: Video depth anything: Consistent depth estimation for super-long videos. In: CVPR (2025)
8. DeepMind, G.: Veo: Google’s most capable generative video model. Google Technical Report (2024)
9. Fu, X., Liu, X., Wang, X., Peng, S., Xia, M., Shi, X., Yuan, Z., Wan, P., Zhang, D., Lin, D.: 3dtrajmaster: Mastering 3d trajectory for multi-entity motion in video generation (2024)
10. Gu, Z., Yan, R., Lu, J., Li, P., Dou, Z., Si, C., Dong, Z., Liu, Q., Lin, C., Liu, Z., Wang, W., Liu, Y.: Diffusion as shader: 3d-aware video diffusion for versatile video generation control. In: SIGGRAPH (2025)
11. Hampali, S., Rad, M., Oberweger, M., Lepetit, V.: Honnotate: A method for 3d annotation of hand and object poses. In: CVPR (2020)
12. He, H., Xu, Y., Guo, Y., Wetzstein, G., Dai, B., Li, H., Yang, C.: Cameractrl: Enabling camera control for video diffusion models. In: ICLR (2025)
13. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. *Advances in neural information processing systems* **33**, 6840–6851 (2020)
14. Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models. In: NeurIPS. pp. 8633–8646 (2022)
15. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., Wang, Y., Chen, X., Wang, L., Lin, D., Qiao, Y., Liu, Z.: VBench: Comprehensive benchmark suite for video generative models. In: CVPR (2024)
16. Jeong, H., Huang, C.H.P., Ye, J.C., Mitra, N., Ceylan, D.: Track4gen: Teaching video diffusion models to track points improves video generation. In: CVPR (2025)
17. Jiang, Z., Han, Z., Mao, C., Zhang, J., Pan, Y., Liu, Y.: Vace: All-in-one video creation and editing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17191–17202 (2025)
18. Kizil, M.B., Sanli, E., Mitra, N.J., Erdem, E., Erdem, A., Ceylan, D.: Lamp: Language-assisted motion planning for controllable video generation. In: CVPR (2026)
19. Lee, Y.C., Zhang, Z., Huang, J., Wang, J.H., Lee, J.Y., Huang, J.B., Shechtman, E., Li, Z.: Generative video motion editing with 3d point tracks. In: CVPR (2026)
20. Li, R., Zheng, C., Rupprecht, C., Vedaldi, A.: Dragapart: Learning a part-level motion prior for articulated objects. In: European Conference on Computer Vision. pp. 165–183. Springer (2024)
21. Li, Y., Wu, X., Chrysanthou, Y., Sharf, A., Cohen-Or, D., Mitra, N.J.: Globfit: Consistently fitting primitives by discovering global relations. *ACM TOG* **30**(4), 52:1–52:12 (2011)
22. Li, Y., Liu, H., Wu, Q., Mu, F., Yang, J., Gao, J., Li, C., Lee, Y.J.: Gligen: Open-set grounded text-to-image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22511–22521 (2023)
23. Lin, H., Zala, A., Cho, J., Bansal, M.: Videodirectorgpt: Consistent multi-scene video generation via llm-guided planning. In: COLM (2024)

24. Liu, S., Zuo, Z., Hou, J., Peng, H., Li, H., Hui, J., Huang, J., Li, F., Zhang, L., et al.: Grounding dino: Marrying dino with grounded pre-training for open-vocabulary object detection. In: ECCV (2023)
25. Luo, G.Y., Luo, Z.H., Gosselin, A., Jolicoeur-Martineau, A., Pal, C.: Ctrl-v: Higher fidelity video generation with bounding-box controlled object motion (2024), <https://arxiv.org/abs/2406.05630>
26. Mou, C., Wang, X., Xie, L., Zhang, J., Qi, Z., Shan, Y., Qie, X.: T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. arXiv preprint arXiv:2302.08453 (2023)
27. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: ICCV (2023)
28. Perazzi, F., Pont-Tuset, J., McWilliams, B., Gool, L.V., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR (2016)
29. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., et al.: Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714 (2024)
30. Reid, M., Savinov, N., Teplyashin, D., Coppin, D., Mumtaz, A., Ma, S., Paduraru, C., Paquet, U., Hayes, P., et al.: Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv preprint arXiv:2403.05530 (2024)
31. Ren, X., Shen, T., Huang, J., Ling, H., Lu, Y., Nimier-David, M., Müller, T., Keller, A., Fidler, S., Gao, J.: Gen3c: 3d-informed world-consistent video generation with precise camera control. In: CVPR (2025)
32. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
33. Saha, O., Krs, V., Mech, R., Maji, S., Blackburn-Matzen, K., Gadelha, M.: Sigma-gen: Structure and identity guided multi-subject assembly for image generation (2025)
34. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems* **35**, 36479–36494 (2022)
35. Shi, Y., Xue, C., Liew, J.H., Pan, J., Yan, H., Zhang, W., Tan, V.Y., Bai, S.: Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8839–8849 (2024)
36. Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., Parikh, D., Gupta, S., Taigman, Y.: Make-a-video: Text-to-video generation without text-video data. In: ICLR (2022)
37. Team, W.: Wan: Open and high-quality video generation with 3d-aware transformers, <https://arxiv.org/abs/2503.20314>
38. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: ECCV (2020)
39. Voynov, A., Aberman, K., Cohen-Or, D.: Sketch-guided text-to-image diffusion models. In: ACM SIGGRAPH 2023 Conference Proceedings. pp. 1–11 (2023)
40. Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., Guibas, L.J.: Normalized object coordinate space for category-level 6d object pose and size estimation. In: CVPR (June 2019)
41. Wang, J., Zhang, Y., Zou, J., Zeng, Y., Wei, G., Yuan, L., Li, H.: Boximator: Generating rich and controllable motions for video synthesis. arXiv preprint arXiv:2402.01566 (2024), <https://arxiv.org/abs/2402.01566>

42. Wang, T., Zhang, T., Zhang, B., Ouyang, H., Chen, D., Chen, Q., Wen, F.: Pretraining is all you need for image-to-image translation. arXiv preprint arXiv:2205.12952 (2022)
43. Wang, Z., Yuan, Z., Wang, X., Li, Y., Chen, T., Xia, M., Luo, P., Shan, Y.: Motionctrl: A unified and flexible motion controller for video generation. In: SIGGRAPH (2023)
44. Yang, S., Hou, L., Huang, H., Ma, C., Wan, P., Zhang, D., Chen, X., Liao, J.: Direct-a-video: Customized video generation with user-directed camera movement and object motion. In: SIGGRAPH (2024)
45. Yang, Z., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. In: ICLR (2025)
46. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3836–3847 (2023)
47. Zhao, M., Wang, R., Bao, F., Li, C., Zhu, J.: Controlvideo: Conditional control for one-shot text-driven video editing and beyond (2023), <https://arxiv.org/abs/2305.17098>
48. Zheng, Z., Peng, X., Yang, T., Shen, C., Li, S., Liu, H., Zhou, Y., Li, T., You, Y.: Open-sora: Democratizing efficient video production for all (2024), <https://arxiv.org/abs/2412.20404>
49. Zhu, H., He, T., Tang, A., Guo, J., Chen, Z., Bian, J.: Compositional 3d-aware video generation with llm director. In: NeurIPS (2024)