LEARNING ROTATION-AWARE VIDEO MOTION CONTROL FROM REAL-WORLD VIDEOS

Anonymous authors
Paper under double-blind review

ABSTRACT

Motion control for video generation prescribes the translation and the rotation of the content of a clip. Training a model to follow an explicit rotation requires per-frame six-degree-of-freedom supervision that real footage does not provide. Prior work therefore relies on simulation or robotics data, but this introduces a domain shift for general motion editing. We present Swivel, a framework that learns 6-DoF video motion control from real-world video alone. It builds on an observation about 3D foundation models: from their outputs alone, the rigid parts of a clip can be proposed automatically and the 6-DoF motion of each part can be measured without any human marking, which turns ordinary footage into a large corpus of clips paired with 6-DoF trajectories. We then train a video diffusion model on this corpus for sparse but precise 6-DoF control, with rotating cubes in screen space as the condition signal, and the model supports both image-to-video generation and inpainting-based video editing. Results show that Swivel follows the 6-DoF motion signals more closely than prior motion controls and generates higher-quality videos at 720p than the baselines, and human evaluation shows a large preference for Swivel over the baselines. Video results are available on our anonymous project page: swivel-videogen.github.io.

1 INTRODUCTION

Video generation models synthesise increasingly realistic footage from a text prompt. However, a prompt only describes a clip, and many applications require the motion of a video to be specified exactly. Motion control adds a signal that prescribes how the content should move, and the model generates a clip that follows it. Such control is useful in production settings, for example when previsualising a shot, redirecting an action in existing footage or animating a character without a rig.

The signals in common use differ mainly in the expressiveness and flexibility of their control modality. Sparse signals such as drags and point trajectories (Wang et al., 2024; Wu et al., 2024; Zhang et al., 2025; Burgert et al., 2026) are inexpensive to author and specify the position of a subject in screen space or in 3D, but they make rotation hard to specify and often introduce ambiguity. Dense signals such as warped noise and 3D tracking videos (Burgert et al., 2025; Gu et al., 2025; Geng et al., 2025) gain expressivity from density alone, since orientation follows from the joint motion of many elements without being specified, but such a signal is hard to author by hand. Six-degree-of-freedom (6-DoF) control requires a signal that specifies rotation as well as position, and neither end of this axis provides one that a user can author cheaply.

Training such a control requires clips paired with the signal that describes their motion. For 2D signals, point trackers and pose estimators supply this pairing on ordinary footage. Methods that need a 6-DoF signal face a harder problem, since such pairs are difficult to collect from real-world video. Some methods work around it with a game engine or other synthetic data (Fu et al., 2025; Shuai et al., 2025), and others use footage in which the hardware itself reports the pose, such as a robot arm (Zhu et al., 2025). The rotation is then exact, but these sources introduce a domain shift when the model is applied to general video generation or editing.

ProxyPose (Zhang et al., 2026b) estimates the rotation of a surface point by video-to-video diffusion, translating the input video into a proxy video of a coloured polyhedron anchored at a user-specified query point. The procedure requires a video diffusion model per query point and is correspondingly expensive. It further requires the user to designate the query point, whereas the motion in a clip

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

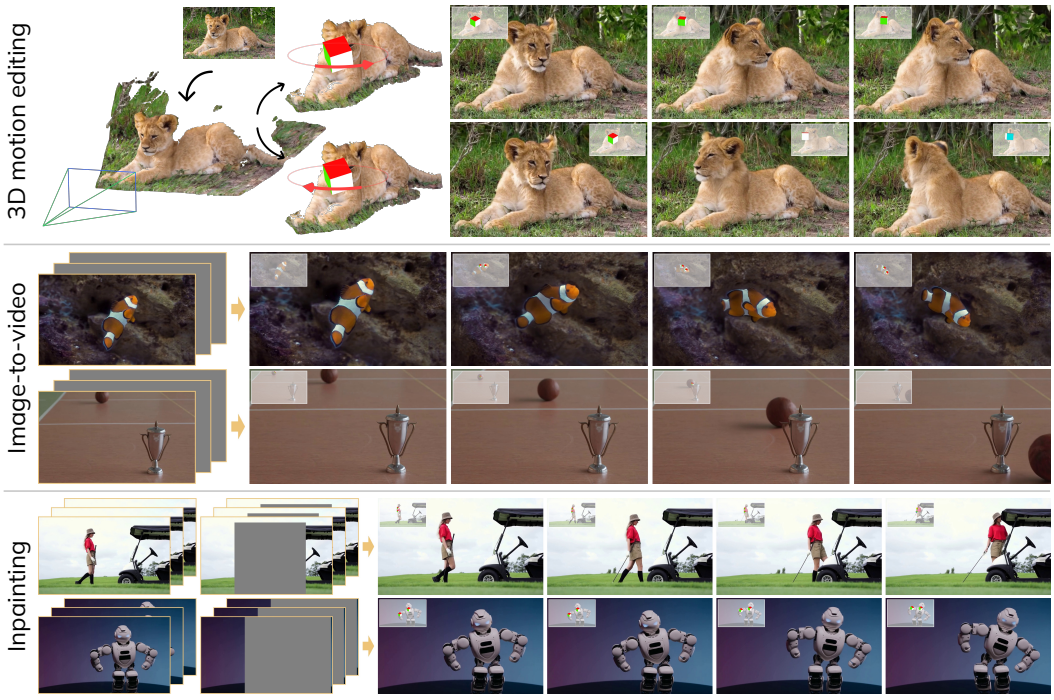

Figure 1: **6-DoF Motion-Controlled Video Generation and Editing.** Swivel specifies where a subject moves and how it rotates with a handful of 6-DoF cubes. The same control supports rotation-aware 3D motion editing of existing footage (top), image-to-video generation (middle) and inpainting-based motion editing (bottom). Each block shows its inputs and cube edits on the left, and each generated frame shows its cube condition as an inset. The video model is trained only on real-world clips whose 6-DoF motion our pipeline measures, with no synthetic data.

already indicates where a measurement can be obtained. The pipeline we introduce below avoids both limitations, and its estimates remain consistent with those of ProxyPose on the same query points, as Section 4.1 shows.

In this paper we present Swivel, a framework that learns 6-DoF motion control from real video alone. Our data pipeline rests on a second observation: what 3D foundation models (Wang et al., 2026; Yang et al., 2026; Xiao et al., 2025) provide for an ordinary real-world clip suffices to (1) estimate the local 6-DoF motion of a rigid part accurately and at scale and (2) automatically propose the parts on which 6-DoF motion is most meaningful. These models provide trajectories for surface points or template vertices. A single trajectory does not define a rotation, but a group of trajectories that move rigidly together does, and we call such a group an anchor. Swivel proposes anchors without manual specification, and the proposal generalises across categories of subject. For a person, a human mesh recovery model (Yang et al., 2026) fits a template mesh whose vertices partition the surface, so anchors are placed at body joints and each joint rotation is estimated from the vertices around it. For an object no template exists, so a 3D point tracker (Xiao et al., 2025) supplies dense 3D tracks, and anchors are placed where neighbouring tracks preserve a constant mutual distance because constant distance is the signature of a single rigid part. Swivel then verifies every candidate and keeps only the reliable ones.

With this corpus we train a video diffusion model for both image-to-video generation and inpainting-based editing, where a handful of cubes specify the 6-DoF translation and rotation of a subject. To condition the model on this 6-DoF motion signal, we project each cube into the 2D screen space as a coloured cube with depth-correct occlusion. On our test clips the generated subject follows both the position and the rotation of its cubes far more faithfully than existing motion control modalities, with higher fidelity and lower FVD than the baselines, and in a human preference study its results outperform every baseline by a large margin. For videos please refer to our anonymous page.

Table 1: **Comparison with Prior Motion Control Methods.** *Real*: no rendered or generated video in training; DaS mixes rendered clips in, PoseTraj pre-trains on them and MotionV2V trains on generated counterfactuals. *Open Domain*: ordinary video rather than one capture rig or one subject category. *Explicit 6-DoF*: the control sets both the position and the rotation of a box, entity, joint or end-effector. *Part-Level*: the control attaches to parts of a subject rather than to the whole.

Method	Real	Open Domain	Explicit 6-DoF	Part-Level
3DTrajMaster (Fu et al., 2025)	✗	✗	✓	✗
IRASim (Zhu et al., 2025)	✓	✗	✓	✗
Champ (Zhu et al., 2024)	✓	✗	✓	✓
PoseTraj (Ji et al., 2025)	✗	✓	✗	✗
VACE (Jiang et al., 2025)	✓	✓	✗	✗
CineMaster (Wang et al., 2025b)	✓	✓	✗	✗
Motion Prompting (Geng et al., 2025)	✓	✓	✗	✓
Go-with-the-Flow (Burgert et al., 2025)	✓	✓	✗	✓
MotionV2V (Burgert et al., 2026)	✗	✓	✗	✓
Diffusion as Shader (DaS) (Gu et al., 2025)	✗	✓	✗	✓
LooseControlVideo (Bhat et al., 2026)	✓	✓	✓	✗
Swivel (ours)	✓	✓	✓	✓

2 Related Work

Motion Control Modality. Motion-control methods differ in the conditioning signal. Sparse signals are point paths given as drags (Yin et al., 2023; Wu et al., 2024), trajectories (Wang et al., 2024; Zhang et al., 2025; Phung et al., 2026), 3D point tracks (Lee et al., 2026) or Gaussian blobs (Burgert et al., 2026), and none of them constrains orientation. Dense signals are fields over the whole frame, such as warped noise (Burgert et al., 2025), dense point trajectories (Geng et al., 2025) or a rendered 3D tracking video (Gu et al., 2025). They determine orientation through their density but are hard to edit by hand at inference. Human animation conditions on 2D skeletons (Hu et al., 2024) or a template mesh (Zhu et al., 2024), but such templates do not extend to general subjects such as animals and objects. Closest to ours are 3D controls given as entity poses (Fu et al., 2025; Shuai et al., 2025), boxes with a camera path (Wang et al., 2025b; Xing et al., 2025), 3D point trajectories (Zhang et al., 2026a), a 2D trajectory with rotation learned in synthetic pre-training (Ji et al., 2025) or oriented boxes (Bhat et al., 2026). Each uses rendered data, omits rotation or acts on whole subjects without part-level control. As shown in Table 1, our cubes combine real footage, user-set rotation and part-level control.

Sources of Rotation Supervision. For 6-DoF control of video generation, supervision has so far mostly come from a simulator (Fu et al., 2025; Shuai et al., 2025) or from a platform that reports its own pose (Zhu et al., 2025), which brings a domain shift for general video control. Perceptual pipelines label real videos with boxes or tracks (Wang et al., 2025b; Gu et al., 2025; Zhang et al., 2026a), but their training labels omit object rotation. In CineMaster the boxes only translate, and the authors leave orientation control to future work because open-set pose estimators are lacking (Wang et al., 2025b). Object pose estimators require depth, masks or CAD models (Wen et al., 2023; 2024; Ponimatkin et al., 2025). ProxyPose tracks the rotation of a user-designated point by generating a proxy-cube video with a video diffusion model (Zhang et al., 2026b). The per-point diffusion run is expensive at corpus scale and fails when the generated cube is distorted, whereas Swivel annotates all parts of a clip in one pass.

3D Foundation Models. Feed-forward 3D foundation models recover per-frame depth and cameras (Wang et al., 2026), dense 3D point tracks (Xiao et al., 2025) and human template meshes (Yang et al., 2026) from ordinary monocular video. They require no per-object input and scale to large corpora, but their outputs are point positions and template vertices rather than the rotation of a local part. Dynamic reconstruction fits rigid motions to such tracks, but only by optimising each video separately (Wang et al., 2025a; Lei et al., 2025). Swivel establishes that the rotation of local rigid parts is recoverable from these outputs alone. It groups points whose mutual distances stay constant (Ullman, 1979), fits their rotation (Kabsch, 1976) and keeps only verified fits.

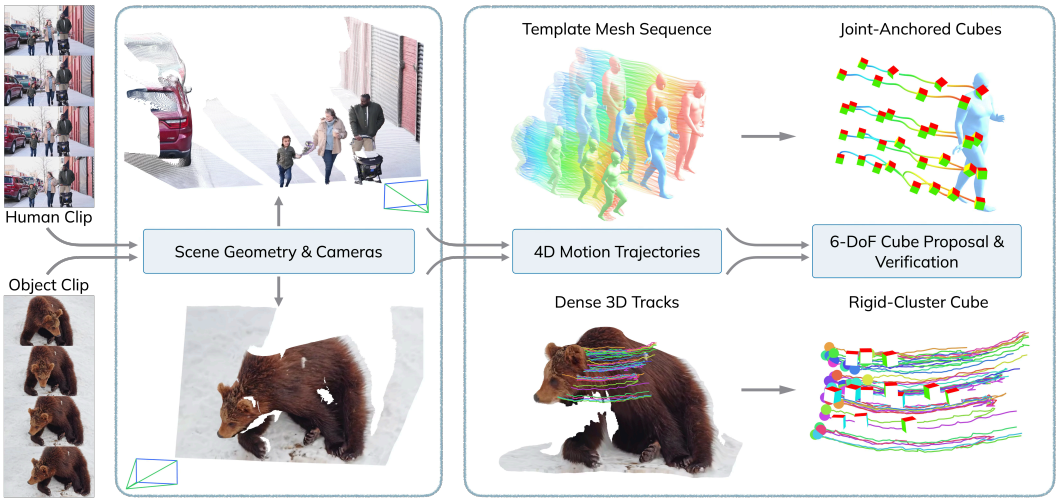

Figure 2: **Swivel Data Curation.** Every clip is first reconstructed into scene geometry and camera motion. The motion of the clip is then recovered as a template mesh sequence for people and as dense 3D tracks for objects. Anchor proposal groups this motion into local rigid parts, at the joints of the mesh and at clusters of tracks that move as one rigid body, and estimates a 6-DoF trajectory for each part, and verification keeps only the trajectories that are reliable. The whole process runs automatically at scale without any human specification. This yields 6-DoF supervision for the video model from real-world clips alone, without any synthetic or domain-specific data.

3 METHOD

Swivel consists of a pipeline that measures the 6-DoF scene dynamics of real videos, as described in Section 3.1. In Section 3.2, we present the video model trained on the resulting real-world clips, with no synthetic or domain-specific data, which provides rotation-aware motion control for image-to-video generation and inpainting-based video editing.

3.1 REAL-WORLD DATA CURATION

Our pipeline annotates general real-world video clips with 6-DoF trajectories at scale. As shown in Figure 2, it reconstructs the scene geometry and cameras of a clip, proposes local rigid anchors on its 3D trajectories, estimates their rotation and verifies every candidate, all without manual intervention.

Scene Geometry and Cameras. Given a real-world video clip of T frames, Pi3X (Wang et al., 2026) estimates per-frame depth, intrinsics $\boldsymbol{K}_t$ and camera poses, which lift every pixel to a 3D point in world coordinates. Since the motion that matters for editing happens on foreground subjects, we apply detection and segmentation models (Liu et al., 2024; Ravi et al., 2025) to separate the foreground from the background as a prefilter. The foreground points form the set $\mathcal{P}$ on which motion is measured, and the background points keep a static position and orientation.

Local Rigid Anchor Proposal. The scene motion is represented by the 3D trajectories $\boldsymbol{x}_i(t)$ of the points $i \in \mathcal{P}$: the vertices of a template mesh sequence (Yang et al., 2026) for people, and dense 3D tracks (Xiao et al., 2025) for animals and other objects without a template. An anchor is a local rigid part $\mathcal{A}_k \subset \mathcal{P}$, a set of points whose mutual distances stay constant over a window $\mathcal{W}_k$, and its 6-DoF trajectory is

$$c_k = \{(\boldsymbol{p}_t, \boldsymbol{R}_t)\}_{t \in \mathcal{W}_k}, \qquad \boldsymbol{p}_t \in \mathbb{R}^3, \quad \boldsymbol{R}_t \in \mathrm{SO}(3), \tag{1}$$

with $\boldsymbol{p}_t$ the centroid of the anchor at frame t and $\boldsymbol{R}_t$ its rotation. For people, an anchor is the vertices around a body joint of the template. For objects, an anchor is grown from a seed: a track a that moves far and smoothly, together with the neighbouring tracks that keep a constant distance to it. How constant the distance $d_i(t) = \|\boldsymbol{x}_i(t) - \boldsymbol{x}_a(t)\|$ stays over time is measured by

$$\alpha_i = \frac{\mathrm{std}_t\ d_i(t)}{\mathrm{mean}_t\ d_i(t)}, \tag{2}$$

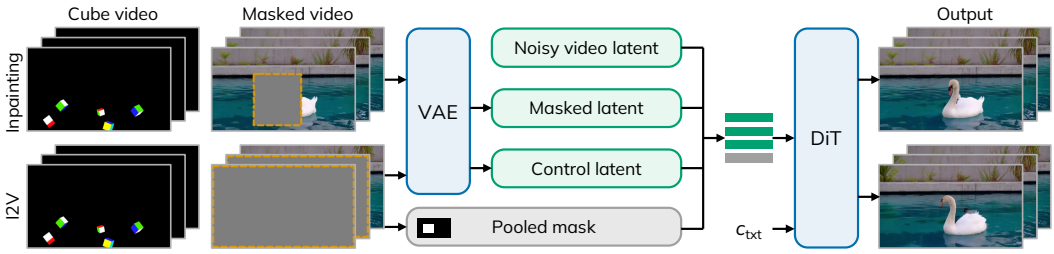

Figure 3: **Video Model Training.** The cube video and the masked clip are encoded by the frozen video VAE, concatenated with the downsampled mask and the noisy latent along the channel axis, and fed to the DiT, which is also conditioned on the text embedding c_{txt}. The two regimes differ only in the mask: inpainting (top) masks a region over part of the clip and regenerates it, while image-to-video (bottom) masks every frame but the first and generates the whole clip.

and the neighbours with the lowest α_i form the anchor together with a.

Rotation Estimation. The rotation of an anchor at frame t is the rotation that best aligns its points at the first frame t_0 of $\mathcal{W}_k$ with its points at frame t, each set centred on its centroid,

$$\boldsymbol{R}_t = \underset{\boldsymbol{R}\in \mathrm{SO}(3)}{\arg\min} \sum_{i\in\mathcal{A}_k} \left\| \boldsymbol{R}\left(\boldsymbol{x}_i(t_0) - \boldsymbol{p}_{t_0}\right) - \left(\boldsymbol{x}_i(t) - \boldsymbol{p}_t\right) \right\|^2, \tag{3}$$

and it has a closed-form solution (Kabsch, 1976). Points that do not move with the part, from tracking errors or from a neighbouring part, would bias this fit, so the points with the largest residuals are discarded and the fit is repeated (Chetverikov et al., 2005).

Verification. Verification tests every candidate anchor and keeps only those whose trajectory is reliable, since equation 3 returns a rotation for any point set, rigid or not. An anchor is kept when its fit is good, with a residual small relative to its size, its motion is informative, with a noticeable rotation and the part in frame, and its background is static, so that the rotation is the part's and not the camera's. Object anchors, whose points come from a tracker rather than a template, must in addition reproject onto the observed tracks, be reproduced when re-selected on half of the frames and scored on the other half, be non-planar, and rotate coherently and several times more than the static background. The accepted trajectories $\{c_k\}$ are the output of the pipeline (Figure 2, right).

3.2 Cube-Conditioned Generation

Rotating Cubes. Given the accepted trajectories, each c_k is rendered with the clip's intrinsics as a solid cube centred at $\boldsymbol{p}_t$ and oriented by $\boldsymbol{R}_t$, with six distinctly coloured faces rendered in screen space, so a change of the visible faces is the measured rotation. As shown in Figure 1, the cube thus encodes an orientation that a point or a track does not express, and a sparse set of cubes specifies complex non-rigid motion.

Video Model Training. Figure 3 shows our training architecture. The pretrained video VAE (Team Wan et al., 2025) encodes the source clip, the rendered cube video and the masked clip into latents $\boldsymbol{z}$, $\boldsymbol{z}^c$ and $\boldsymbol{z}^m$, and the transformer receives these latents together

$$\boldsymbol{u}_\tau = \left[\, \boldsymbol{z}_\tau,\ \boldsymbol{z}^c,\ \bar{\boldsymbol{m}},\ \boldsymbol{z}^m \,\right] \tag{4}$$

with the noised latent $\boldsymbol{z}_\tau$ and the downsampled mask $\bar{\boldsymbol{m}}$. The transformer is fine-tuned with flow matching (Lipman et al., 2023) on $\boldsymbol{z}_\tau = (1-\sigma_\tau)\,\boldsymbol{z} + \sigma_\tau \boldsymbol{\epsilon}$ to predict $\boldsymbol{\epsilon} - \boldsymbol{z}$. Cube videos and masks are rendered on the fly from random subsets of cubes, and the prompt is dropped for 50% of the samples and the cube video for 10%, to enable classifier-free guidance.

Motion Editing. We provide an editing interface, shown in Figure 10. Given an image or a clip, the pipeline first recovers its scene geometry, and for a clip also its tracks and proposed cubes, so that a cube placed on the image lands on a surface point and starts from the measured motion. The user places cubes with a click and keyframes their translation and rotation. The interface renders the cubes and the mask as in training and passes them to the model for inference.



Figure 4: **Comparison with ProxyPose.** Left: our cube in green and the cube ProxyPose tracks for the same query point in orange, drawn at the same size on the same frames. Right: rotation between consecutive frames for two clips. ProxyPose spikes where its tracker slips, while ours stays smooth.

4 Experiments

Training. We fine-tune Wan 2.2 14B (Team Wan et al., 2025) on 150k curated real-world clips, balanced across subject categories such as people, animals, vehicles and objects, with cube videos and holes rendered on the fly. We fine-tune all parameters of both the high-noise and low-noise experts. Training runs on 81-frame clips, first for one epoch at 240p and then at 720p, with AdamW at a learning rate of 10^{-5} and a batch size of 16, and takes about 10 days per model on H200 GPUs. At inference, the model samples 30 steps to generate videos at the full resolution of 1248×720.

Baselines. We compare with publicly available methods. VACE (Jiang et al., 2025) is a general video editing model that we drive with the subject's per-frame 2D box, so it receives the position and extent of the subject but no orientation or depth. Diffusion as Shader (DaS) (Gu et al., 2025) controls motion with a dense tracking video, which we render from the 3D tracks our cubes are built from, and supports image-to-video only. 3DTrajMaster (Fu et al., 2025) takes the 6-DoF pose of the subject's cube in the camera frame with the caption. It controls a subject as one rigid pose per frame, whereas our control is point-based, with several cubes on the parts of a subject, and it supports text-to-video only, so we compare with it qualitatively. Every method is run on the same real test clips, covering people, animals, vehicles and rigid objects, in each setting it supports. Further experimental details are given in Appendix D.

Metrics. We measure how closely the generated subject follows its cubes by its position error. The position error tracks each cube's anchor on the generated video with CoTracker3 (Karaev et al., 2025) and measures its distance to the cube's projected centre, reported as the median in pixels and as the fraction of frames within 32 pixels. We report PSNR, SSIM and LPIPS (Zhang et al., 2018) inside the generated region, and FVD (Unterthiner et al., 2018) against the source videos. Rotation accuracy of the generated video is evaluated qualitatively and in the user study.

4.1 Annotation versus ProxyPose

ProxyPose treats 6-DoF pose estimation as a video-to-video task: the user labels a query point on the first frame, a video diffusion model generates a video of a coloured proxy cube at that point, and tracking the cube gives the pose. We run its 14B model on 60 query points from 20 test clips of diverse categories, and compare its pose with the cube our pipeline produces for the same point. ProxyPose falls behind in three respects. (1) Reliability: ProxyPose fails when the proxy cannot be detected in the first frame of its generated video, which happens for query points near the frame border and for proxies the diffusion model draws poorly, and 5% of the runs return no pose, whereas our pipeline computes the cube from geometry and has no such failure. (2) Cost: ProxyPose runs a 50-step video diffusion once per query point, 962 seconds per point on one RTX Pro 6000 GPU, whereas our pipeline produces all cubes of a clip in one pass, 39 seconds per cube on the same device, about 25 times faster. (3) Automation: ProxyPose requires the user to specify each query point, whereas our pipeline proposes the cubes itself and ranks them by rotation, so the informative ones are selected automatically.

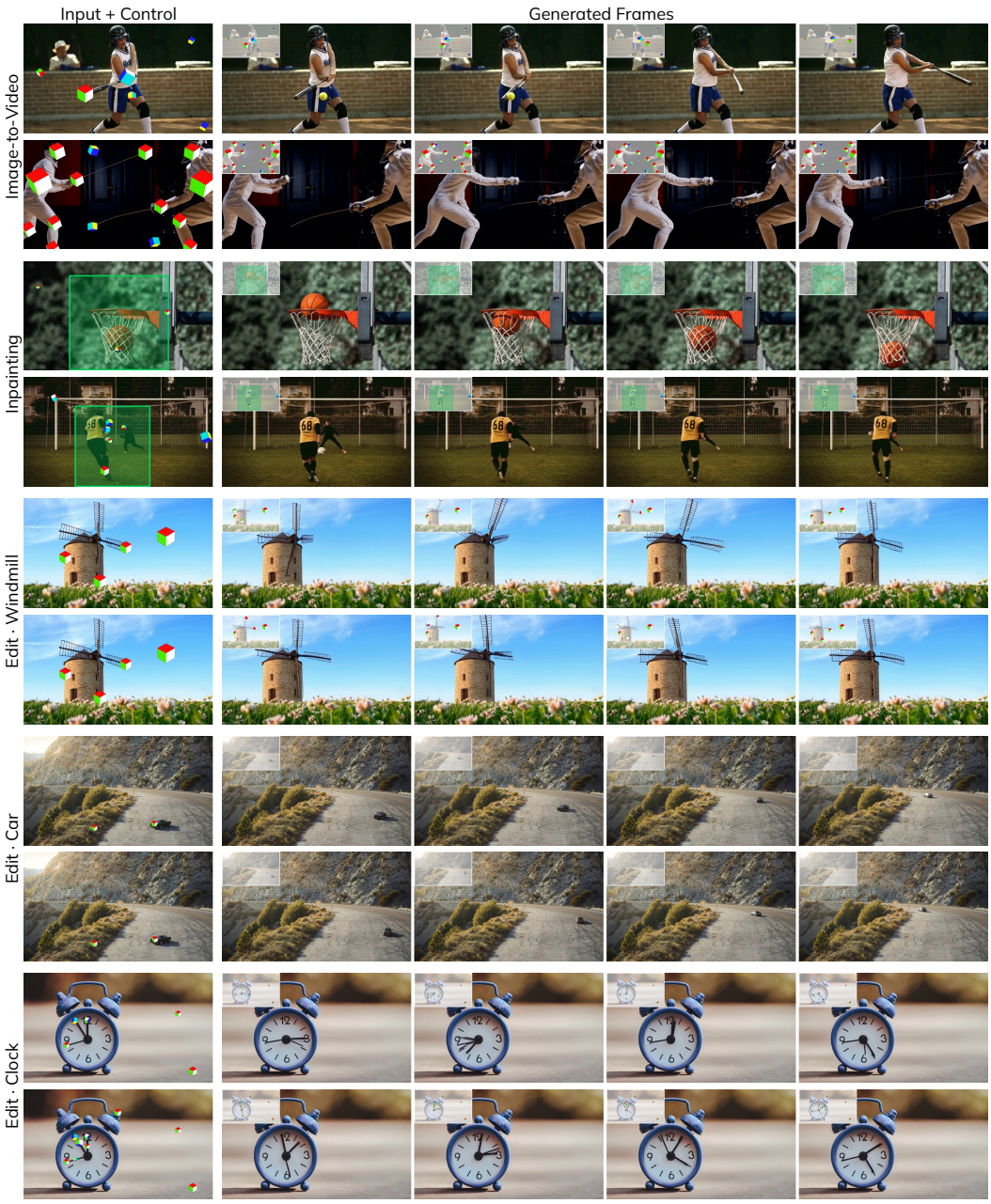

Figure 5: **Editing Results.** Left: the input frame with the user's cubes. Right: generated frames with the cube condition inset. Top: image-to-video. Middle: inpainting, where the green region is regenerated. Bottom: one frame and one prompt edited with two cube trajectories.

On the 57 points where both methods return a pose, the two estimates are consistent, with a median disagreement of 10 degrees in rotation and 10 pixels in position, and Figure 4 and Figure 13 show examples. Notably, the two differ in smoothness. The per-frame rotation of ProxyPose jumps by more than 15 degrees in 14% of the runs, whereas ours does not, and its frame-to-frame variation is four times larger, as the right panel of Figure 4 shows. See our anonymous page for video results.

4.2 Generation Results and Baseline Comparisons

Our model provides general 6-DoF motion control across diverse categories, covering people, animals and rigid objects. Figure 5 shows edits made in the interface of Appendix C, and the last six

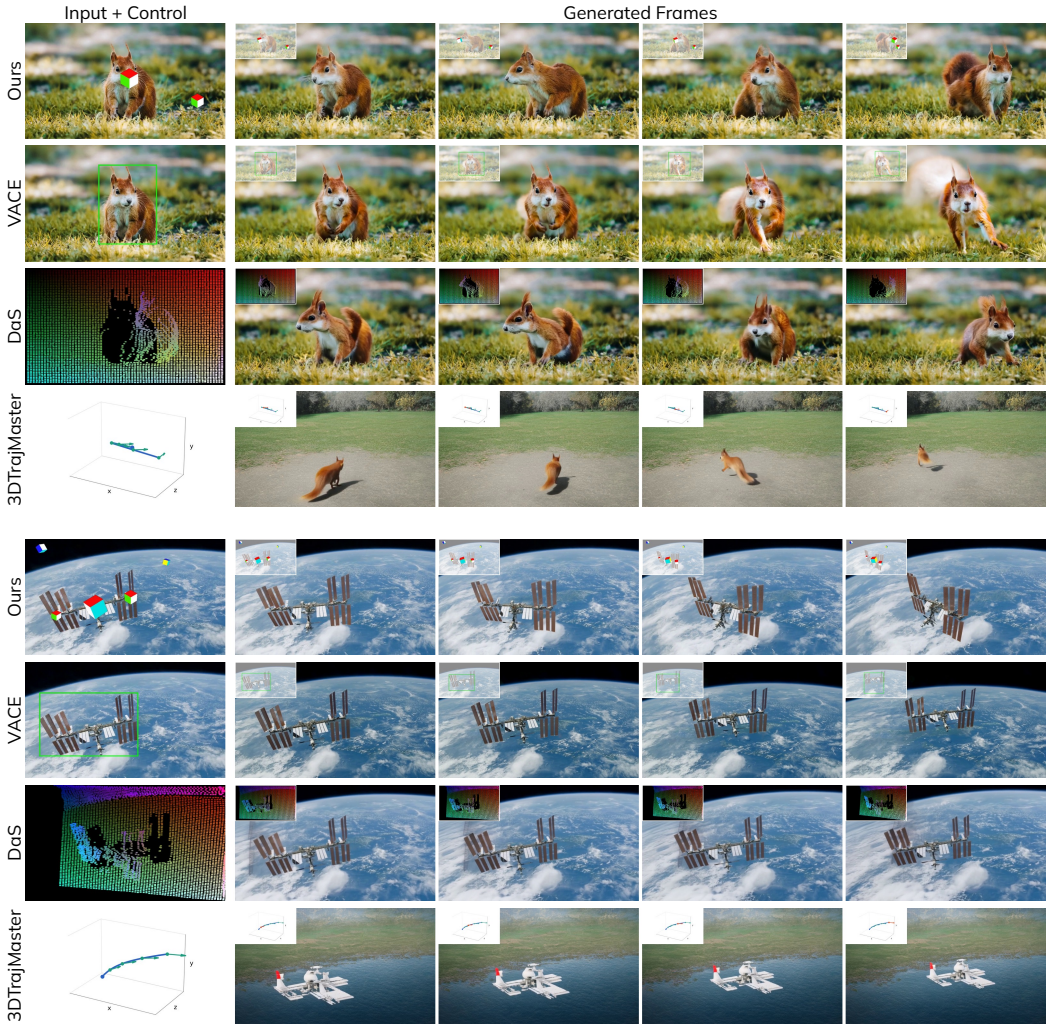

Figure 6: **Qualitative Comparison with Baselines.** All methods start from the same first frame. Each row shows the method's input and control, then generated frames with the control inset. Top: a squirrel turns its head and body. Bottom: a space station turns while drifting across the frame.

rows edit the same scene with two different motions. Only the cubes change, and the motion changes with them: the windmill reverses direction, the car takes a different line through the turnout, and the clock hands advance by a different amount. See our anonymous page for video results.

Table 2 compares the methods on generation quality with objective metrics. In both settings our model keeps the subject closest to its cubes, about 14 pixels against a floor of 10, while the error of DaS and VACE is two to four times larger, and our method leads on every other metric. Since objective metrics do not fully reflect generation quality and motion following, and human judgement remains the more reliable measure of generated video, we also report a human evaluation in Figure 7. We evaluate three dimensions: motion following, whether the video follows the specified motion, subject consistency, whether the edited subject keeps its identity and shape, and overall quality. The human study shows that our result is preferred over every baseline on all three dimensions by a wide margin. The lead is largest on motion following, and smallest on subject consistency against VACE, which edits inside a box and leaves the rest of the frame untouched. The protocol is described in Appendix E. Figure 6 shows a qualitative comparison with the baselines. Our model follows both the path and the turn of the subject. VACE keeps the subject inside its box but loses its shape and orientation, DaS follows the tracks but deforms the subject, and 3DTrajMaster generates the scene purely from text, and its visual quality is lower comparing to our method and other baselines.

Table 2: **Quantitative Baseline Comparisons.** Position: median distance in pixels between the tracked subject and its cubes. In-32: fraction of frames in which that distance is below 32 pixels. Tracking the source video with the same procedure gives a floor of about 10 pixels. PSNR, SSIM and LPIPS are computed inside the generated region: for image-to-video it is all frames except the conditioning frame, and for inpainting it is the masked region.

Method	Position ↓	In-32 ↑	PSNR ↑	SSIM ↑	LPIPS ↓	FVD ↓
Image-to-video						
VACE (Jiang et al., 2025)	52.0	0.41	17.0	0.643	0.253	460.9
DaS (Gu et al., 2025)	29.2	0.67	20.8	0.758	0.153	353.0
Swivel (ours)	**14.6**	**0.91**	**21.8**	**0.800**	**0.115**	**263.9**
Inpainting						
VACE (Jiang et al., 2025)	31.3	0.61	19.2	0.757	0.203	265.6
Swivel (ours)	**13.8**	**0.94**	**21.5**	**0.811**	**0.135**	**190.9**

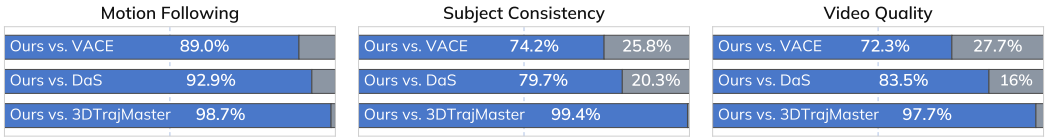

Figure 7: **2AFC User Study.** Share of comparisons in which our result (blue) or the baseline (grey) is preferred. 30 participants each judged 48 video pairs. The evaluated dimensions are motion following, subject consistency and overall quality.

4.3 Ablation on the Number of Cubes

A user places only a few cubes on a subject, so the model must follow a sparse control as well as a dense one. This ablation tests whether the results depend on how many cubes the model is given. Training draws one to ten cubes per subject, and the evaluation in Table 2 uses a denser draw. Table 3 repeats the evaluation with the training draw, a random draw, and every verified cube, from about 6 to 16 cubes per clip. Across the four settings the position error varies by at most 0.7 pixels in image-to-video and 0.2 pixels in inpainting, and FVD by at most nine points, while DaS and VACE sit at 29.2 and 52.0 pixels. The model follows whatever cubes it is given, sparse or dense, which is what makes a handful of user-placed cubes sufficient.

		Image-to-video		Inpainting	
Cube selection	Cubes	Pos. ↓	FVD ↓	Pos. ↓	FVD ↓
Training draw	6.5	15.3	263.6	13.7	188.7
Random draw	10.7	14.7	255.3	13.6	198.1
Evaluation draw	12.2	14.6	263.9	13.8	190.9
All verified cubes	16.2	15.1	255.9	13.8	193.6

Table 3: **Ablation on Cubes Density.** Four ways of choosing the cubes a clip receives, from sparse to dense, everything else fixed. Cubes: average per clip.

5 Conclusion

We presented Swivel, a framework that learns 6-DoF motion control from real-world video alone. Its pipeline recovers the 6-DoF trajectories of local rigid parts of people and objects from ordinary clips, verifies them, and renders them as rotating cubes, a sparse control that carries orientation. A video diffusion model trained on these cubes follows both the position and the rotation a user specifies, in image-to-video generation and in inpainting-based editing, more faithfully than prior motion controls and with a clear preference in human evaluation. The pipeline keeps only parts that move rigidly and therefore discards many object clips, and the control does not yet cover camera motion. Extending the annotation to non-rigid motion and adding camera control on top of the same cubes are directions for future work.

AI USE STATEMENT

We used AI tools to help implement our codebase, process experimental results and caption and filter the training data. We have not used them to design the method or the experiments or to interpret the results, and the remaining tasks that require disclosure do not apply to this work. Additionally, we used them to help with rendering, polish the writing and check it for errors. We have reviewed all AI-assisted work and take responsibility for the final content of this work.

ETHICS STATEMENT

Our models are trained on publicly available video datasets, and some of the clips contain people. We process and filter these clips internally and remove sensitive content before training. The data are used only to train and evaluate our models. The user study collected anonymous preference judgements.

REPRODUCIBILITY STATEMENT

Sections 3 and 4 and Appendices B and D describe the pipeline, the model, the training and the evaluation. The supplementary material provides example outputs of our pipeline and the code of the video model.

REFERENCES

Shuai Bai et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Shariq Farooq Bhat, Niloy J. Mitra, and Kalyan Sunkavalli. Loosecontrolvideo: Directorial video control using spatial blocking. *arXiv preprint arXiv:2606.19495*, 2026.

Ryan Burgert, Yuancheng Xu, Wenqi Xian, Oliver Pilarski, Pascal Clausen, Mingming He, Li Ma, Yitong Deng, Lingxiao Li, Mohsen Mousavi, Michael Ryoo, Paul Debevec, and Ning Yu. Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Ryan Burgert, Charles Herrmann, Forrester Cole, Michael S. Ryoo, Neal Wadhwa, Andrey Voynov, and Nataniel Ruiz. MotionV2V: Editing motion in a video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35988–35997, 2026.

Dmitry Chetverikov, Dmitry Stepanov, and Pavel Krsek. Robust Euclidean alignment of 3D point sets: The trimmed iterative closest point algorithm. *Image and Vision Computing*, 23(3):299–309, 2005. doi: 10.1016/j.imavis.2004.05.007.

Xiao Fu, Xian Liu, Xintao Wang, Sida Peng, Menghan Xia, Xiaoyu Shi, Ziyang Yuan, Pengfei Wan, Di Zhang, and Dahua Lin. 3DTrajmaster: Mastering 3D trajectory for multi-entity motion in video generation. In *International Conference on Learning Representations (ICLR)*, 2025.

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

Zekai Gu, Rui Yan, Jiahao Lu, Peng Li, Zhiyang Dou, Chenyang Si, Zhen Dong, Qifeng Liu, Cheng Lin, Ziwei Liu, Wenping Wang, and Yuan Liu. Diffusion as shader: 3d-aware video diffusion for versatile video generation control. In *ACM SIGGRAPH 2025 Conference Papers*, 2025.

Li Hu, Xin Gao, Peng Zhang, Ke Sun, Bang Zhang, and Liefeng Bo. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8153–8163, 2024.

Longbin Ji, Lei Zhong, Pengfei Wei, and Changjian Li. PoseTraj: Pose-aware trajectory control in video diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. VACE: All-in-one video creation and editing. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17191–17202, 2025.

W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, 1976. doi: 10.1107/S0567739476001873.

Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker3: Simpler and better point tracking by pseudo-labelling real videos. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Yao-Chih Lee, Zhoutong Zhang, Jiahui Huang, Jui-Hsien Wang, Joon-Young Lee, Jia-Bin Huang, Eli Shechtman, and Zhengqi Li. Generative video motion editing with 3D point tracks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Jiahui Lei, Yijia Weng, Adam Harley, Leonidas Guibas, and Kostas Daniilidis. MoSca: Dynamic gaussian fusion from casual videos via 4D motion scaffolds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6165–6177, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023. URL https://arxiv.org/abs/2210.02747.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *European Conference on Computer Vision (ECCV)*, 2024.

Quynh Phung, Long Mai, Cusuh Ham, Feng Liu, Jia-Bin Huang, and Aniruddha Mahapatra. TRACE: Object motion editing in videos with first-frame trajectory guidance. *arXiv preprint arXiv:2603.25707*, 2026.

Georgy Ponimatkin, Martin Cífka, Tomáš Souček, Médéric Fourmy, Yann Labbé, Vladimir Petrik, and Josef Sivic. 6d object pose tracking in internet videos for robotic manipulation. In *ICLR*, 2025. URL https://arxiv.org/abs/2503.10307.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations (ICLR)*, 2025.

Xincheng Shuai, Henghui Ding, Zhenyuan Qin, Hao Luo, Xingjun Ma, and Dacheng Tao. Free-form motion control: Controlling the 6D poses of camera and objects in video generation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12449–12458, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint*, 2025. URL https://arxiv.org/abs/2503.20314.

Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 203(1153):405–426, 1979. doi: 10.1098/rspb.1979.0006.

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

Qianqian Wang, Vickie Ye, Hang Gao, Weijia Zeng, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4D reconstruction from a single video. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025a.

Qinghe Wang, Yawen Luo, Xiaoyu Shi, Xu Jia, Huchuan Lu, Tianfan Xue, Xintao Wang, Pengfei Wan, Di Zhang, and Kun Gai. CineMaster: A 3D-aware and controllable framework for cinematic text-to-video generation. In *ACM SIGGRAPH 2025 Conference Papers*, 2025b. doi: 10.1145/3721238.3730755.

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

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, 2024.

Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Müller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. In *CVPR*, 2023. URL https://arxiv.org/abs/2303.14158.

Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *CVPR*, 2024. URL https://arxiv.org/abs/2312.08344.

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

Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Yuri Makarov, Bingyi Kang, Xing Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. SpatialTrackerV2: Advancing 3D point tracking with explicit camera motion. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Jinbo Xing, Long Mai, Cusuh Ham, Jiahui Huang, Aniruddha Mahapatra, Chi-Wing Fu, Tien-Tsin Wong, and Feng Liu. MotionCanvas: Cinematic shot design with controllable image-to-video generation. In *ACM SIGGRAPH 2025 Conference Papers*, 2025. doi: 10.1145/3721238.3730604.

Xitong Yang, Devansh Kukreja, Don Pinkus, Taosha Fan, Jinhyung Park, Soyong Shin, Jinkun Cao, Jiawei Liu, Nicolas Ugrinovic, Anushka Sagar, Jitendra Malik, Matt Feiszli, Piotr Dollár, and Kris Kitani. SAM 3D Body: Robust full-body human mesh recovery. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7209–7219, 2026.

Shengming Yin, Chenfei Wu, Jian Liang, Jie Shi, Houqiang Li, Gong Ming, and Nan Duan. Dragnuwa: Fine-grained control in video generation by integrating text, image, and trajectory, 2023.

Guiyu Zhang, Yabo Chen, Xunzhi Xiang, Junchao Huang, Zhongyu Wang, and Li Jiang. SymphoMotion: Joint control of camera motion and object dynamics for coherent video generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026a.

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.

Ruihang Zhang, Felix Taubner, Pooja Ravi, Kiriakos N. Kutulakos, and David B. Lindell. Proxypose: 6-DoF pose tracking via video-to-video translation. *arXiv preprint arXiv:2607.06555*, 2026b.

Zhenghao Zhang, Junchao Liao, Menghao Li, Zuozhuo Dai, Bingxue Qiu, Siyu Zhu, Long Qin, and Weizhi Wang. Tora: Trajectory-oriented diffusion transformer for video generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. IRASim: A fine-grained world model for robot manipulation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. URL https://arxiv.org/abs/2406.14540.

Shenhao Zhu, Junming Leo Chen, Zuozhuo Dai, Zilong Dong, Yinghui Xu, Xun Cao, Yao Yao, Hao Zhu, and Siyu Zhu. Champ: Controllable and consistent human image animation with 3d parametric guidance. In *European Conference on Computer Vision (ECCV)*, 2024.

A Additional Visual Results

We refer the reader to our anonymous page for videos. Figures 8 and 14 show the baseline comparison on four more edits, two batters, a space station and a fencer, with every method starting from the same first frame and given the same motion. Figure 13 extends the ProxyPose comparison of Figure 4 to further query points on rigid parts and animals.



Figure 8: **Baseline Comparisons.** Samples: Batters.

B Curated Data

We build the training corpus from publicly available video datasets in four stages: (i) a vision-language model (Bai et al., 2025) annotates every clip with a structured caption and taxonomy labels; (ii) we discard clips with undesired taxonomy flags such as low-quality, NSFW, AI-generated, or lacking motion; (iii) we define categories spanning humans, animals, vehicles, and other objects, and retrieve candidates for each by keyword matching on the captions; (iv) we embed the candidates with Qwen3-VL-Embedding-8B (Bai et al., 2025), remove near-duplicates using embedding clusters, and keep clips whose similarity to per-category positive prompts exceeds that to empirically decided negative prompts.

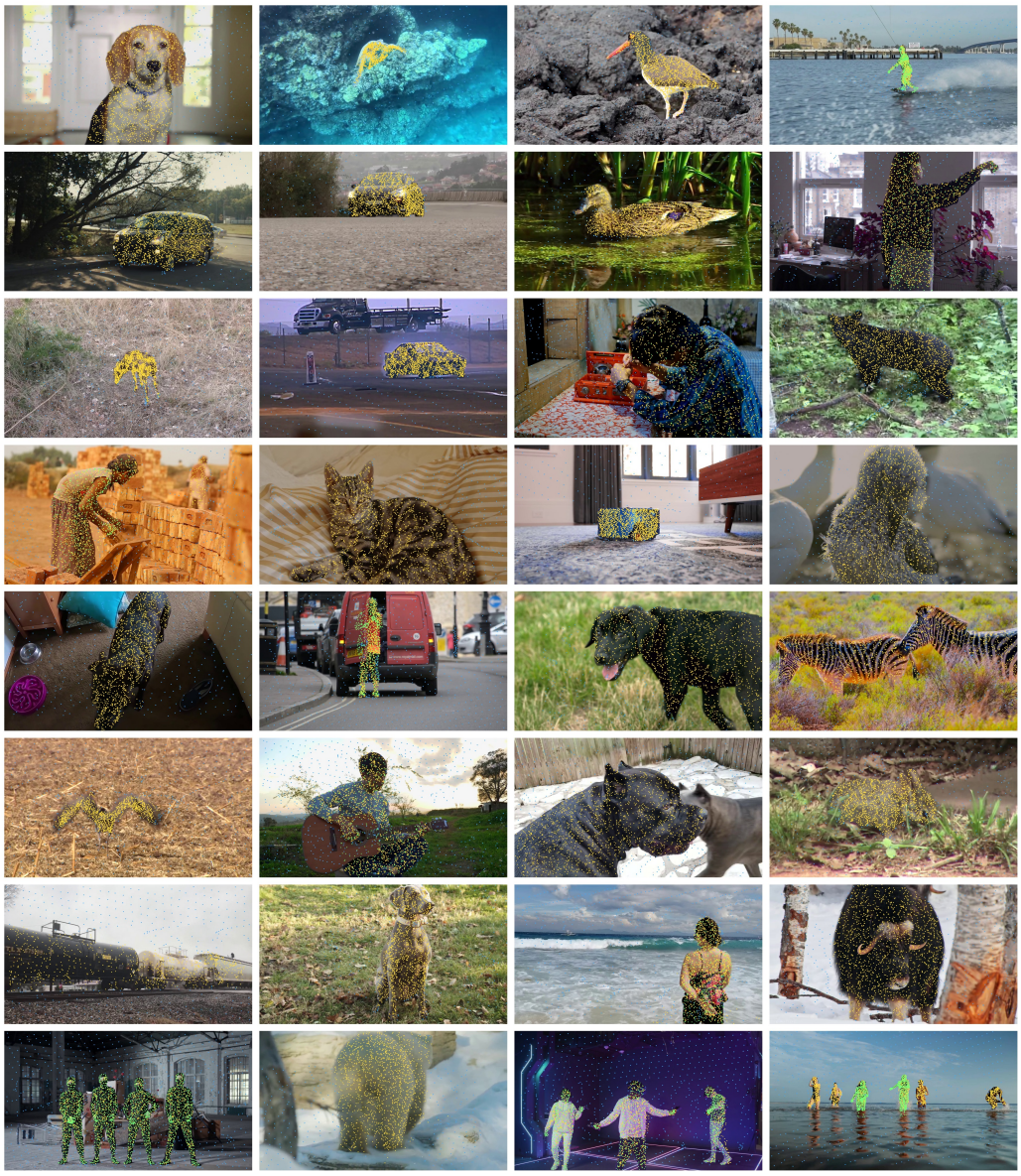

Figure 9: **Curated Real-World Data.** Human clips carry recovered mesh trajectories, object clips (animals, vehicles) dense 3D tracks. Yellow points are foreground trajectories, blue points are background, and green points are the recovered body mesh.

As shown in Figure 9, the final corpus covers people, animals, vehicles and other objects, with mesh trajectories on the human clips and dense 3D tracks on the object clips.

C EDITING INTERFACE

The interface (Figure 10) shows a clip's 3D reconstruction side by side with the original video, onto which all cubes are projected. Users place cubes anywhere in the scene and animate them on a keyframe timeline that stores position and rotation keys separately and interpolates between them with a selectable easing profile. Beyond per-frame poses, a cube can spin continuously about an arbitrary axis (e.g., a windmill hub or a helicopter rotor), or be attached to a parent cube so that it inherits the parent's translation and, optionally, its rotation. The interface is provided in the

supplementary material as an HTML page, together with example sessions that can be opened and edited in it.

A typical session proceeds as follows: (i) place a cube on the target object, or select one that the pipeline initialised at the nearest 3D point under the click, and key its 3D trajectory, checking in the 2D view that the projection moves smoothly and plausibly; (ii) for harder edits such as non-rigid objects or human motion, add more cubes on the object for stronger guidance, either as independent cubes on separately moving parts or as child cubes on parts that move rigidly with an existing one; (iii) add cubes on the background to hold the camera fixed or to specify a camera motion; (iv) draw inpainting masks over regions to be regenerated; (v) export the video, rendered cubes, and mask as a bundle for the model of Section 3.2.

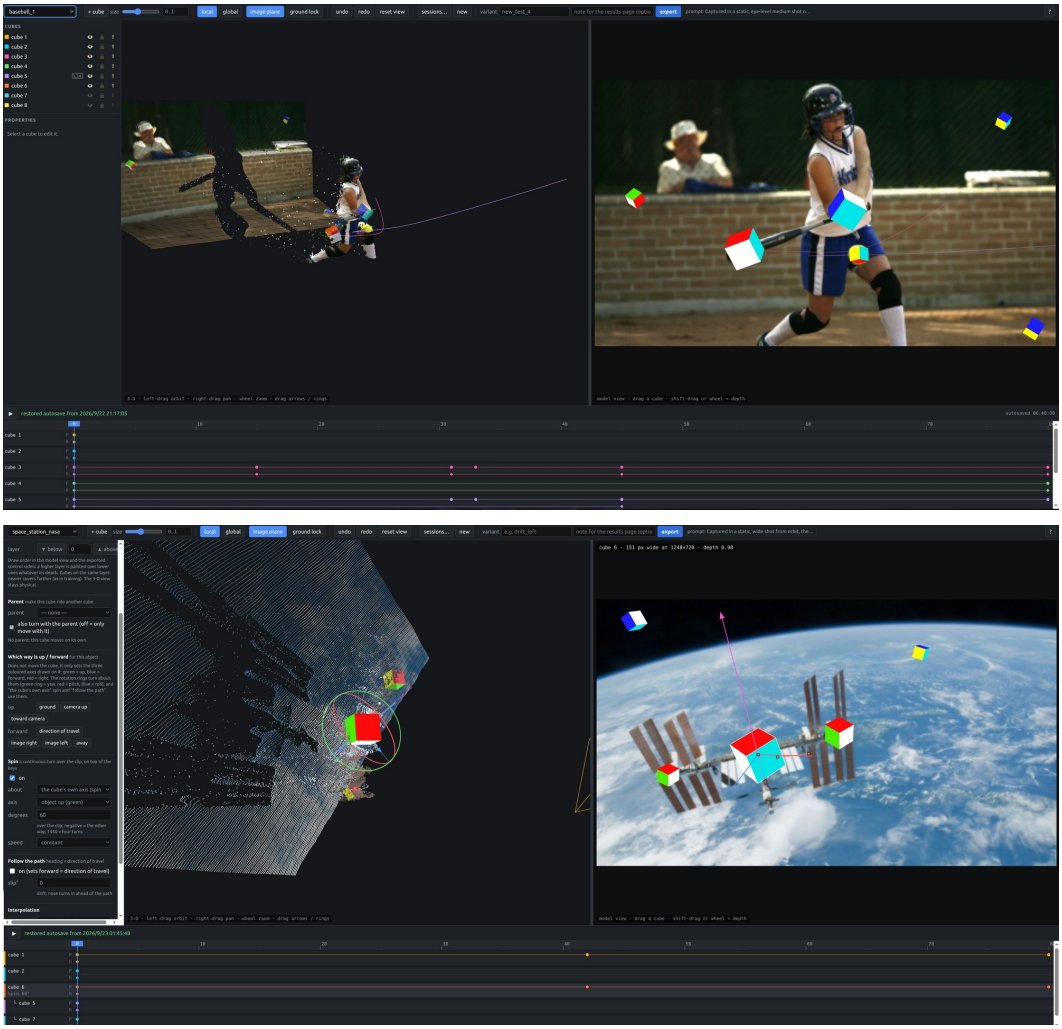

Figure 10: **3D Motion Editing Interface.** Two sessions: a batter and a space station. In each, the 3D viewer on the left shows the clip's reconstruction with its cubes and the rotation rings of the selected cube, the 2D viewer on the right shows the cubes projected on the video, and the timeline below holds the per-frame keys. Users translate and rotate cubes over time and export the condition for the model.

D Evaluation Settings

Test Clips. The 128 test clips have 81 frames at 1248×720: 42 with people, 38 with animals, 29 with rigid objects and 19 with vehicles, drawn at random from held-out data. Inpainting is run on

the 100 test clips whose subject rectangle covers at most three quarters of the frame (30 people, 30 animals, 25 objects, 15 vehicles). The rectangle is the extent of the subject's projected cubes over the clip. The first five frames are kept and the rectangle is filled with grey on the remaining frames, as in training. In image-to-video only the first frame is kept.

Evaluation Cubes. A person receives a cube on seven to ten joints chosen so that head, torso, arms and legs are all covered, drawn with a fixed seed per clip. An object receives all of its verified rigid-part cubes, thinned by farthest-point sampling to 70% when there are more than six, so that they are spread over the object rather than piled on one part. As in training, a few cubes are placed on static background points. Each cube is rendered only on the frames of its verified window.

Sampling Settings. Ours: 30 sampling steps, guidance scale 1, the VAE run on whole frames, and the switch from the high-noise to the low-noise expert at $\sigma_\tau = 0.9$, as in training. VACE, based on Wan2.1-VACE-14B: 30 steps, guidance scale 5. The per-frame 2D box of the subject is drawn into the control frames and the inpainting mask is the same rectangle. DaS, based on CogVideoX-5B image-to-video: 50 steps, guidance scale 6. The tracking video is rendered with SpatialTrackerV2 from the same 3D tracks, and its 49 output frames at 720×480 are resampled to 81 frames at 1248×720 before measurement. 3DTrajMaster, based on CogVideoX-5B with its trajectory injector and LoRA: 50 steps, guidance scale 6, 49 frames at 720×480. It receives one 6-DoF pose per latent frame taken from the subject's cube in the camera frame, and the first sentence of the caption as its entity prompt. All methods receive the same caption.

Metric Details. Position is measured with CoTracker3 (Karaev et al., 2025) started at the first visible anchor of each dynamic cube on the generated video, and the reported error is the median over cubes and frames of the distance to the cube's projected centre, excluding the frames given as input. PSNR, SSIM and LPIPS are computed inside the generated region on the same frames. FVD uses I3D features on 16 frames per clip sampled evenly over the generated frames, at 224×224, between the generated videos and the source videos of the same set.

E User Study

The study pairs our result with VACE, DaS and 3DTrajMaster on 16 edits, 48 pairs in total. Each pair is shown as two videos side by side in random left–right order, first with the cubes drawn on both videos for the question on motion following, then without the cubes and next to the input frame for the questions on subject consistency and overall quality. Figure 11 shows the instructions and Figure 12 one pair. 30 participants judged the 48 pairs in random order.

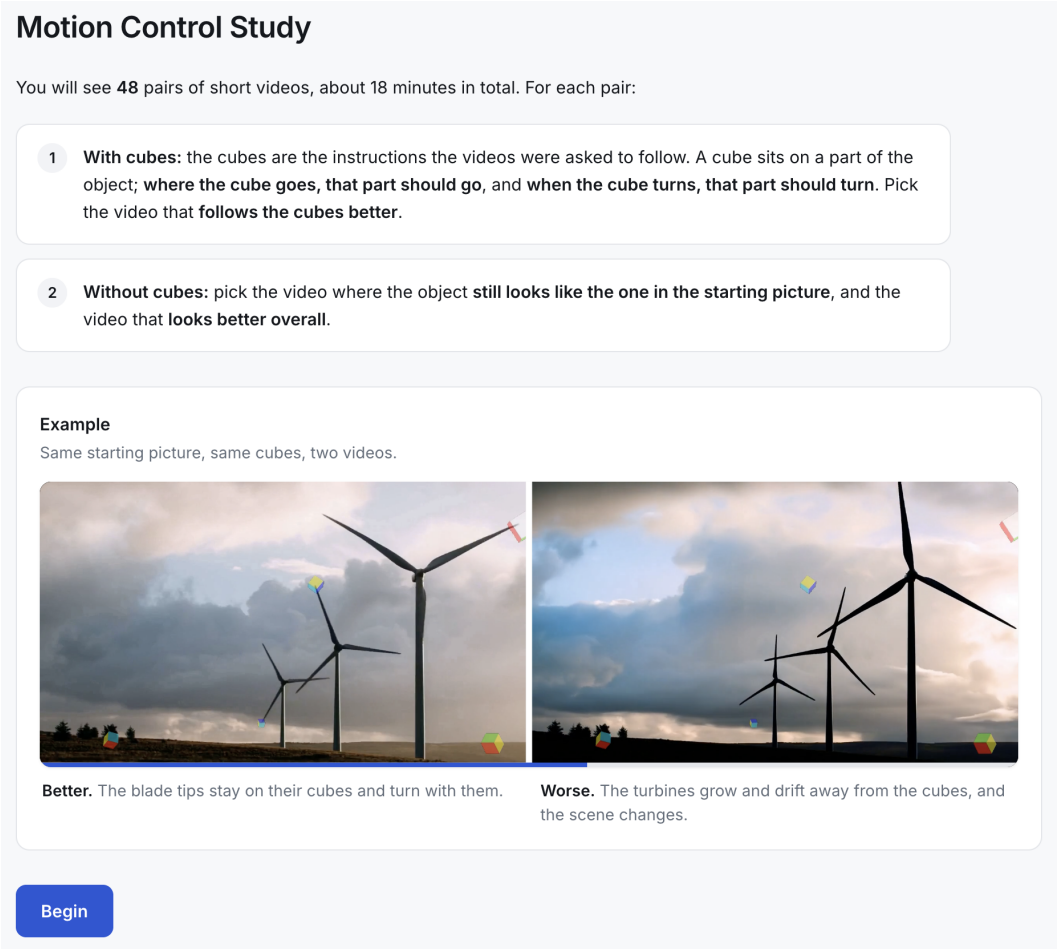

Figure 11: **User Study Instructions.** The page every participant reads first: the three questions to answer, and a worked example with a better and a worse video for the same cubes.

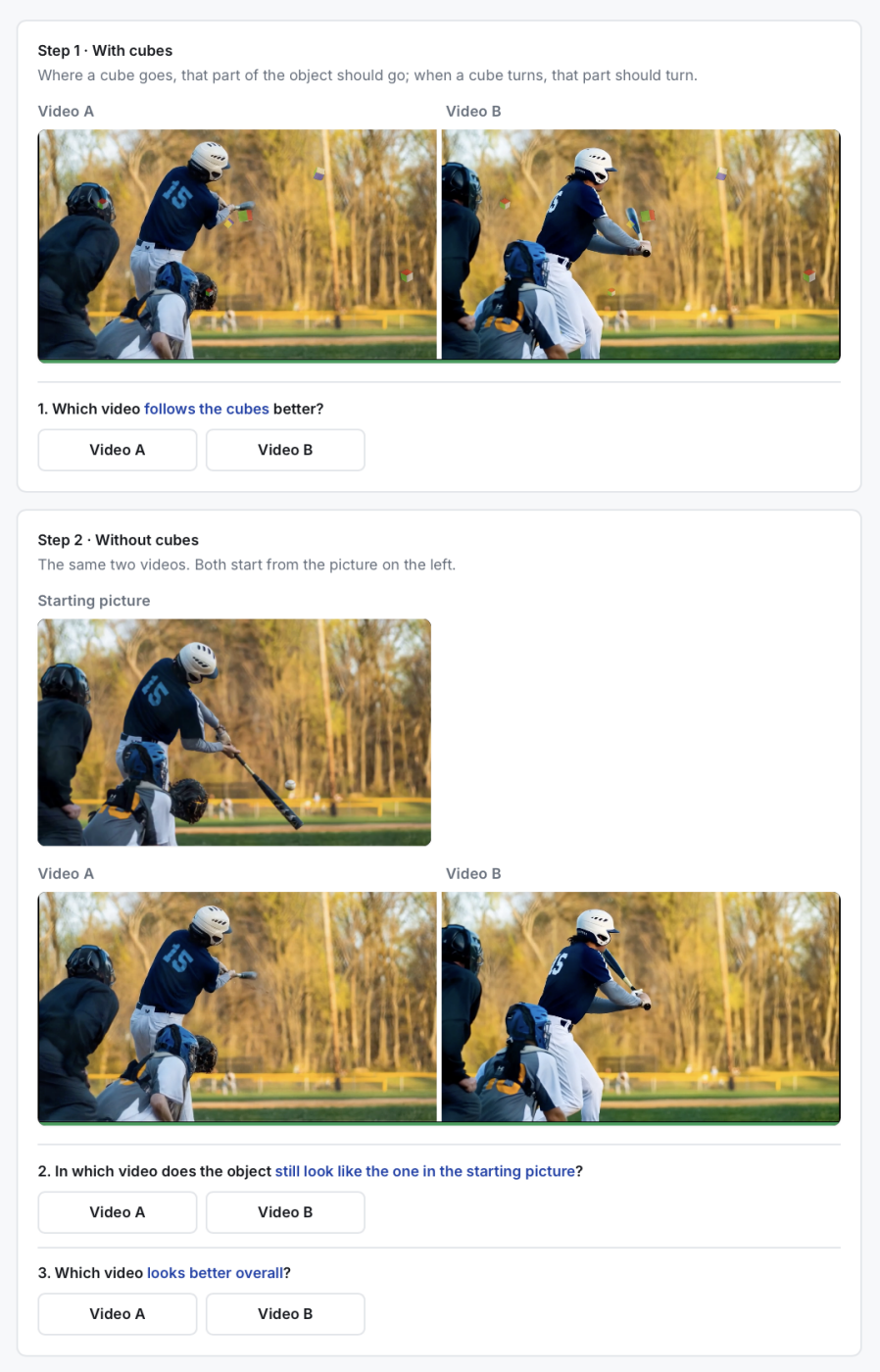

Figure 12: **User Study Interface.** Left, Step 1: the two videos side by side with the cubes drawn on them, and the question on motion. Right, Step 2: the same two videos without the cubes, next to the input frame, and the questions on the subject and on overall quality.

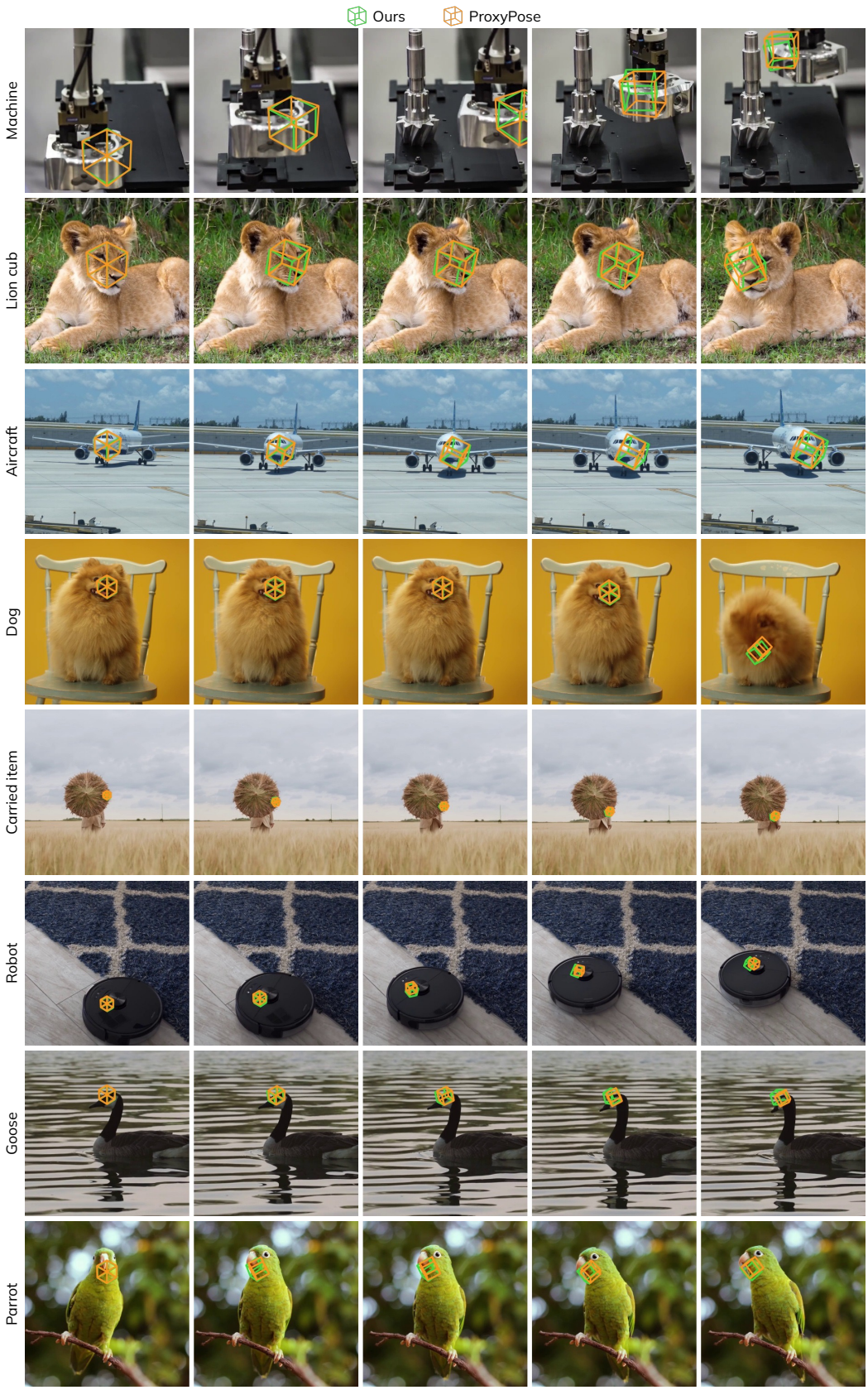

Figure 13: **More Comparisons with ProxyPose.** Further query points, our cube in green and ProxyPose in orange on the same frames.

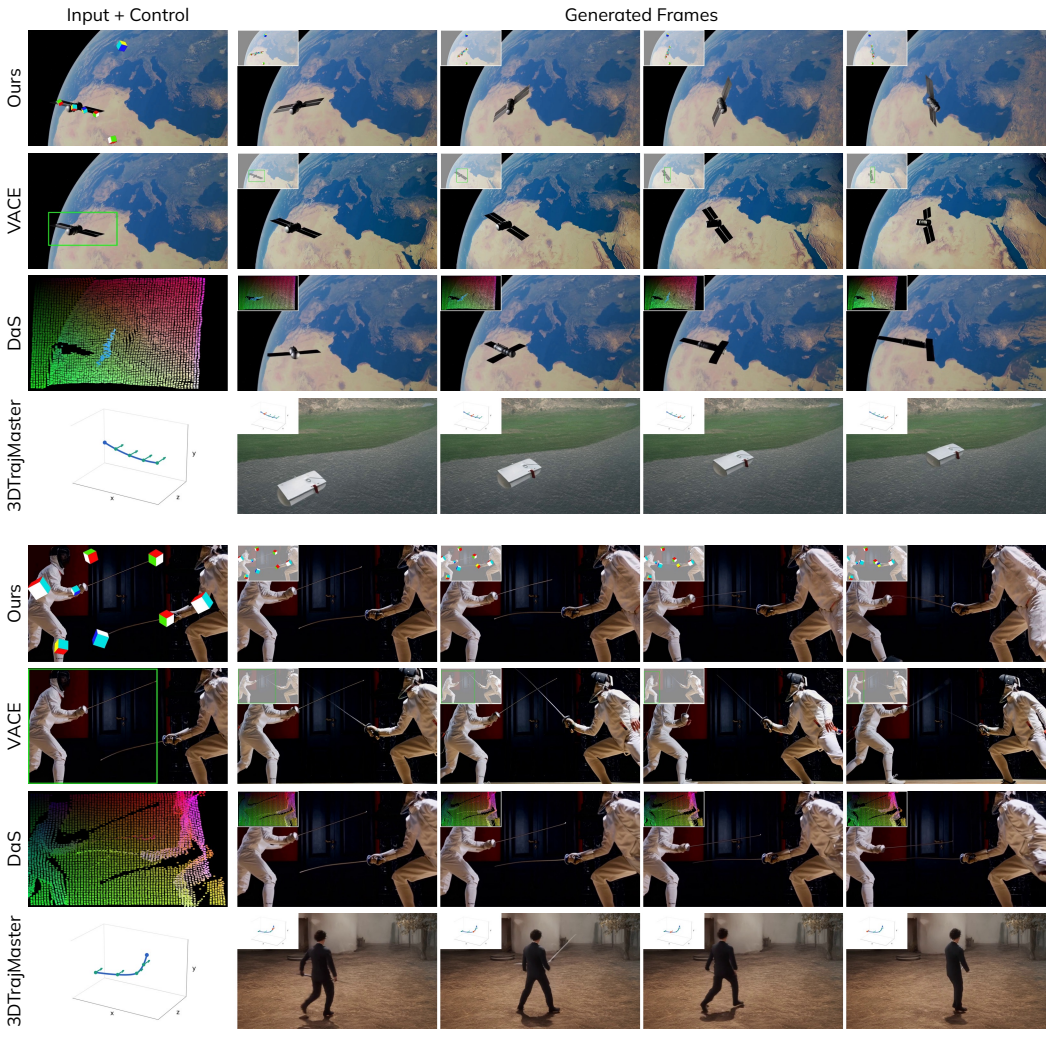

Figure 14: **Baseline Comparisons.** Samples: Space Station, Fencer.