FoMo: Generative 4D Morphing for Motion Tracking

Anonymous authors
Paper under double-blind review

Abstract

We present FoMo, a generative model for 4D motion tracking of a known mesh from monocular video. We formulate the problem as generative 4D morphing natively in 4D vertex trajectory space with flow matching, with FoMo most notably resolving the resulting noise-to-vertex transport ambiguity through a dual-geometry architecture. FoMo significantly outperforms prior methods on mesh tracking and, when applied to skeletal tracking, also surpasses specialized skeletal trackers. It further generalizes to in-the-wild and synthetic videos, enabling tracking of reconstructed real-world meshes, cross-identity motion capture, and the use of video generation models as 4D motion generators.



Figure 1: Given a monocular video and a reference mesh, FoMo generates its 4D motion with vertex correspondence. This example shows FoMo generalizing to generated video and recovering rich dragon motion.

1 Introduction

In this work, we study 4D motion tracking of a known mesh from monocular video. Recent generative approaches have made substantial progress toward holistic 4D content generation from images and videos (Bahmani et al., 2024; Jiang et al., 2023; Liang et al., 2024; Ren et al., 2024; Wu et al., 2024). We focus on the setting where the 4D motion of a provided 3D mesh is recovered while preserving correspondence both across frames and to the reference geometry, a problem that has also been extensively studied (Chen et al., 2025; Jiang et al., 2026; Sabathier et al., 2026; Chen et al., 2026; Wu et al., 2026). Correspondence has long been recognized as a fundamental problem in computer vision (Wang et al., 2019); in our setting, it keeps the recovered motion tied to the reference geometry and directly usable for animation and subsequent manipulation.

Recovering non-rigid 4D motion from monocular video is inherently ill-posed: multiple 3D motions can explain the same 2D observations (Garg et al., 2013; Kokkinos & Kokkinos, 2021;

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.

Tretschk et al., 2023). The problem becomes particularly challenging when the model must operate on arbitrary reference meshes rather than a fixed template or category-specific representation such as SMPL (Loper et al., 2023). Despite substantial progress, existing methods (Jiang et al., 2026; Sabathier et al., 2026; Wu et al., 2026; Chen et al., 2026) can still exhibit implausible motion, severe mesh distortion, and inconsistent geometry recovery under ambiguous observations, large deformation, and occlusion. Supporting variable mesh resolutions and long or varying sequence lengths further complicates the problem.

We take a direct generative view of this ill-posed problem and formulate 4D motion as *generative 4D morphing* natively in vertex trajectory space. Given a video and a reference mesh, a simple generative model directly produces the trajectories of its vertices which explicitly modeling their correspondences, rather than compressing motion into a learned latent representation.

We instantiate this formulation with flow matching (Lipman et al., 2023) and a diffusion transformer (DiT) (Vaswani et al., 2023; Peebles & Xie, 2023) operating directly on vertex trajectories. The model generates the complete $T \times N \times 3$ trajectory tensor, for T frames and N vertices. Direct flow matching in this space, however, differs fundamentally from image generation. Image noise lives on a fixed spatial lattice, where each noisy variable retains an implicit spatial identity throughout generation. Vertex trajectories instead live in continuous 3D space: once trajectories are perturbed toward noise, their noisy positions provide no intrinsic association with the vertices of the reference mesh. The model must therefore transport unstructured noise into structured vertex trajectories while incorporating information from the mesh being morphed.

Our key idea is to use the reference geometry itself to provide persistent vertex correspondence throughout this generative process. FoMo realizes this through a *dual-geometry* architecture while retaining a single set of $T \times N$ generative tokens. Each token is embedded with respect to its evolving position in the generation geometry, representing the trajectory currently being generated. The same token is simultaneously embedded with respect to the fixed reference geometry, through which it incorporates corresponding reference information including shape, appearance, and local topology. The reference and generation geometries thus provide complementary geometric contexts for the same token: one describes the mesh being morphed, while the other describes its evolving motion. This allows reference-mesh structure to continuously inform trajectory generation without introducing a separate latent motion representation.

Operating directly on vertex trajectories also gives FoMo a flexible spatial and temporal representation. The same transformer can process reference meshes with varying numbers of vertices and videos of varying temporal length, without requiring a fixed mesh template or sequence length, and supports a degree of spatial and temporal extrapolation beyond the mesh resolutions and sequence lengths observed during training. Topology-aware reference information and temporal interactions further coordinate motion across the structure of the mesh and over the sequence.

The trajectory formulation further extends naturally from surface vertices to skeletal joints. Both can be represented as structured 3D points whose trajectories describe the underlying motion, allowing FoMo to process mesh and skeletal motion within the same generative formulation rather than treating them as fundamentally separate tasks.

FoMo significantly outperforms prior methods on mesh tracking and, when applied to skeletal tracking, also surpasses specialized skeletal trackers. As shown in Figure 1, it further generalizes to in-the-wild and synthetic videos, enabling tracking of reconstructed real-world meshes, cross-identity motion capture, and the use of video generation models as 4D motion generators.

Our contributions are:

- We formulate 4D motion tracking as *generative 4D morphing* directly in vertex trajectory space, preserving correspondence to an arbitrary reference mesh throughout generation.
- We introduce FoMo, a flow-matching model for direct trajectory generation, with a dual-geometry architecture that integrates reference shape, appearance, and topology with the evolving generation geometry.
- We demonstrate that this formulation achieves state-of-the-art 4D mesh tracking, generalizes across mesh resolutions and sequence lengths, and extends directly to skeletal tracking, where FoMo surpasses specialized methods.

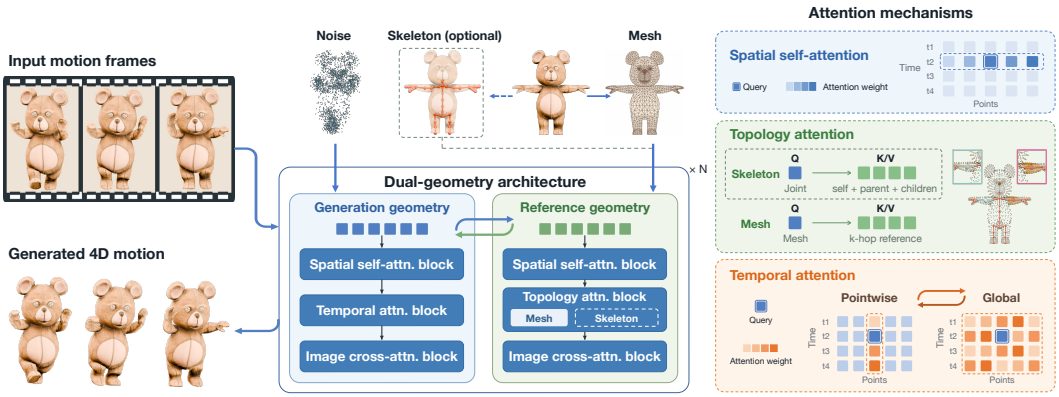

Figure 2: Overview of FoMo. Given a reference geometry and a driving video, FoMo generates a 3D trajectory for each reference point. The same trajectory tokens are processed under a fixed reference geometry and an evolving generation geometry throughout denoising.

2 Related Work

2.1 4D Generation without Tracking

Previous 4D generation methods synthesize dynamic 3D content from text, images, or monocular video using representations such as radiance fields and 3D Gaussians. One line of work distills image- or video-diffusion priors through score distillation sampling (SDS) (Bahmani et al., 2024; Ling et al., 2024; Jiang et al., 2023; 2024; Zhao et al., 2024), while another directly predicts dynamic 4D representations with feed-forward generative models (Liang et al., 2024; Zhang et al., 2024; Wu et al., 2024; Xie et al., 2025; Yao et al., 2025; Ren et al., 2024; Zhang et al., 2025; Li et al., 2025). These works primarily address holistic 4D content generation, while the setting considered in this work additionally requires persistent correspondence to a provided reference mesh.

2.2 4D Mesh Tracking with Monocular Video

Recent video-to-mesh methods directly track and animate dynamic 3D meshes from video, which maintain or recover correspondence through different mechanisms. V2M4 (Chen et al., 2025) adopts an optimization-based reconstruction pipeline; Mesh4D (Jiang et al., 2026), Motion 3-to-4 (Chen et al., 2026), R-DMesh (Wu et al., 2026), and related methods (Shi et al., 2025) use learned motion representations; ActionMesh (Sabathier et al., 2026) and ShapeGen4D (Yenphraphai et al., 2025) recover correspondence through additional post-processing. FoMo is likewise an end-to-end generative approach, but operates directly in vertex trajectory space, predicting correspondence-preserving motion without an intermediate learned motion representation.

2.3 Category-Agnostic Skeletal Tracking

Monocular 3D skeletal tracking has traditionally focused on humans, where methods such as SMPL-based reconstruction and video pose estimation assume a fixed body model and kinematic structure (Loper et al., 2023; Kocabas et al., 2020; Kanazawa et al., 2018). More recent category-agnostic motion-capture methods condition on an arbitrary reference skeleton. MoCapAnything (Gong et al., 2026a) predicts 3D skeletal motion from video together with reconstructed 4D geometry, while MoCapAnything V2 (Gong et al., 2026b) and TopoCap (Pu et al., 2026) learn motion priors that generalize across skeleton topologies. These methods are designed specifically for skeletal motion. FoMo represents skeletal joints with the same point-trajectory formulation used for mesh vertices, allowing skeletal and mesh tracking to be handled within a shared generative model.

3 METHOD

FoMo tracks a reference mesh in a monocular video by directly generating the 3D trajectory of each vertex. We formulate this as *generative 4D morphing* in vertex trajectory space and instantiate it with flow matching. The resulting point tokens are processed by a dual-geometry transformer, where the same token is embedded under both the fixed reference geometry and the evolving generation geometry. Figure 2 gives an overview.

3.1 TRAJECTORY REPRESENTATION AND TOKENIZATION

Given a reference mesh with N vertices and a video of T target frames, we predict vertex positions $\mathbf{P}^{1:T} \in \mathbb{R}^{T\times N\times 3}$. If a reference skeleton with K joints is available, we also predict joint positions $\mathbf{Q}^{1:T} \in \mathbb{R}^{T\times K\times 3}$. We denote the complete target trajectory by $\mathbf{X}_1$: it equals $\mathbf{P}^{1:T}$ for mesh-only inputs, or $[\mathbf{P}^{1:T}; \mathbf{Q}^{1:T}]$ when joints are included, concatenated along the point dimension.

Each vertex or joint at each target frame corresponds to a single token. These tokens are the generative variables themselves and directly form the model output; no additional generative tokens are introduced. Mesh vertices and skeletal joints therefore share the same trajectory representation and are processed by the same network.

We write $\mathcal{C}^r$ for the reference-side conditioning, including the reference geometry, its attributes, and reference-image features. We denote driving-video features extracted with DINOv3 (Siméoni et al., 2025) by $\mathcal{I}^{1:T}$.

3.2 GENERATIVE 4D MORPHING WITH FLOW MATCHING

We instantiate generative 4D morphing with flow matching (Lipman et al., 2023), learning a generative transport from Gaussian noise to the complete set of point trajectories. Given a clean trajectory $\mathbf{X}_1$ and independent noise $\mathbf{X}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ of the same shape, we construct

$$\mathbf{X}_\lambda = (1-\lambda)\mathbf{X}_0 + \lambda\mathbf{X}_1, \qquad \mathbf{V}^* = \mathbf{X}_1 - \mathbf{X}_0.$$

Here $\lambda \in [0,1]$ denotes flow time from noise to data and is independent of video time. FoMo predicts the velocity field from the noisy trajectories conditioned on the reference geometry and driving video:

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{\mathbf{X}_0,\mathbf{X}_1,\lambda}\left[\left\|\mathbf{V}_\theta\left(\mathbf{X}_\lambda, \lambda; \mathcal{C}^r, \mathcal{I}^{1:T}\right) - \mathbf{V}^*\right\|_2^2\right].$$

This directly models the distribution of vertex and joint trajectories without introducing an intermediate learned motion space.

3.3 DUAL-GEOMETRY TRANSFORMER

Direct trajectory-space generation poses a positional-encoding problem. Unlike image generation, where tokens lie on a fixed spatial lattice, the coordinates of our point tokens are themselves being noised and generated. A permutation-equivariant transformer therefore has no fixed positional system that associates a noisy token with its corresponding reference point.

FoMo addresses this by embedding each single token under two geometric positional systems: the fixed reference geometry and the evolving generation geometry. The reference geometry uses the clean reference coordinates, providing a persistent geometric association with the corresponding vertex or joint. The generation geometry uses the current noisy coordinates $\mathbf{X}_\lambda$ and represents the trajectory state being generated. We refer to this one-token, two-geometry construction as our *dual-geometry* network design.

Overall structure. We linearly embed the noisy coordinates and process the resulting tokens through L pairs of reference and generation blocks. Within each pair, the reference block first updates the tokens using the fixed reference geometry and reference-side conditioning, after which the generation block updates the same tokens using the current generation geometry and driving video. The two block types use separate parameters and are conditioned on flow time λ. A final linear decoder predicts a 3D flow velocity for each point at each target frame.

Reference Geometry Block. The reference geometry block performs spatial self-attention with 3D rotary position embeddings (3D RoPE) (Su et al., 2023) evaluated at the clean reference coordinates. It additionally aggregates local reference information through topology-aware attention. For mesh vertices, local neighborhoods are defined by mesh connectivity and provide attributes including RGB colors and surface normals. For skeletal joints, local neighborhoods are defined by the skeleton graph, with joint-to-parent vectors and a learned skeleton-type embedding as attributes. Cross-attention to features from the rendered reference image provides additional visual information. The reference coordinates and attributes are shared across all target frames.

Generation Geometry Block. The generation geometry block performs spatial self-attention with 3D RoPE evaluated at the current noisy coordinates in $\mathbf{X}_\lambda$. Cross-attention to the corresponding driving-video frame provides motion evidence, while temporal attention propagates information across target frames. We alternate between point-wise temporal attention, which connects the same vertex or joint across time, and global temporal attention, which captures coordinated motion across points and frames. Both use 1D temporal RoPE.

3.4 Implementation Details

Training setup. We use the 4D mesh dataset from Chen et al. (2026) which contains 16k mesh-animation sequences. We additionally extract valid skeletal animations for training where available from the dataset. We train the model for about 450k steps with a batch size of 32 a learning rate of 10^{-4}. Training uses 8 NVIDIA A800 GPUs for approximately 8 days. The training details are given in Applendix A. .

Inference. Though only trained on limited number of input vertices and frames, FoMo can flexibly extrapolate to variable mesh resolution and video length. It can take all input mesh vertices and generates the full target sequence without vertices sampling or auto-regressive generation during inference. For long clips, global temporal attention can be switched to temporal sliding window attention to reduce memory and computation. The context schedule is given in Appendix C.

4 Experiments

We evaluate FoMo on two complementary tracking tasks: dense 4D mesh tracking, where every native mesh vertex follows the observed motion, and category-agnostic skeleton tracking, where the same formulation predicts the trajectories of an arbitrary reference skeleton.

4.1 Evaluation Protocol

Evaluation datasets. For 4D mesh tracking, we evaluate on **Motion-80** (Chen et al., 2026). The benchmark has ground-truth animated meshes and rendered videos with variable-length sequences. The test set is strictly excluded from our training set with no overlapping models. We report both the full benchmark (*Motion-80-full*) and a 16-frame benchmark (*Motion-80 (16 frames)*), obtained by truncating each target sequence to its first 16 frames. Motion-80 (full) measures complete-sequence tracking, but its long, variable-length sequences require autoregressive generation for baselines that only support short clips. We therefore introduce Motion-80 (16 frames) to fairly compare single-clip generation performance: every method generates the same 16-frame target in one rollout.

We use two reference protocols. In *with pose alignment* (**w/ PA**), following other 4D mesh tracking methods' evaluation protocol (Chen et al., 2026; Sabathier et al., 2026), the first frame of each sequence is used as the reference mesh. In *without pose alignment* (**w/o PA**), we use the same 80 sequences and the rest-pose mesh of each sequence is used as the reference. FoMo only takes and outputs mesh vertices other than skeleton joints.

For category-agnostic skeletal tracking, we evaluate on Motion-80 (16 frames). Only 78 out of 80 with valid skeletal animations are evaluated. Similarly, the w/ PA reference is the first-frame skeleton, while the w/o PA reference is the rest-pose skeleton.

Baselines. For 4D mesh tracking, we compare against ActionMesh (Sabathier et al., 2026), R-DMesh (Wu et al., 2026), and Motion 3-to-4 (Chen et al., 2026). All baselines use their official

Table 1: **Quantitative 4D mesh tracking on Motion-80.** w/ PA uses the first animation frame as the reference mesh, while w/o PA uses the asset's rest-pose mesh as the reference.

Motion-80 (full).

	w/ PA				w/o PA			
Method	$L_1 \downarrow$	$L_2 \downarrow$	VE↓	CD↓	$L_1 \downarrow$	$L_2 \downarrow$	VE↓	CD↓
ActionMesh (Sabathier et al., 2026)	0.258	0.181	0.032	0.180	0.520	0.359	0.044	0.275
R-DMesh (Wu et al., 2026)	0.241	0.170	0.040	0.156	0.388	0.274	0.043	0.219
Motion 3-to-4 (Chen et al., 2026)	0.094	0.066	0.022	0.070	0.374	0.262	0.033	0.208
FoMo (Ours)	**0.079**	**0.057**	**0.020**	**0.064**	**0.190**	**0.138**	**0.025**	**0.110**

Motion-80 (16 frames).

	w/ PA				w/o PA			
Method	$L_1 \downarrow$	$L_2 \downarrow$	VE↓	CD↓	$L_1 \downarrow$	$L_2 \downarrow$	VE↓	CD↓
ActionMesh (Sabathier et al., 2026)	0.121	0.087	0.030	0.090	0.435	0.305	0.049	0.219
R-DMesh (Wu et al., 2026)	0.191	0.136	0.039	0.133	0.346	0.242	0.043	0.197
Motion 3-to-4 (Chen et al., 2026)	0.079	0.055	0.023	0.060	0.332	0.232	0.039	0.191
FoMo (Ours)	**0.056**	**0.040**	**0.017**	**0.050**	**0.176**	**0.125**	**0.024**	**0.098**

Table 2: **Quantitative category-agnostic skeleton tracking on Motion-80 (16 frames).** w/ PA uses the first frame as the reference pose, while w/o PA uses the asset's rest pose as the reference.

	w/ PA		w/o PA	
Method	MPJPE↓	MPJVE↓	MPJPE↓	MPJVE↓
MoCapAnything V2 (Gong et al., 2026b)	0.1198	0.0291	0.1258	0.0299
ActionMesh (Sabathier et al., 2026)	0.0592	0.0219	0.1536	0.0387
R-DMesh (Wu et al., 2026)	0.0838	0.0272	0.1213	0.0300
Motion 3-to-4 (Chen et al., 2026)	0.0464	0.0193	0.1287	0.0302
FoMo (mesh+inverse fit)	<u>0.0382</u>	<u>0.0147</u>	<u>0.0727</u>	<u>0.0192</u>
FoMo (Ours)	**0.0352**	**0.0134**	**0.0659**	**0.0177**

implementations and configurations. For skeletal tracking, we compare with current state-of-the-art category-agnostic skeletal tracking specialist method, MoCapAnything V2 (Gong et al., 2026b). ActionMesh, R-DMesh, and Motion 3-to-4 are also evaluated as skeletal trackers: their predicted vertex trajectories are converted to skeletal animations with a standardized skinning-weight-based post-processing protocol. The algorithm is given in Appendix C. We apply the same inverse-fit protocol to FoMo's mesh output and report it as *FoMo (mesh+inverse fit)*. Thus, all methods are evaluated in a common skeletal representation.

Metrics. For mesh evaluation, we follow Motion-80's evaluation protocol and normalize the ground truth and prediction independently using their first target frame, mapping each mesh to the canonical cube $[-1, 1]^3$. We estimate ICP on this frame and apply the same alignment to all target frames before computing metrics. For 4D mesh tracking, we report correspondence-aware vertex L_1 and L_2 errors, vertex velocity error (**VE**), and Chamfer distance (**CD**) over the complete mesh vertices. For skeletal tracking, we report mean per-joint position error (**MPJPE**) and mean per-joint velocity error (**MPJVE**) over skeleton joints with valid skinning weights. Detailed alignment protocols, metric definitions, and full-sequence inference settings are given in Appendix C.

4.2 Quantitative Evaluation

Tables 1 and 2 give the quantitative result layouts for mesh reconstruction and pose estimation.

4D mesh tracking. As demonstrated in Table 1, FoMo achieves the best performance across L_1, L_2, VE, and CD in both the full-sequence and 16-frames settings, under both reference protocols. Particularly, all the competing methods show a decline in performance in pose misalignment settings while FoMo maintains stable performance, demonstrating its robustness to reference mesh.



Figure 3: **Qualitative 4D mesh tracking with frame-0 references.** Each block shows several frames from a ground-truth sequence and the corresponding predictions from FoMo and competing methods.



Figure 4: **Qualitative 4D mesh tracking with rest-pose references.** The reference mesh is shown at the left of each example, followed by several frames from the ground-truth and predicted sequences.

Category-agnostic skeletal tracking. Table 2 shows the same pattern for category-agnostic skeletal tracking. FoMo obtains the best MPJPE and MPJVE with and without pose alignment, and the advantage remains after applying the standardized mesh-to-skeleton inverse fit to its mesh output.

4.3 Qualitative Evaluation

We qualitatively compare 4D mesh tracking under both reference protocols and category-agnostic skeletal tracking. Figures 3 and 4 show several mesh sequences with frame-0 and rest-pose references, respectively. Figure 8 compares direct skeletal predictions with skeletons obtained by inverse fitting mesh trajectories. The corresponding video demo is provided in the supplementary materials.

FoMo more closely follows the ground-truth articulation while preserving body shape, with fewer stretched or collapsed parts than the baselines. With rest-pose references, it better recovers arm gestures and leg poses despite the initial pose mismatch. The skeletal examples show closer limb configurations for both humanoid and non-humanoid characters, using either direct prediction or mesh-based inverse fitting.

4.4 Applications

4.4.1 In-the-Wild Video to 4D

We apply FoMo to in-the-wild monocular videos using generated reference meshes. For each mesh, we first use image generation model (e.g. GPT Image 2 (OpenAI, 2026)) to generate a rest-pose reference image, then reconstruct the corresponding reference mesh with 3D generator (e.g. Tripo P2.0 (Tripo AI, 2026)). We segment the target foreground in the driving video using segmentation model (e.g. SAM 3 (Carion et al., 2026)) and feed the segmented video together with the reference mesh into FoMo to reconstruct its motion. Figure 6 shows in-the-wild results. Our method generalizes well to in-the-wild video input and produces high-quality 4D motion reconstruction. Video demo and more visualizations with ai-generated synthetic video input are provided in the supplementary material and Appendix D.

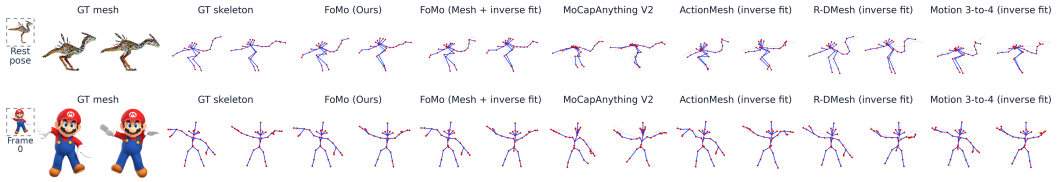

Figure 5: **Qualitative category-agnostic skeletal tracking.** The upper block uses rest-pose references and the lower block uses frame-0 references. FoMo is compared with direct skeletal prediction and with skeletons obtained by inverse fitting mesh trajectories.

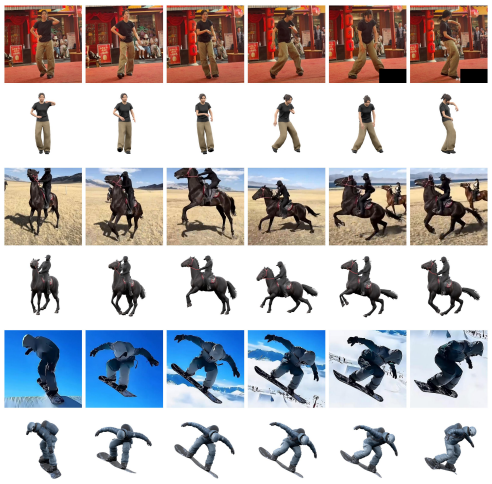

Figure 6: **In-the-wild reconstruction.** Dance, horseback riding, and snowboarding (top to bottom). Each pair shows video frames above reconstructed meshes; time progresses left to right.



Figure 7: **Motion transfer.** Each pair shows source motion above the animated target mesh. FoMo transfers articulated motion across different shapes and appearances.

4.4.2 Motion Transfer

FoMo also supports motion transfer by using a target character's mesh as the reference geometry and a video of a different source character as the motion condition. This allows the target mesh to follow the source motion while retaining its own shape, appearance, and vertex connectivity. Figure 7 shows transfer between dinosaur characters and between humanoid characters with different body proportions and clothing. The target sequences reproduce the source's leg movements, crouching, and upper-body gestures, illustrating the separation of reference geometry from video-conditioned motion.

4.5 Discussion

4.5.1 Native 4D vs. VAE

FoMo keeps a direct correspondence between generative tokens and mesh vertices: each vertex is represented by its own trajectory token. In a VAE representation, mesh motion is compressed into a smaller latent space and vertex-level geometry is recovered by the decoder, which can introduce distortion.

To quantify this effect, we train the VAE architecture of R-DMesh (Wu et al., 2026) on our training set with 16-frame clips, using 64, 256, or 512 latent tokens per mesh. All variants are trained for 200 epochs until reconstruction converges. We evaluate on Motion-80 (16 frames). This comparison heavily favors the VAE: it reconstructs the complete ground-truth mesh sequence given as input, whereas FoMo generates motion from a reference mesh and monocular video. EDR and FDR

Table 3: **Native 4D generation versus VAE reconstruction on Motion-80 (16 frames).** VAE variants directly reconstruct GT inputs; FoMo generates motion from video and a reference mesh. Lower is better.

Method	EDR (%)↓	FDR (%)↓	Mean seq. max.↓	Global max.↓
VAE (64 tokens)	0.857	2.46	0.082	0.353
VAE (256 tokens)	0.778	1.87	0.092	0.412
VAE (512 tokens)	0.738	1.57	0.091	0.399
FoMo (Ours)	**0.223**	**0.66**	**0.064**	**0.298**

Table 4: **Ablation on Motion-80 (16 frames).** Each variant removes one attention component. Lower is better for all metrics.

Variant	$L_1 \downarrow$	$L_2 \downarrow$	VE↓	CD↓
w/o reference spatial attention	0.290	0.203	0.035	0.105
w/o reference topology attention	0.091	0.065	0.026	0.068
w/o generation spatial attention	0.101	0.072	0.027	0.069
w/o generation temporal attention	0.115	0.083	0.063	0.086
FoMo (full)	**0.087**	**0.062**	**0.024**	**0.067**

measure the fractions of GT edge lengths and face areas whose predicted-to-GT ratios fall outside $[0.5, 2.0]$. We also report the mean per-sequence maximum and global maximum edge-length errors.

Table 3 shows that latent compression introduces local geometric distortion even when reconstructing complete GT inputs. Despite the harder generation setting, FoMo achieves the lowest distortion on all metrics, supporting direct vertex-trajectory modeling for preserving local geometry and correspondence.

4.5.2 Ablation Study

We ablate four components by removing one at a time from the full model: spatial self-attention and topology attention in the reference geometry block, and spatial self-attention and temporal attention in the generation geometry block. We train each model variant for 120k steps under identical settings. Table 4 presents the evaluation on Motion-80 (16 frames) under the w/ PA protocol, using the first-frame mesh as the reference. We use the same L_1, L_2, VE, and CD metrics as in the main 4D mesh tracking evaluation.

Removing any attention component worsens all four metrics (Table 4). Reference spatial attention has the largest effect on position accuracy as it contains mostly positional information of reference. Removing generation temporal attention raises VE, highlighting its role in temporal consistency. Generation spatial attention and reference topology attention provide further gains.

Regarding the dual-geometry design, we demonstrate the noise-to-target transport issue by removing reference geometry entirely, forcing the DiT to recover trajectories directly from pure noise while also resolving the noise-to-target pairing. The model fails substantially, with $L_1 = 1.016$ and Chamfer distance $= 0.227$.

5 Conclusion

We presented FoMo, a generative framework for tracking 4D motion naitvely in point-trajectory space. FoMo preserves correspondence throughout generation and provides a unified model for mesh and skeletal tracking. Experiments show that it outperforms prior methods across different reference settings, while generalizing to in-the-wild reconstruction and motion transfer. We hope FoMo offers a simple and effective step toward general 4D motion reconstruction.

6 Ethics Statement

This work uses publicly available and synthetic video and 3D assets. No human participants or animal experiments were involved, and no personally identifiable information was collected. We follow the applicable terms of use for the datasets and pretrained models used in our experiments.

7 Reproducibility Statement

The main paper and appendix provide the problem formulation, model architecture, training objective, inference strategy, evaluation protocols, metric definitions, and ablations. Experiments use the publicly available Motion-80 benchmark. These details are intended to support independent reproduction and verification of our results.

References

Sherwin Bahmani, Ivan Skorokhodov, Victor Rong, Gordon Wetzstein, Leonidas Guibas, Peter Wonka, Sergey Tulyakov, Jeong Joon Park, Andrea Tagliasacchi, and David B. Lindell. 4d-fy: Text-to-4d generation using hybrid score distillation sampling, 2024. URL https://arxiv.org/abs/2311.17984.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

Hongyuan Chen, Xingyu Chen, Youjia Zhang, Zexiang Xu, and Anpei Chen. Motion 3-to-4: 3d motion reconstruction for 4d synthesis, 2026. URL https://arxiv.org/abs/2601.14253.

Jianqi Chen, Biao Zhang, Xiangjun Tang, and Peter Wonka. V2m4: 4d mesh animation reconstruction from a single monocular video, 2025. URL https://arxiv.org/abs/2503.09631.

Ravi Garg, Anastasios Roussos, and Lourdes Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pp. 1272–1279, 2013.

Kehong Gong, Zhengyu Wen, Weixia He, Mingxi Xu, Qi Wang, Ning Zhang, Zhengyu Li, Dongze Lian, Wei Zhao, Xiaoyu He, and Mingyuan Zhang. Mocapanything: Unified 3d motion capture for arbitrary skeletons from monocular videos, 2026a. URL https://arxiv.org/abs/2512.10881.

Kehong Gong, Zhengyu Wen, Dao Thien Phong, Mingxi Xu, Weixia He, Qi Wang, Ning Zhang, Zhengyu Li, Guanli Hou, Dongze Lian, Xiaoyu He, Mingyuan Zhang, and Hanwang Zhang. Mocapanything v2: End-to-end motion capture for arbitrary skeletons, 2026b. URL https://arxiv.org/abs/2604.28130.

Yanqin Jiang, Li Zhang, Jin Gao, Weimin Hu, and Yao Yao. Consistent4d: Consistent 360deg $dynamicobjectgeneration from monocular video$, 2023. URL.

Yanqin Jiang, Chaohui Yu, Chenjie Cao, Fan Wang, Weiming Hu, and Jin Gao. Animate3d: Animating any 3d model with multi-view video diffusion, 2024. URL https://arxiv.org/abs/2407.11398.

Zeren Jiang, Chuanxia Zheng, Iro Laina, Diane Larlus, and Andrea Vedaldi. Mesh4d: 4d mesh reconstruction and tracking from monocular video, 2026. URL https://arxiv.org/abs/2601.05251.

Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7122–7131. IEEE, 2018.

Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5252–5262. IEEE, 2020.

Filippos Kokkinos and Iasonas Kokkinos. Learning monocular 3d reconstruction of articulated categories from motion. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1737–1746. IEEE, 2021.

Zhibing Li, Mengchen Zhang, Tong Wu, Jing Tan, Jiaqi Wang, and Dahua Lin. Ss4d: Native 4d generative model via structured spacetime latents. *ACM Transactions on Graphics*, 44(6):1–12, December 2025. ISSN 1557-7368. 10.1145/3763302. URL http://dx.doi.org/10.1145/3763302.

Hanwen Liang, Yuyang Yin, Dejia Xu, Hanxue Liang, Zhangyang Wang, Konstantinos N. Plataniotis, Yao Zhao, and Yunchao Wei. Diffusion4d: Fast spatial-temporal consistent 4d generation via video diffusion models, 2024. URL https://arxiv.org/abs/2405.16645.

Huan Ling, Seung Wook Kim, Antonio Torralba, Sanja Fidler, and Karsten Kreis. Align your gaussians: Text-to-4d with dynamic 3d gaussians and composed diffusion models, 2024. URL https://arxiv.org/abs/2312.13763.

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

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pp. 851–866. 2023.

OpenAI. GPT Image 2. https://developers.openai.com/api/docs/models/gpt-image-2, 2026. Model documentation. Accessed September 17, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. URL https://arxiv.org/abs/2212.09748.

Cheng-Feng Pu, Jia-Peng Zhang, Meng-Hao Guo, Yan-Pei Cao, and Shi-Min Hu. Topocap: Learning topology-agnostic motion priors for monocular video-to-animation. *arXiv preprint arXiv:2606.12153*, 2026.

Jiawei Ren, Kevin Xie, Ashkan Mirzaei, Hanxue Liang, Xiaohui Zeng, Karsten Kreis, Ziwei Liu, Antonio Torralba, Sanja Fidler, Seung Wook Kim, and Huan Ling. L4gm: Large 4d gaussian reconstruction model, 2024. URL https://arxiv.org/abs/2406.10324.

Remy Sabathier, David Novotny, Niloy J. Mitra, and Tom Monnier. Actionmesh: Animated 3d mesh generation with temporal 3d diffusion, 2026. URL https://arxiv.org/abs/2601.16148.

Yahao Shi, Yang Liu, Yanmin Wu, Xing Liu, Chen Zhao, Jie Luo, and Bin Zhou. Drive any mesh: 4d latent diffusion for mesh deformation from video. *arXiv preprint arXiv:2506.07489*, 2025.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, 2025. URL https://arxiv.org/abs/2508.10104.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

Edith Tretschk, Navami Kairanda, Mallikarjun BR, Rishabh Dabral, Adam Kortylewski, Bernhard Egger, Marc Habermann, Pascal Fua, Christian Theobalt, and Vladislav Golyanik. State of the art in dense monocular non-rigid 3d reconstruction. In *Computer Graphics Forum*, volume 42, pp. 485–520. Wiley Online Library, 2023.

Tripo AI. Tripo. https://www.tripo3d.ai/, 2026. Online 3D generation platform.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time, 2019. URL https://arxiv.org/abs/1903.07593.

Rundi Wu, Ruiqi Gao, Ben Poole, Alex Trevithick, Changxi Zheng, Jonathan T. Barron, and Aleksander Holynski. Cat4d: Create anything in 4d with multi-view video diffusion models, 2024. URL https://arxiv.org/abs/2411.18613.

Zijie Wu, Lixin Xu, Puhua Jiang, Sicong Liu, Chunchao Guo, and Xiang Bai. R-dmesh: Video-guided 3d animation via rectified dynamic mesh flow. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–12, 2026.

Yiming Xie, Chun-Han Yao, Vikram Voleti, Huaizu Jiang, and Varun Jampani. Sv4d: Dynamic 3d content generation with multi-frame and multi-view consistency, 2025. URL https://arxiv.org/abs/2407.17470.

Chun-Han Yao, Yiming Xie, Vikram Voleti, Huaizu Jiang, and Varun Jampani. Sv4d 2.0: Enhancing spatio-temporal consistency in multi-view video diffusion for high-quality 4d generation, 2025. URL https://arxiv.org/abs/2503.16396.

Jiraphon Yenphraphai, Ashkan Mirzaei, Jianqi Chen, Jiaxu Zou, Sergey Tulyakov, Raymond A. Yeh, Peter Wonka, and Chaoyang Wang. Shapegen4d: Towards high quality 4d shape generation from videos, 2025. URL https://arxiv.org/abs/2510.06208.

Bowen Zhang, Sicheng Xu, Chuxin Wang, Jiaolong Yang, Feng Zhao, Dong Chen, and Baining Guo. Gaussian variation field diffusion for high-fidelity video-to-4d synthesis, 2025. URL https://arxiv.org/abs/2507.23785.

Haiyu Zhang, Xinyuan Chen, Yaohui Wang, Xihui Liu, Yunhong Wang, and Yu Qiao. 4diffusion: Multi-view video diffusion model for 4d generation, 2024. URL https://arxiv.org/abs/2405.20674.

Yuyang Zhao, Zhiwen Yan, Enze Xie, Lanqing Hong, Zhenguo Li, and Gim Hee Lee. Animate124: Animating one image to 4d dynamic scene, 2024. URL https://arxiv.org/abs/2311.14603.

APPENDIX

A TRAINING DETAILS

For each mesh, we sample up to 1,024 vertices for efficiency and keep the same indices in the reference and all target frames. We sample 12 target frames with a random start and a random temporal stride intervals of 1, 2, or 4 frames, and choose the reference frame independently from the full sequence. For the monocular video, we render each clip at a resolution of 256×256 from 16 viewpoints with uniformly sampled azimuth angles in $[0°, 90°]$. Skeleton inputs are dropped with a dropout rate of 0.3. Samples are normalized to $[-0.5, 0.5]^3$.

B ARCHITECTURE DETAILS

B.1 MODEL CONFIGURATION

FoMo processes $T(N+K)$ continuous-coordinate tokens for T target frames, N mesh vertices, and K optional joints ($K = 0$ for mesh-only inputs). During inference, we use standard 50-step Euler ODE solver. Tokens retain their correspondence indices and are packed to support variable mesh sizes. Table 5 lists the model dimensions.

Table 5: Architecture of the evaluated FoMo checkpoint. The parameter count excludes the frozen image encoder.

Component	Configuration
Interleaved stages	16 (16 reference + 16 generation blocks)
Hidden / feed-forward width	512 / 1,536
Main attention	8 query heads, 8 key/value heads, head width 64
Feed-forward activation	SwiGLU with SiLU gate
Topology attention	4 query heads, 2 key/value heads, head width 64
Mesh / skeleton neighborhoods	5 hops / 1 hop; at most 64 neighbors
Spatial RoPE	Continuous 3D; coordinate scale 128; base 10,000
Temporal RoPE	1D; base 10,000
Image encoder	Frozen DINOv3 ViT-H/16+, feature width 1,280
Trainable parameters	186,640,643

B.2 NETWORK MODULES

Embeddings and conditioning. Noisy XYZ coordinates are linearly projected to width 512. Reference RGB–normal attributes use a $6 \to 512 \to 512$ SiLU MLP. Flow time uses a 256-dimensional sinusoidal embedding and an MLP to produce adaptive LayerNorm shifts, scales, and residual gates. Frozen DINOv3 features are projected separately for the two streams by $\mathrm{LN}(1280) \to 1280 \to \mathrm{GELU} \to 512 \to \mathrm{LN}(512)$. For joint tokens, a skeleton-type embedding is added to distinguish them from mesh vertex tokens.

Reference and generation blocks. Each stage runs a reference block followed by a generation block, with separate parameters. Their operation order is:

Reference: spatial attention $\to$ topology attention $\to$ reference-image cross-attention $\to$ feed-forward layer. Spatial RoPE uses clean reference coordinates. Mesh topology keys/values encode RGB–normal attributes with a $6 \to 256 \to 256$ SiLU MLP and RMS normalization; skeleton topology uses latent joint features. The two topology graphs are disjoint.

Generation: spatial attention $\to$ temporal attention $\to$ target-frame image cross-attention $\to$ feed-forward layer. Spatial RoPE uses noisy target coordinates. Temporal attention alternates between point-wise and global layouts (eight blocks each), excluding the reference frame.

Table 6: Sequence length and native mesh resolution in the Motion-80 full-sequence evaluation set. Percentiles are computed across evaluation samples.

	Mean	Median	P90	P95	Max
Target frames per sample	88.76	43	243	395	628
Vertices per sample	4,208	3,129	7,899	8,991	13,127

C Evaluation Details

Evaluation Set Distribution. Table 6 summarizes the sequence lengths and native mesh resolutions of the Motion-80 full-sequence evaluation set. The median sample contains 43 target frames and 3,128.5 vertices, while the 90th percentiles reach 242.7 frames and 7,898.7 vertices. The longest sequence has 628 frames, and the largest mesh has 13,127 vertices. These distributions provide context for evaluating FoMo beyond its training configuration. During full-sequence evaluation, FoMo processes each target sequence jointly and predicts trajectories for all native mesh vertices, without auto-regressive rollout or vertex subsampling.

Evaluation setup. For Motion-80 (full), we follow the official baseline implementations and use autoregressive (AR) windows of 256 frames for Motion 3-to-4, 64 frames for R-DMesh, and 16 frames for ActionMesh. Sequences longer than these windows require successive AR windows. FoMo processes the full target sequence jointly without AR rollout. To reduce memory and computation, we choose its global temporal-attention setting according to the target length: we use full temporal attention for sequences shorter than 96 frames, local temporal attention with a context of 32 frames for sequences from 96 to 256 frames, and local temporal attention with a context of 12 frames for sequences longer than 256 frames. Only the global temporal blocks use this frame-level sliding-window pattern; point-wise temporal blocks retain full-trajectory attention. The context parameter controls attention locality within a joint full-sequence sampling run, rather than an AR generation window. On Motion-80 (16 frames), all methods generate the target clip without AR rollout. Following the Motion-80 protocol, front-view rendered videos are used as input. ActionMesh is trained with camera azimuths in $[40^\circ, 85^\circ]$ and fails to produce usable outputs when directly evaluated under this frontal-view convention. For a fair comparison, we rotate its input reference mesh by 60° in azimuth to align the relative orientation between the reference mesh and the driving video with its training view range, while keeping the input video unchanged.

Tracking alignment. For mesh evaluation, we follow the Motion-80 evaluation protocol. The ground truth and prediction are independently normalized using their first target frame, mapping each mesh to the canonical cube $[-1, 1]^3$. The resulting normalization parameters are fixed and applied to all target frames. We estimate point-to-point ICP on the first normalized target frame and apply the same alignment to the predicted mesh in every target frame before computing the metrics. For skeleton evaluation, we evaluate joints with positive skinning weights. GT and predicted skeletons are independently normalized using the first target frame's bounding box. We then fit a similarity transform (rotation, translation, and uniform scale) to corresponding joints in that frame using Umeyama alignment, and apply it to the entire predicted sequence.

Tracking metrics. All metrics are lower-is-better and averaged over target frames and evaluated points. L_1 is the mean corresponding-vertex Manhattan distance (sum over XYZ), and L_2 is the mean Euclidean distance. CD sums the two directed mean nearest-neighbor Euclidean distances, without squaring or dividing by two. Let X_t and $\hat{X}_t$ denote aligned GT and predicted meshes, with N corresponding vertices and T target frames. Then

$$L_p = \frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} \|\hat{\mathbf{x}}_{t,i} - \mathbf{x}_{t,i}\|_p, \qquad p \in \{1, 2\},$$

$$\mathrm{CD} = \frac{1}{T} \sum_{t=1}^{T} \left[\frac{1}{|\hat{X}_t|} \sum_{\hat{\mathbf{x}} \in \hat{X}_t} \min_{\mathbf{x} \in X_t} \|\hat{\mathbf{x}} - \mathbf{x}\|_2 + \frac{1}{|X_t|} \sum_{\mathbf{x} \in X_t} \min_{\hat{\mathbf{x}} \in \hat{X}_t} \|\mathbf{x} - \hat{\mathbf{x}}\|_2 \right].$$

VE measures the Euclidean error in corresponding-vertex displacements between adjacent target frames:

$$\mathrm{VE} = \frac{1}{(T-1)N}\sum_{t=2}^{T}\sum_{i=1}^{N}\left\|(\hat{\mathbf{x}}_{t,i}-\hat{\mathbf{x}}_{t-1,i})-(\mathbf{x}_{t,i}-\mathbf{x}_{t-1,i})\right\|_2.$$

MPJPE and MPJVE are the corresponding Euclidean position and displacement errors for J evaluated joints. With $\Delta\mathbf{q}_{t,j} = \mathbf{q}_{t,j} - \mathbf{q}_{t-1,j}$,

$$\mathrm{MPJPE} = \frac{1}{TJ}\sum_{t=1}^{T}\sum_{j=1}^{J}\|\hat{\mathbf{q}}_{t,j}-\mathbf{q}_{t,j}\|_2,$$

$$\mathrm{MPJVE} = \frac{1}{(T-1)J}\sum_{t=2}^{T}\sum_{j=1}^{J}\|\Delta\hat{\mathbf{q}}_{t,j}-\Delta\mathbf{q}_{t,j}\|_2.$$

Since MoCapAnything V2 does not model global skeleton translation, we additionally translate its prediction per frame so that its skeleton centroid matches the ground-truth centroid before computing the skeleton metrics. This yields a unified evaluation of motion quality and structural integrity across all methods.

Mesh-to-skeleton inverse fitting. We recover skeleton motion from generated meshes using the known reference rig and skinning weights. For each frame, we fit joint rotations and translations so that their linear-blend-skinned reference mesh matches the prediction. For reference vertices $\mathbf{v}_i$, weights w_{ij}, and a generated target mesh $\mathbf{y}_i$, the per-frame objective is

$$\min_{\{R_j\in SO(3),\,\mathbf{d}_j\}}\sum_i\left\|\sum_j w_{ij}(R_j\mathbf{v}_i+\mathbf{d}_j)-\mathbf{y}_i\right\|_2^2.$$

Eight parent-first sweeps update one joint at a time using weighted Kabsch fitting, with squared skinning weights, while holding other joint transforms fixed. Applying the fitted transforms to reference joints gives $\hat{\mathbf{q}}_{t,j} = R_{t,j}\mathbf{q}_j^r + \mathbf{d}_{t,j}$.

Geometry-preservation metrics. The VAE comparison uses all native vertices and original faces over 1,276 valid target frames from 80 sequences; reference and repeated padding frames are excluded. These metrics use stored coordinates without additional scaling or alignment. EDR is the fraction of total GT edge length belonging to edges whose predicted-to-GT length ratio falls outside $[0.5, 2.0]$. FDR analogously measures the fraction of GT face area with an out-of-range area ratio. Both rates are computed per frame, averaged within each sequence. They weight by GT length or area. For GT edge lengths ℓ_e and face areas A_f (hats denote predictions), let $I(r) = \mathbf{1}\{r < 0.5 \text{ or } r > 2.0\}$. The per-frame definitions are

$$\mathrm{EDR}_{s,t} = \frac{\sum_{e\in E^*_{s,t}}\ell_e I(\hat{\ell}_e/\ell_e)}{\sum_{e\in E^*_{s,t}}\ell_e}, \qquad \mathrm{FDR}_{s,t} = \frac{\sum_{f\in F^*_{s,t}}A_f I(\hat{A}_f/A_f)}{\sum_{f\in F^*_{s,t}}A_f}.$$

Here $E^*_{s,t}$ and $F^*_{s,t}$ exclude degenerate GT elements as specified below. For either rate R, the reported percentage is $100S^{-1}\sum_{s=1}^{S}|\mathcal{T}_s|^{-1}\sum_{t\in\mathcal{T}_s}R_{s,t}$, where $\mathcal{T}_s$ contains valid target frames.

For each sequence, we also take the largest absolute edge-length error across all valid frames and unique undirected non-self edges. *Mean seq. max.* averages these sequence maxima, while *Global max.* takes their maximum. For the unique undirected non-self edge set E_s,

$$M_s = \max_{t\in\mathcal{T}_s,\,e\in E_s}|\hat{\ell}_{s,t,e}-\ell_{s,t,e}|, \qquad \text{Mean seq. max.} = \frac{1}{S}\sum_s M_s, \qquad \text{Global max.} = \max_s M_s.$$

The VAE reconstructs GT geometry, whereas FoMo generates from a reference mesh and video, as specified in Section 4.5.1.

D MORE IN-THE-WILD VIDEO-TO-4D VISUALIZATION

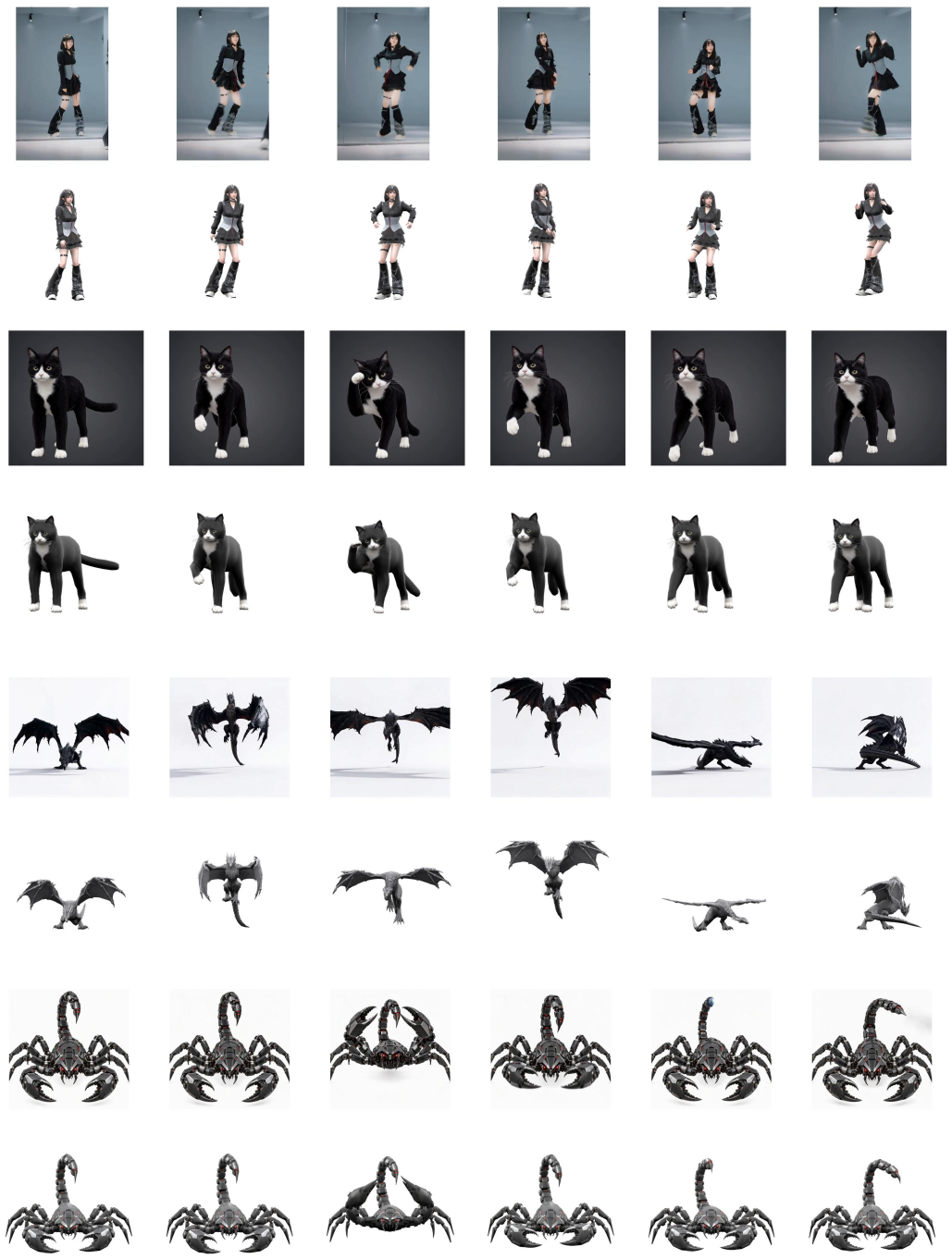

Figure 8: More in-the-wild video-to-4D results. The first example uses a real-world video as input, while the other three use AI-generated synthetic videos as input.

E AI USE STATEMENT

Large language models (LLMs) were used to edit and polish portions of the manuscript for clarity and readability. LLMs also served as supplementary tools in implementing certain algorithms of the proposed framework. The authors reviewed and revised all AI-assisted text, and verified the resulting implementations and experimental outputs. The authors take full responsibility for the final manuscript, claims, methods, code, and results, including content produced with the assistance of generative AI tools.