TRoPE: Topology-aware Relative Positional Encoding for 4D Generation

Anonymous authors
Paper under double-blind review

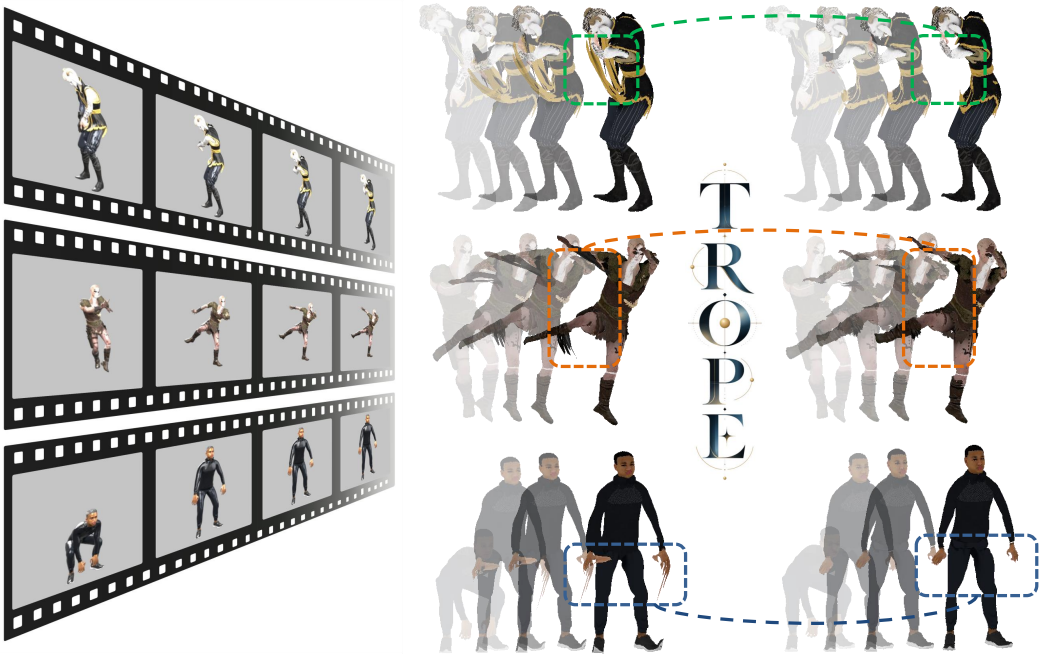

Figure 1: **TRoPE reduces sticking artifacts in 4D generation.** Given a reference mesh and a monocular video (left), TRoPE incorporates mesh structure into motion generation. Compared with the baseline (middle), our method (right) reduces sticking artifacts between nearby body parts, producing more faithful motion.

Abstract

Video-to-4D mesh generation has made substantial progress in recovering dynamic 3D geometry and motion from monocular videos. However, commonly used feed-forward approaches remain prone to **sticking artifacts**, where geodesically distant vertices move together incorrectly when they become spatially close. We attribute this failure to **cross-part motion leakage**: spatially nearby yet structurally distant vertices may retrieve similar motion information when motion decoding fails to *distinguish spatial proximity from mesh connectivity*. To address this issue, we introduce **TRoPE**, a topology-aware relative positional encoding that incorporates mesh structure into motion generation. TRoPE derives spectral features from the graph Laplacian and constructs pairwise topological relations from distances between multi-scale diffusion features, yielding a stable topology representation across meshes. We further introduce a vertex-wise tail-risk objective that emphasizes persistently high-error vertices to explicitly suppress sticking artifacts. To better characterize and quantify these failures, we introduce **MeshSticking-4D**, a dedicated benchmark for this failure mode. Experiments on Motion-80 and MeshSticking-4D show that TRoPE consistently suppresses sticking artifacts while maintaining comparable overall generation quality. Compared with the baseline, it reduces the worst-vertex error by 39.4% and lowers the sepa-

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.

ration failure rate from 49.35% to 34.90%. Additional evaluations across different experiments further demonstrate the general applicability of TRoPE. The code and benchmark will be made publicly available.

1 INTRODUCTION

Video-to-4D mesh generation recovers dynamic 3D geometry and motion from monocular videos. Traditional pipelines bind vertices to a skeleton before driving the mesh; automatic-rigging methods such as Puppeteer (Song et al., 2025) reduce this setup. Feed-forward methods, including Motion 3-to-4, R-DMesh, ActionMesh, and Mesh4D (Chen et al., 2026; Wu et al., 2026a; Sabathier et al., 2026; Jiang et al., 2026), directly infer vertex trajectories or deformations, avoiding explicit rigging and enabling efficient animation of diverse meshes. This paradigm benefits applications such as game development, simulation, robotics, and world modeling.

Despite this progress, feed-forward models exhibit **sticking artifacts**: as shown in Figure 1, geodesically distant vertices may move together incorrectly when they become spatially close. Examples include hands sticking to the torso and garments remaining attached to nearby limbs. Such failures become more pronounced in long sequences and complex motions, limiting the practical use of these methods.

This behavior reflects **cross-part motion leakage** in the decoder. Feed-forward models compress video into motion tokens and let each dense vertex query them through pairwise attention. These vertices are typically parameterized by their spatial or geometric features. As a result, vertices from geodesically distant but spatially nearby regions may attend to similar motion latents. This causes motion to leak across unrelated parts and ultimately produces **sticking artifacts**. (Chen et al., 2026; Wu et al., 2026b). Because Euclidean proximity changes with articulation while reference-mesh connectivity remains fixed, motion decoding needs a persistent, connectivity-aware relative signal.

To this end, we introduce **TRoPE**, a topology-aware relative positional encoding that incorporates mesh connectivity into motion decoding. We first derive spectral features from the graph Laplacian to characterize structural relationships on the mesh, enabling motion retrieval to *distinguish mesh connectivity from spatial proximity*. However, directly using spectral features across meshes is challenging due to the sign and eigenspace ambiguities of Laplacian eigenvectors. (Lim et al., 2023) To overcome this limitation, we construct pairwise topological relations using distances between multi-scale diffusion features, resulting in a stable representation of mesh topology. Furthermore, we introduce a vertex-wise tail-risk measure that emphasizes vertices with persistently high errors and formulate it as a training objective to explicitly suppress sticking artifacts.

To evaluate our method, we introduce **MeshSticking-4D**, a benchmark centered on sticking artifacts. Experiments on Motion-80 (Chen et al., 2026) and MeshSticking-4D show that TRoPE effectively reduces sticking artifacts while maintaining comparable overall generation quality, reducing the worst-vertex error by 39.4% on MeshSticking-4D. Evaluations across different experiments further demonstrate its general applicability.

Our contributions are summarized as follows:

- We identify cross-part motion leakage as a key cause of sticking artifacts in feed-forward Video-to-4D generation, and highlight the mismatch between spatial proximity and mesh connectivity.
- We introduce TRoPE, a topology-aware relative positional encoding that incorporates mesh connectivity into motion generation. TRoPE constructs stable relative topology relations from pairwise distances between multi-scale diffusion features.
- We introduce MeshSticking-4D, a benchmark specifically designed to evaluate sticking artifacts. Experiments on Motion-80 and MeshSticking-4D show that TRoPE consistently reduces sticking artifacts while preserving overall generation quality.

2 RELATED WORK

Absolute and Relative Positional Encodings. Transformer positional information has progressed from absolute token embeddings to pairwise relative representations (Shaw et al., 2018). Relative

positions can be injected into attention via additive biases or SO(2)-based rotation (Vaswani et al., 2017; Liutkus et al., 2021; Raffel et al., 2020; Su et al., 2024; Press et al., 2022). In geometric vision, camera-conditioning methods such as CaPE, GTA, Repast, and PRoPE replace global camera coordinates with relative relationships between interacting views (Kong et al., 2024; Safin et al., 2023; Miyato et al., 2024; Li et al., 2025). Across multiple domains, positional encodings are shifting from absolute to relative formulations. In this work, we compare absolute and relative encodings for mesh representations to incorporate mesh connectivity into motion generation.

Mesh Representations. Mesh connectivity provides structural relationships that are stable under deformation. Classical spectral approaches to shape analysis, including the heat kernel signature, wave kernel signature, and functional maps, capture intrinsic relationships between surface locations (Bronstein & Kokkinos, 2010; Sun et al., 2009; Aubry et al., 2011; Attaiki & Ovsjanikov, 2023; Ovsjanikov et al., 2012). Modern mesh and graph networks use adjacency, surface diffusion, graph shortest-path distances, or spectral features to model local and global structure, as demonstrated by DiffusionNet, MeshGraphNets, Mesh Graphormer, Graphormer, GraphiT, SAN, and GraphGPS (Sharp et al., 2022; Pfaff et al., 2021; Lin et al., 2021; Ying et al., 2021; Mialon et al., 2021; Kreuzer et al., 2021; Rampásek et al., 2022). SignNet and BasisNet further address sign and eigenspace ambiguities in spectral representations (Lim et al., 2023). These studies provide powerful tools for representing mesh structure. In this work, we study a critical design decision: how to incorporate mesh connectivity into motion generation.

Video-to-4D Generation. Video-to-4D generation has evolved from optimizing implicit or Gaussian representations with image and video priors to feed-forward models that explicitly reconstruct dynamic geometry. Methods such as 4D-fy, Consistent4D, SV4D, STAG4D, and CAT4D exploit multi-view or video diffusion priors to synthesize temporally varying scenes, while L4GM and GVFD model dynamic objects with Gaussian or latent representations (Bahmani et al., 2024; Jiang et al., 2024; Xie et al., 2025; Zeng et al., 2024; Wu et al., 2025; Ren et al., 2024; Zhang et al., 2025). A parallel line focuses on explicit mesh representations. V2M4 and ShapeGen4D reconstruct meshes frame by frame and align them over time, whereas Motion 3-to-4, R-DMesh, Mesh4D, and ActionMesh learn explicit vertex trajectories or mesh deformations from video (Chen et al., 2025a; Yenphraphai et al., 2025; Chen et al., 2026; Wu et al., 2026a; Jiang et al., 2026; Sabathier et al., 2026). While these methods opened a path to 4D mesh generation, they suffer from sticking artifacts. Our work takes a step forward by designing a topology-aware relative positional encoding for 4D generation.

3 METHOD

We introduce **TRoPE**, a topology-aware relative positional encoding for 4D generation. The host generator encodes the video into compact motion tokens, while TRoPE changes how mesh vertices retrieve those tokens. A vertex-wise tail-risk objective emphasizes the persistent high-error vertices that form sticking artifacts.

3.1 VIDEO-TO-4D MOTION DECODING

Given a monocular video $\mathbf{I}_{1:T}$ and a reference mesh $\mathcal{M}^{\text{ref}} = (\boldsymbol{V}^{\text{ref}}, \mathcal{F})$, video-to-4D generation predicts a dynamic mesh sequence

$$\widehat{\mathcal{M}}_{1:T} = \left\{(\widehat{\boldsymbol{V}}_t, \mathcal{F})\right\}_{t=1}^{T}, \qquad \widehat{\boldsymbol{V}}_t \in \mathbb{R}^{N\times 3}. \tag{1}$$

The vertex indexing and face connectivity of the reference mesh are fixed over time, whereas Euclidean distances can change substantially during articulation.

The host generator encodes the video and reference geometry into a compact set of per-frame motion tokens $\boldsymbol{Z}_t = \{\boldsymbol{z}_{t,j}\}_{j=1}^{S}$, with $S \ll N$. Each mesh vertex forms a query and retrieves motion through content-based attention. For head h,

$$\ell^h_{itj} = \frac{(\boldsymbol{q}^h_i)^\top \boldsymbol{k}^h_{t,j}}{\sqrt{d_h}}, \qquad p^h_{itj} = \mathrm{softmax}_j(\ell^h_{itj}), \qquad \boldsymbol{o}^h_{i,t} = \sum_{j=1}^{S} p^h_{itj} \boldsymbol{v}^h_{t,j}. \tag{2}$$

Thus, each vertex chooses motion information from the token set. When two geodesically distant parts become close in the Euclidean space, their query features can become similar and retrieve

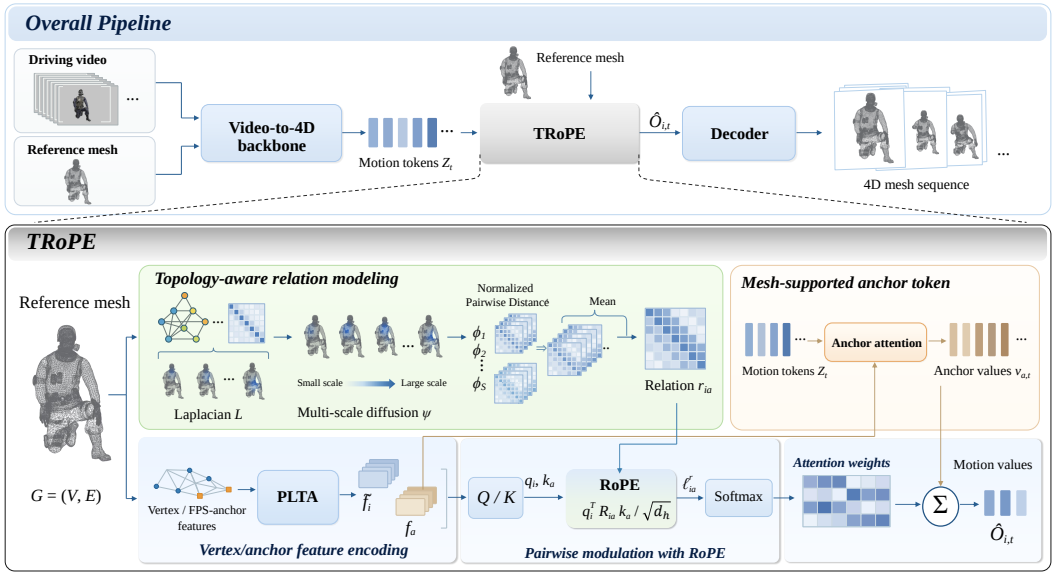

Figure 2: **Overview of TRoPE for Topology-Aware Video-to-4D Mesh Generation.** Given a driving video and a reference mesh, a video-to-4D backbone extracts motion tokens, which TRoPE associates with mesh-supported anchors before a decoder generates the 4D mesh sequence. The lower panel details TRoPE: PLTA encodes local vertex and anchor features, while the reference connectivity graph is transformed through the graph Laplacian and multi-scale diffusion into pairwise topology-relative relations. These relations modulate query–key compatibility through rotary attention, producing motion features that preserve mesh structure and suppress cross-part motion leakage and sticking artifacts.

similar tokens. This *cross-part motion leakage* couples trajectories that should remain separate and appears as *sticking artifacts*. An absolute coordinate or spectral code attached to each query does not specify the relationship between that query and a motion token; raw spectral channels also depend on the arbitrary orientation of the returned eigenbasis (Lim et al., 2023). We therefore give the motion source a mesh support and encode the query–source relation directly.

3.2 Mesh-supported Topological Relations

To distinguish mesh connectivity from spatial proximity during motion decoding, we construct relative positions on the reference graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ induced by the mesh faces.

Mesh-supported motion sources. To give motion features explicit mesh locations, we sample surface anchors $\mathcal{A}$ by farthest-point sampling. To incorporate local connectivity into their interaction with vertex queries, we adapt the topology-aware aggregation of AnimateAnyMesh++ (Wu et al., 2026b) into a local sparse attention block, denoted PLTA:

$$\widetilde{\boldsymbol{f}}_i = \mathrm{PLTA}\big(\boldsymbol{f}_i, \{\boldsymbol{f}_j : j \in \mathcal{N}_{\mathcal{G}}(i)\}\big). \tag{3}$$

Here, $\boldsymbol{f}_i$ is the initial feature and $\mathcal{N}_{\mathcal{G}}(i)$ its mesh neighborhood. The enriched anchor features query the host motion tokens:

$$\boldsymbol{m}_{a,t} = \mathrm{Readout}(\widetilde{\boldsymbol{f}}_a, \boldsymbol{Z}_t), \qquad a \in \mathcal{A}. \tag{4}$$

This associates each retrieved motion feature with a reference-surface location, allowing vertex–anchor connectivity to inform subsequent attention.

Multi-scale connectivity. To describe connectivity beyond local neighborhoods, we derive diffusion features from the graph Laplacian (Sharp et al., 2022). We process each connected component independently to capture its internal structure. For component c,

$$\boldsymbol{L}_c = \boldsymbol{I} - \boldsymbol{D}_c^{-1/2} \boldsymbol{A}_c \boldsymbol{D}_c^{-1/2} = \boldsymbol{\Phi}_c \boldsymbol{\Lambda}_c \boldsymbol{\Phi}_c^{\top}. \tag{5}$$

where $\boldsymbol{A}_c$ and $\boldsymbol{D}_c$ are its adjacency and degree matrices. We retain non-trivial modes indexed by $\mathcal{I}_c$. Let $\bar{\lambda}_{c,m}$ denote eigenvalues normalized by the median positive retained value, and $\widetilde{\boldsymbol{\phi}}_{c,m}$ the component-standardized, clipped eigenvectors. To capture different structural ranges, we construct features at diffusion scale τ:

$$\begin{aligned} \boldsymbol{\psi}_{c,\tau}(i) &= \left(\sqrt{w_{\tau,c,m}}\,\widetilde{\boldsymbol{\phi}}_{c,m}(i)\right)_{m\in\mathcal{I}_c}, \\ w_{\tau,c,m} &= \frac{\exp\left(-\tau\bar{\lambda}_{c,m}\right)}{\left\|\left(\exp\left(-\tau\bar{\lambda}_{c,r}\right)\right)_{r\in\mathcal{I}_c}\right\|_2}. \end{aligned} \tag{6}$$

To express the relationship between a query and a motion source, we compare their diffusion features through normalized distances across scales $\mathcal{T}$. For valid endpoints in the same component,

$$r_{ia} = \frac{1}{|\mathcal{T}|}\sum_{\tau\in\mathcal{T}} \frac{\left\|\boldsymbol{\psi}_{c,\tau}(i) - \boldsymbol{\psi}_{c,\tau}(a)\right\|_2}{\max\left\{\left\|\boldsymbol{\psi}_{c,\tau}(i)\right\|_2 + \left\|\boldsymbol{\psi}_{c,\tau}(a)\right\|_2, \epsilon\right\}}. \tag{7}$$

Distances and norms are unchanged by a common orthogonal transformation of the diffusion factors, avoiding the need to interpret their individual coordinate directions. Surface samples inherit spectral features by barycentric interpolation. We assign $r_{ia} = 0$ to cross-component or invalid pairs, preserving content-based attention for those pairs. To keep the structural information independent of articulation, we compute the relation from reference connectivity and reuse it across frames.

3.3 Topology-aware Relative RoPE

To make motion retrieval depend on both content and connectivity, we use r_{ia} to modulate query–key compatibility through a rotary phase (Su et al., 2024). Rotation changes feature alignment while preserving feature norms. For fixed rotary frequencies $\{\omega_f\}_{f=1}^{F_r}$ with $2F_r \le d_h$,

$$\begin{aligned} \theta_{ia,f} &= 2\pi\omega_f r_{ia}, \qquad \boldsymbol{R}(\theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}, \\ \boldsymbol{R}_{ia} &= \operatorname{diag}\left(\boldsymbol{R}(\theta_{ia,1}), \ldots, \boldsymbol{R}(\theta_{ia,F_r}), \boldsymbol{I}_{d_h-2F_r}\right). \end{aligned} \tag{8}$$

Queries $\boldsymbol{q}_i^h$ and keys $\boldsymbol{k}_a^h$ are projected from the enriched reference features $\widetilde{\boldsymbol{f}}_i$ and $\widetilde{\boldsymbol{f}}_a$. We compute their compatibility as

$$\widetilde{\ell}_{ia}^h = \frac{(\boldsymbol{q}_i^h)^\top \boldsymbol{R}_{ia}\boldsymbol{k}_a^h}{\sqrt{d_h}}. \tag{9}$$

We use this form to let the same structural relation affect different feature pairs differently. For one two-dimensional channel pair,

$$\boldsymbol{q}^\top \boldsymbol{R}(\theta)\boldsymbol{k} = (q_1k_1 + q_2k_2)\cos\theta + (q_1k_2 - q_2k_1)\sin\theta. \tag{10}$$

Connectivity sets the cosine and sine coefficients, while the learned features determine the aligned and cross-channel matches. To translate this joint comparison into retrieved motion, we normalize over anchors:

$$\widetilde{p}_{ia}^h = \operatorname{softmax}_a(\widetilde{\ell}_{ia}^h), \qquad \widetilde{\boldsymbol{o}}_{i,t}^h = \sum_{a\in\mathcal{A}} \widetilde{p}_{ia}^h \boldsymbol{v}_{a,t}^h. \tag{11}$$

Here, $\boldsymbol{v}_{a,t}^h$ is projected from $\boldsymbol{m}_{a,t}$. To retain a consistent structural comparison throughout the sequence, we share queries, keys, and r_{ia} across frames while updating motion values with the video. The aggregated features pass to the existing coordinate prediction head. Additional derivations of TRoPE are provided in Appendix A.1.

3.4 Sticking-Aware Tail-Risk Training

To emphasize persistent local errors that global reconstruction averages can underweight, we supplement coordinate MSE with vertex-wise CVaR (Rockafellar & Uryasev, 2000). We rank vertices by temporal mean error to focus on sustained trajectory errors. Let $e_i = T^{-1}\sum_t \|\widehat{\boldsymbol{v}}_{t,i} - \boldsymbol{v}_{t,i}\|_2$ and let $\mathcal{H}_\rho$ contain the $K_\rho = \lceil \rho N_s \rceil$ largest temporal errors among the N_s supervised surface locations. The training objective is

$$\mathcal{L} = \mathcal{L}_{\text{coord}} + \lambda_{\text{tail}} w(s) \frac{1}{K_\rho}\sum_{i\in\mathcal{H}_\rho} e_i, \qquad w(s) = \min\left(1, \frac{s}{200}\right), \quad \rho = \lambda_{\text{tail}} = 0.01. \tag{12}$$

The hardest vertices are selected independently per sample.

Using Motion 3-to-4 (Chen et al., 2026) as our host generator, we fine-tune TRoPE on MeshSticking-4D for 42,600 updates with batch size eight and a frozen image encoder. We use 12 frames at stride two, 4,096 surface queries, 1,024 FPS anchors, $K = 16$ modes per component, $\mathcal{T} = \{0.25, 1, 4, 16\}$, and eight fixed logarithmically spaced frequencies in $[0.5, 16]$. AdamW uses a base learning rate of 3×10^{-5} for the pretrained backbone, while the newly introduced modules use a learning rate of 9×10^{-4}, with cosine decay after 500 warm-up updates (Loshchilov & Hutter, 2019).

4 EXPERIMENTS

We first evaluate whether TRoPE reduces sticking artifacts while maintaining comparable overall generation quality, then examine the positional relation and structural design behind this behavior.

4.1 EXPERIMENTAL SETUP

Datasets. We first construct **MeshSticking-4D**, a benchmark designed to evaluate sticking artifacts in video-to-4D mesh generation. We combine Mixamo character meshes with source animations and select clips with pronounced articulation and close approaches between structurally distant body parts (Guo et al., 2025). Each sample contains a reference mesh, a 32-frame mesh sequence, and monocular videos rendered from 8 synchronized viewpoints. The benchmark consists of 10080 training samples and 256 test samples. For evaluation on overall generation quality, we additionally use the public Motion-80 benchmark of Motion 3-to-4 (Chen et al., 2026) under a reference-mesh-conditioned 16-frame protocol. Additional details are provided in Appendix A.2.

Metrics. We evaluate motion fidelity using three correspondence-based metrics: Euclidean Distance (EucD), worst-vertex error (WVE), and framewise maximum vertex error (FMVE). EucD measures average reconstruction accuracy, while WVE and FMVE capture vertices with persistent errors and the largest error in each frame, respectively. To evaluate sticking artifacts, we identify contact-to-separation events between mesh-distant regions. Event-endpoint tail error (EETE_{25}) measures the hardest 25% of event-endpoint vertices, separation failure rate (SFR) measures the fraction of events that fail to reach the separation threshold, and mean separation deficit (MSD) measures the shortfall in predicted separation. On Motion-80, we report surface CD, HD95, and F-scores. Additional details are provided in Appendix A.3.1.

Baselines. We compare our approach against five representative methods for video-guided 4D mesh generation: Motion 3-to-4 (Chen et al., 2026), Puppeteer (Song et al., 2025), ActionMesh (Sabathier et al., 2026), R-DMesh (Wu et al., 2026a), and Mesh4D (Jiang et al., 2026). These methods cover representative feed-forward and topology-preserving mesh-generation paradigms. Additional details are provided in Appendix A.3.2.

4.2 MAIN RESULTS

4.2.1 RESULTS ON MESHSTICKING-4D

We first evaluate the effectiveness of TRoPE through qualitative comparisons. As shown in Figure 3, our method more faithfully reconstructs the driving poses while reducing sticking artifacts between nearby body parts. Compared with Motion 3-to-4 (Chen et al., 2026), TRoPE better separates the hands from the thighs in the third example, avoiding the stretched hands caused by incorrect coupling between the two regions.

The comparisons also illustrate the importance of recovering motion together with local geometry. In the second example, R-DMesh and Mesh4D produce distorted limb configurations, while Puppeteer exhibits limited arm displacement. TRoPE more closely matches the raised-arm pose and preserves the separation of the arms from the body. The reference and novel views provide complementary evidence that our method improves the reconstructed geometry, reducing the pronounced sticking artifacts.



Figure 3: **Qualitative Comparisons on MeshSticking-4D.** Each pair of columns shows a motion example from the reference (Ref) and novel viewpoints. The top row presents the driving video and the corresponding novel-view ground truth; the remaining rows compare five baseline methods and TRoPE. Compared with the baselines, TRoPE more faithfully recovers articulated poses while preserving separation between nearby body parts, reducing sticking artifacts.

We quantitatively evaluate TRoPE on the 256 test samples of MeshSticking-4D. As shown in Table 1, TRoPE achieves the best EucD, WVE, FMVE, EETE_{25}, and SFR among the compared methods. Relative to Motion 3-to-4, EucD decreases from 59.657 to 54.588, while WVE and FMVE decrease by 39.4% and 36.2%, respectively. These reductions show that the improvement extends beyond average reconstruction accuracy to the vertices with the largest motion errors.

TRoPE also obtains the best EETE_{25} and SFR. EETE_{25} decreases from 137.872 to 98.191, SFR from 49.349% to 34.898%, and MSD from 35.849 to 29.431. Together with the qualitative comparisons, these results demonstrate that TRoPE improves global motion fidelity while reducing errors associated with sticking artifacts. More results and limitations are discussed in Appendix A.4.

4.2.2 Generalization to Motion-80

We further evaluate TRoPE on Motion-80. As shown in Table 2, TRoPE achieves CD 12.238, HD95 34.845, and F-scores of 0.5940 and 0.8656. It outperforms the reference-conditioned ActionMesh, Mesh4D, Puppeteer, and R-DMesh adapters across all reported metrics, while achieving overall performance close to the Motion 3-to-4 baseline. These results demonstrate that TRoPE remains applicable across diverse objects beyond the sticking-focused MeshSticking-4D benchmark.

Table 1: **Quantitative Comparison on MeshSticking-4D.** We report EucD, WVE, and FMVE for motion fidelity, and $EETE_{25}$, SFR, and MSD for sticking robustness, with lower values indicating better performance across all metrics. Results are evaluated on 256 test samples. Bold indicates the best result, and underline indicates the second-best result.

Method	Global Motion Fidelity			Sticking Robustness		
	EucD ↓	WVE ↓	FMVE ↓	$EETE_{25}$ ↓	SFR (%) ↓	MSD ↓
Motion 3-to-4 (Chen et al., 2026)	<u>59.657</u>	<u>285.420</u>	<u>309.116</u>	<u>137.872</u>	49.349	35.849
Puppeteer (Song et al., 2025)	229.205	688.001	736.039	307.215	78.49	35.330
ActionMesh (Sabathier et al., 2026)	122.559	305.978	338.738	188.150	<u>43.73</u>	**27.492**
R-DMesh (Wu et al., 2026a)	147.973	529.012	578.764	223.051	76.80	41.657
Mesh4D (Jiang et al., 2026)	142.397	435.920	485.914	236.312	99.44	49.565
TRoPE (Ours)	**54.588**	**172.874**	**197.135**	**98.191**	**34.898**	<u>29.431</u>

Table 2: **Quantitative Comparison on Motion-80.** We report surface CD, HD95, and F-scores at normalized distances of 0.01 and 0.02. All methods are evaluated on the same 80 sequences and frames 0–15 using one frame-0 similarity-ICP transform reused through time. Lower CD and HD95 and higher F-scores indicate better performance.

Method	CD $\times 10^3$ ↓	HD95 $\times 10^3$ ↓	F@0.01 ↑	F@0.02 ↑
Motion 3-to-4 (Chen et al., 2026)	11.255	30.926	0.6386	0.8891
TRoPE	12.238	34.845	0.5940	0.8656
ActionMesh (Sabathier et al., 2026)	17.957	47.193	0.5370	0.7907
Mesh4D (Jiang et al., 2026)	34.757	111.227	0.3524	0.5764
Puppeteer (Song et al., 2025)	39.609	106.828	0.2523	0.4500
R-DMesh (Wu et al., 2026a)	82.277	259.940	0.1940	0.3296

4.3 Analysis of Topology-Aware Relative Encoding

To understand why TRoPE works, we compare alternative encodings with the same Motion324 host. CVaR is disabled to isolate the query–anchor relation.

We first examine whether an absolute descriptor attached to each mesh location can provide the relation needed for suppressing sticking artifacts. Absolute Cartesian RoPE, Absolute LapPE (Dwivedi & Bresson, 2020) and Absolute HKS RoPE remain close to the Content-only Anchor Attention control. These results show that describing where a query lies does not establish how it is related to the motion source. We next replace the absolute descriptor with a pairwise relation. Euclidean Relative RoPE performs similarly to the Content-only Anchor Attention control, whereas Geodesic Relative RoPE improves quantitative performance, demonstrating that mesh connectivity cannot be replaced by spatial proximity and provides crucial structural information for reducing sticking artifacts. TRoPE further reaches better performance, showing the value of combining mesh connectivity information across multiple structural ranges. Finally, we examine whether the injection mechanism affects how the relation is used. The matched Topology-Relative Additive Bias shares TRoPE's relation, normalization, and diffusion scales, but adds the relation to the attention logits. It performs similarly to the Content-only Anchor Attention control. In contrast, TRoPE applies the same relation inside the query–key compatibility through its rotary phase. As summarized in Table 3, the comparison identifies the key ingredients behind TRoPE: a persistent connectivity-derived relation and direct modulation of feature compatibility.

4.4 Ablation of Topology-Aware Modeling

We further analyze the structural choices of TRoPE. Component-only Relation reduces WVE/SFR from 208.456/44.517% to 195.604/38.104%, showing that component identity provides a coarse information for suppressing cross-part motion leakage. The complete multi-scale relation further reduces these metrics to 181.599/36.152%, demonstrating that within-component connectivity provides a finer structural information beyond component membership. We next investigate the effect

Table 3: **Positional-Encoding Analysis on MeshSticking-4D.** We compare various encoding methods under the same host and training protocol. Results are evaluated on 256 test samples. Lower values indicate better performance. Bold indicates the best result.

Method	Global Motion Fidelity			Sticking Robustness		
	EucD ↓	WVE ↓	FMVE ↓	EETE_{25} ↓	SFR (%) ↓	MSD ↓
Content-only Anchor Attention	55.609	208.456	230.146	118.312	44.517	32.999
Absolute Cartesian RoPE	55.681	208.739	231.014	116.667	45.957	33.491
LapPE (Dwivedi & Bresson, 2020)	**55.240**	207.396	229.150	116.712	46.190	33.248
Absolute HKS RoPE	55.957	207.241	229.023	117.982	44.981	33.638
Euclidean Relative RoPE	55.715	207.962	231.409	118.190	46.515	33.808
Geodesic Relative RoPE	55.312	191.074	215.022	102.974	37.268	30.098
Topology-Relative Additive Bias	56.129	208.702	231.328	119.721	46.143	33.878
TRoPE	55.601	**181.599**	**206.476**	**101.484**	**36.152**	**29.827**

Table 4: **Ablation Analysis on MeshSticking-4D.** We investigate the structural choice of TRoPE. Results are evaluated on 256 test samples. Bold indicates the best result, and underline indicates the second-best result.

Configuration	Global Motion Fidelity			Sticking Robustness		
	EucD ↓	WVE ↓	FMVE ↓	EETE_{25} ↓	SFR (%) ↓	MSD ↓
Content-only Anchor Attention	55.609	208.456	230.146	118.312	44.517	32.999
Component-only Relation	56.042	195.604	218.892	110.509	38.104	30.499
Single-scale Diffusion, $\tau = 1$	<u>55.594</u>	184.127	208.673	102.306	**35.223**	**29.637**
TRoPE, $K = 8$	55.840	186.165	210.596	**100.347**	37.128	29.828
TRoPE, $K = 16$	55.601	<u>181.599</u>	<u>206.476</u>	101.484	<u>36.152</u>	29.827
TRoPE, $K = 32$	**55.361**	**179.965**	**205.558**	<u>100.682</u>	36.524	<u>29.683</u>
TRoPE w/o Local PLTA	56.920	194.891	219.809	107.659	37.314	31.718

of diffusion scale. A single diffusion scale ($\tau = 1$) obtains WVE/FMVE of 184.127/208.673, whereas the multi-scale relation $\{0.25, 1, 4, 16\}$ reaches 181.599/206.476. Other metrics remain nearly unchanged. Multi-scale diffusion therefore improves worst-case trajectory while preserving comparable average and separation quality, indicating that complementary structural ranges provide a more informative relation for severe cross-part errors.

We also vary the number of retained modes over $K \in \{8, 16, 32\}$. The $K = 8$ setting yields slightly worse WVE and FMVE scores. Increasing the spectral resolution improves performance, with diminishing gains beyond $K = 16$. With $K = 8$, fewer spectral modes are retained, primarily capturing low-frequency, coarse-scale structure within each connected component. This makes it harder to distinguish local regions within the same component that are topologically close but differ in their connectivity patterns. We evaluate the contribution of local feature aggregation. As shown in Table 4, the consistent degradation across all metrics shows that local topology aggregation complements the multi-scale relation by preserving neighborhood-level geometric coherence.

5 Conclusion

In this work, we present TRoPE, a topology-aware relative positional encoding for 4D mesh generation that addresses cross-part motion leakage and the resulting sticking artifacts. TRoPE derives stable multi-scale relations from mesh connectivity and injects them directly into query–key compatibility through rotary modulation, while a vertex-wise tail-risk objective emphasizes persistent high-error regions. We also introduced MeshSticking-4D, a benchmark dedicated to evaluating sticking failures. Experiments show that TRoPE improves motion fidelity and sticking robustness on MeshSticking-4D, while maintaining comparable overall generation quality on Motion-80. These results demonstrate that TRoPE consistently reduces sticking artifacts while preserving overall generation quality in 4D generation.

AI USE STATEMENT

In this work, we used generative AI tools, including large language models, to assist with research ideation, mathematical derivations and proofs, code implementation, literature retrieval, and manuscript preparation. Specifically, these tools supported the discussion and refinement of research ideas, the development of mathematical arguments and proof drafts, and the implementation of research code. We also used these tools to identify relevant prior work, draft and revise portions of the manuscript, and improve the clarity and readability of the writing.

The authors reviewed and revised the AI-assisted material incorporated into this work. In particular, we examined the assumptions and reasoning in AI-assisted mathematical derivations and proofs, and reviewed and tested AI-assisted code to check its correctness and consistency with the methodology described in the paper. The authors made the final decisions regarding the methodology, implementation, interpretation of results, and presentation of the work. We take full responsibility for the final content of this paper and its associated research artifacts, including all text, mathematical claims, and code.

ETHICS STATEMENT

This work investigates topology-aware video-to-4D mesh generation for animation and geometry reconstruction. MeshSticking-4D is constructed from Mixamo character and animation assets through synthetic rendering, with additional evaluation on the existing Motion-80 benchmark. Generated content should not be presented as authentic recordings, and downstream applications should respect privacy, consent, and intellectual-property rights. Any use or release of third-party assets should respect their applicable licenses and redistribution restrictions.

REPRODUCIBILITY STATEMENT

To support reproducibility, Secs. 3.1–3.3 describe the motion decoding framework, mesh-supported topological relations, and topology-aware rotary attention, while Sec. 3.4 specifies the training objective and principal optimization settings. Appendix A.1 provides additional mathematical derivations of the multi-scale relation and its rotary formulation. Appendix A.2.1 documents the construction of MeshSticking-4D, including asset collection, motion screening, rendering, and train–test splits with disjoint character and source-animation identities. The contact-to-separation event construction and Motion-80 evaluation protocol are described in Appendices A.2.2 and A.2.3, respectively. Appendix A.2.4 details the implementation of PLTA. Evaluation metrics, baseline implementations, and the configurations of positional-encoding comparisons and structural ablations are provided in Appendices A.3.1, A.3.2, and A.3.3, respectively. Additional qualitative comparisons and failure cases are presented in Appendix A.4. We plan to make our code and benchmark resources publicly available, subject to the applicable licensing and redistribution restrictions of third-party assets.

REFERENCES

Souhaib Attaiki and Maks Ovsjanikov. Understanding and improving features learned in deep functional maps. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 1316–1326, 2023.

Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *IEEE International Conference on Computer Vision Workshops, ICCV 2011 Workshops, Barcelona, Spain, November 6-13, 2011*, pp. 1626–1633, 2011.

Sherwin Bahmani, Ivan Skorokhodov, Victor Rong, Gordon Wetzstein, Leonidas J. 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. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 7996–8006, 2024.

Michael M. Bronstein and Iasonas Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pp. 1704–1711, 2010.

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

Jianqi Chen, Biao Zhang, Xiangjun Tang, and Peter Wonka. V2M4: 4d mesh animation reconstruction from a single monocular video. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pp. 11643–11653, 2025a.

Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 22831–22840, 2025b.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *CoRR*, abs/2012.09699, 2020.

Zhiyang Guo, Jinxu Xiang, Kai Ma, Wengang Zhou, Houqiang Li, and Ran Zhang. Make-it-animatable: An efficient framework for authoring animation-ready 3d characters. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 10783–10792, 2025.

Yanqin Jiang, Li Zhang, Jin Gao, Weiming Hu, and Yao Yao. Consistent4d: Consistent 360° dynamic object generation from monocular video. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

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

Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *CoRR*, abs/2410.11831, 2024.

Xin Kong, Shikun Liu, Xiaoyang Lyu, Marwan Taher, Xiaojuan Qi, and Andrew J. Davison. Eschernet: A generative model for scalable view synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 9503–9513, 2024.

Devin Kreuzer, Dominique Beaini, William L. Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 21618–21629, 2021.

Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025.

Derek Lim, Joshua David Robinson, Lingxiao Zhao, Tess E. Smidt, Suvrit Sra, Haggai Maron, and Stefanie Jegelka. Sign and basis invariant networks for spectral graph representation learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 12919–12928, 2021.

Antoine Liutkus, Ondrej Cífka, Shih-Lun Wu, Umut Simsekli, Yi-Hsuan Yang, and Gaël Richard. Relative positional encoding for transformers with linear complexity. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7067–7079, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

Grégoire Mialon, Dexiong Chen, Margot Selosse, and Julien Mairal. Graphit: Encoding graph structure in transformers. *CoRR*, abs/2106.05667, 2021.

Takeru Miyato, Bernhard Jaeger, Max Welling, and Andreas Geiger. GTA: A geometry-aware attention mechanism for multi-view transformers. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

Henrique Morimitsu, Xiaobin Zhu, Roberto M. Cesar, Xiangyang Ji, and Xu-Cheng Yin. Dpflow: Adaptive optical flow estimation with a dual-pyramid framework. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 17810–17820, 2025.

Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas J. Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Trans. Graph.*, 31(4): 30:1–30:11, 2012.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

Ladislav Rampásek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Jiawei Ren, Cheng 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. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

R Rockafellar and Stan Uryasev. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 01 2000.

Remy Sabathier, David Novotný, Niloy J. Mitra, and Tom Monnier. Actionmesh: Animated 3d mesh generation with temporal 3d diffusion. *CoRR*, abs/2601.16148, 2026.

Aleksandr Safin, Daniel Duckworth, and Mehdi S. M. Sajjadi. Repast: Relative pose attention scene representation transformer. *CoRR*, abs/2304.00947, 2023.

Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. Diffusionnet: Discretization agnostic learning on surfaces. *ACM Trans. Graph.*, 41(3):27:1–27:16, 2022.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pp. 464–468, 2018.

Chaoyue Song, Xiu Li, Fan Yang, Zhongcong Xu, Jiacheng Wei, Fayao Liu, Jiashi Feng, Guosheng Lin, and Jianfeng Zhang. Puppeteer: Rig and animate your 3d models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Jian Sun, Maks Ovsjanikov, and Leonidas J. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Comput. Graph. Forum*, 28(5):1383–1392, 2009.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017.

Rundi Wu, Ruiqi Gao, Ben Poole, Alex Trevithick, Changxi Zheng, Jonathan T. Barron, and Aleksander Holynski. CAT4D: create anything in 4d with multi-view video diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 26057–26068, 2025.

Zijie Wu, Lixin Xu, Puhua Jiang, Sicong Liu, Chunchao Guo, and Xiang Bai. R-dmesh: Video-guided 3d animation via rectified dynamic mesh flow: R-dmesh. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference, SIGGRAPH Conference Papers 2026, Los Angeles, CA, USA, July 19-23, 2026*, pp. 183:1–183:11, 2026a.

Zijie Wu, Chaohui Yu, Fan Wang, and Xiang Bai. Animateanymesh++: A flexible 4d foundation model for high-fidelity text-driven mesh animation. *CoRR*, abs/2604.26917, 2026b.

Yiming Xie, Chun-Han Yao, Vikram Voleti, Huaizu Jiang, and Varun Jampani. SV4D: dynamic 3d content generation with multi-frame and multi-view consistency. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025.

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. *CoRR*, abs/2510.06208, 2025.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 28877–28888, 2021.

Yifei Zeng, Yanqin Jiang, Siyu Zhu, Yuanxun Lu, Youtian Lin, Hao Zhu, Weiming Hu, Xun Cao, and Yao Yao. STAG4D: spatial-temporal anchored generative 4d gaussians. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XXXVI*, volume 15094 of *Lecture Notes in Computer Science*, pp. 163–179, 2024.

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. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pp. 12502–12513, 2025.

A APPENDIX

The Appendix provides additional technical details, derivations, implementation procedures, supplementary experimental settings, and additional results that complement the main paper. It is organized as follows:

- Appendix A.1 presents the derivations and theoretical properties of TRoPE.
- Appendix A.2 describes the construction of MeshSticking-4D, the evaluation protocols, and additional implementation details.
- Appendix A.3 summarizes the supplementary metrics, baseline implementations, and the settings of the encoding and ablation experiments.
- Appendix A.4 reports additional qualitative results and limitations.

A.1 DERIVATION AND PROPERTIES OF TROPE

In this section, we detail the construction of the topology-relative relation and its rotary realization introduced in Secs. 3.2 and 3.3. We first expand the multi-scale diffusion relation, then establish its invariance to admissible spectral basis transformations, and finally derive how the relation modulates query–key compatibility.

Construction of the Multi-scale Diffusion Relation. For each connected component c of the reference mesh, let $\boldsymbol{A}_c$ and $\boldsymbol{D}_c$ denote its adjacency and degree matrices. The symmetric normalized graph Laplacian is

$$\boldsymbol{L}_c = \boldsymbol{I} - \boldsymbol{D}_c^{-1/2}\boldsymbol{A}_c\boldsymbol{D}_c^{-1/2} = \boldsymbol{\Phi}_c\boldsymbol{\Lambda}_c\boldsymbol{\Phi}_c^{\top}. \tag{13}$$

We remove the constant mode and retain the non-trivial eigenpairs indexed by $\mathcal{I}_c$. Let $\bar{\lambda}_{c,m}$ denote the eigenvalues normalized by the median positive retained eigenvalue of component c. For a diffusion scale τ, the normalized spectral response is

$$w_{\tau,c,m} = \frac{\exp(-\tau\bar{\lambda}_{c,m})}{\left\|\left(\exp(-\tau\bar{\lambda}_{c,r})\right)_{r\in\mathcal{I}_c}\right\|_2}. \tag{14}$$

The corresponding diffusion feature of a mesh location i is

$$\boldsymbol{\psi}_{c,\tau}(i) = \left(\sqrt{w_{\tau,c,m}}\,\widetilde{\boldsymbol{\phi}}_{c,m}(i)\right)_{m\in\mathcal{I}_c}, \tag{15}$$

where $\widetilde{\boldsymbol{\phi}}_{c,m}$ denotes the retained component-wise spectral coordinate used by TRoPE (Sharp et al., 2022). Given a query location i and an anchor location a on the same component, their normalized relation at scale τ is

$$r_{ia}^{(\tau)} = \frac{\|\boldsymbol{\psi}_{c,\tau}(i) - \boldsymbol{\psi}_{c,\tau}(a)\|_2}{\max\left\{\|\boldsymbol{\psi}_{c,\tau}(i)\|_2 + \|\boldsymbol{\psi}_{c,\tau}(a)\|_2, \epsilon\right\}}. \tag{16}$$

Averaging over the fixed set of diffusion scales $\mathcal{T}$ gives the final query–anchor relation

$$r_{ia} = \frac{1}{|\mathcal{T}|}\sum_{\tau\in\mathcal{T}} r_{ia}^{(\tau)}. \tag{17}$$

Thus, r_{ia} compares the structural neighborhoods of the two endpoints.

Invariance to Spectral Basis Transformations. The eigenvectors associated with a repeated eigenvalue are not uniquely defined (Lim et al., 2023). Any admissible change of basis within such an eigenspace can be represented by an orthogonal matrix $\boldsymbol{Q}$ satisfying $\boldsymbol{Q}^{\top}\boldsymbol{Q} = \boldsymbol{I}$. Because the

diffusion response is shared within the repeated eigenspace, the corresponding diffusion features transform jointly as

$$\boldsymbol{\psi}'_{c,\tau}(i) = \boldsymbol{\psi}_{c,\tau}(i)\boldsymbol{Q}. \tag{18}$$

For two locations i and a, the transformed numerator in Eq. 16 satisfies

$$\begin{aligned}\left\|\boldsymbol{\psi}'_{c,\tau}(i) - \boldsymbol{\psi}'_{c,\tau}(a)\right\|_2^2 &= \left\|\left(\boldsymbol{\psi}_{c,\tau}(i) - \boldsymbol{\psi}_{c,\tau}(a)\right)\boldsymbol{Q}\right\|_2^2 \\ &= \left(\boldsymbol{\psi}_{c,\tau}(i) - \boldsymbol{\psi}_{c,\tau}(a)\right)\boldsymbol{Q}\boldsymbol{Q}^\top\left(\boldsymbol{\psi}_{c,\tau}(i) - \boldsymbol{\psi}_{c,\tau}(a)\right)^\top \\ &= \left\|\boldsymbol{\psi}_{c,\tau}(i) - \boldsymbol{\psi}_{c,\tau}(a)\right\|_2^2 .\end{aligned} \tag{19}$$

The same argument preserves the two feature norms in the denominator. Therefore,

$$r'^{(\tau)}_{ia} = r^{(\tau)}_{ia}, \qquad r'_{ia} = r_{ia}. \tag{20}$$

A sign flip is the special case in which $\boldsymbol{Q}$ is diagonal with entries in $\{-1, +1\}$. Consequently, the relation used by TRoPE depends on the structural comparison between two locations, with its invariance to admissible spectral basis transformations.

Rotary Realization of Relation-conditioned Compatibility. Let $\{\omega_f\}_{f=1}^{F_\mathrm{r}}$ denote the fixed rotary frequencies. The relation-dependent phase for frequency f is

$$\theta_{ia,f} = 2\pi\omega_f r_{ia}. \tag{21}$$

For one two-dimensional rotary channel pair, we define

$$\boldsymbol{R}(\theta) = \begin{bmatrix}\cos\theta & \sin\theta \\ -\sin\theta & \cos\theta\end{bmatrix}. \tag{22}$$

Let $\boldsymbol{q} = (q_1, q_2)^\top$ and $\boldsymbol{k} = (k_1, k_2)^\top$ denote the corresponding query and key channel pairs. Their relation-conditioned compatibility is

$$\boldsymbol{q}^\top\boldsymbol{R}(\theta)\boldsymbol{k} = (q_1k_1 + q_2k_2)\cos\theta + (q_1k_2 - q_2k_1)\sin\theta. \tag{23}$$

The cosine coefficient measures the usual aligned feature similarity, whereas the sine coefficient captures the cross-channel interaction. The structural relation determines the two coefficients through the phase, while the query and key projections determine the feature-dependent amplitudes. Applying the block rotation to all rotary channel pairs gives

$$\widetilde{\ell}^h_{ia} = \frac{\left(\boldsymbol{q}^h_i\right)^\top\boldsymbol{R}(\theta_{ia})\boldsymbol{k}^h_a}{\sqrt{d_h}}, \tag{24}$$

where $\boldsymbol{R}(\theta_{ia})$ is block diagonal over the rotary frequencies. Since each block is orthogonal,

$$\left\|\boldsymbol{R}(\theta_{ia})\boldsymbol{k}^h_a\right\|_2 = \left\|\boldsymbol{k}^h_a\right\|_2. \tag{25}$$

The rotary operation therefore changes the compatibility between the query and the motion source without rescaling the key. The resulting attention weights and motion readout are

$$\widetilde{p}^h_{ia} = \frac{\exp\left(\widetilde{\ell}^h_{ia}\right)}{\sum_{b\in\mathcal{A}}\exp\left(\widetilde{\ell}^h_{ib}\right)}, \qquad \widetilde{\boldsymbol{o}}^h_{i,t} = \sum_{a\in\mathcal{A}}\widetilde{p}^h_{ia}\boldsymbol{v}^h_{a,t}. \tag{26}$$

Thus, the topology relation directly affects which motion source is compatible with a query.

Temporal Reuse and Cross-component Neutrality. The relation is computed from the reference connectivity and is therefore independent of the articulated pose. For every frame t,

$$r_{ia}^{(t)} = r_{ia}^{\mathrm{ref}}. \tag{27}$$

This separates the persistent structural relation from the instantaneous motion content.

For locations belonging to different connected components, or for invalid spectral locations, we define $r_{ia} = 0$. It follows from Eq. 21 that

$$r_{ia} = 0 \Longrightarrow \theta_{ia,f} = 0 \Longrightarrow \boldsymbol{R}(\theta_{ia}) = \boldsymbol{I}, \tag{28}$$

and hence

$$\widetilde{\ell}_{ia}^{h} = \frac{\left(\boldsymbol{q}_i^h\right)^\top \boldsymbol{k}_a^h}{\sqrt{d_h}}. \tag{29}$$

Cross-component pairs therefore retain the ordinary content-based compatibility. The equations above establish the form used in Sec. 3.3: TRoPE supplies a persistent and basis-invariant query–anchor relation, and injects it into motion compatibility through a norm-preserving rotary phase.

A.2 Additional Implementation Details

A.2.1 Details of MeshSticking-4D

MeshSticking-4D is a benchmark dataset specifically designed to evaluate sticking artifacts. It consists of 10080 training samples and 256 test samples. Each sample contains a reference mesh, a 32-frame mesh sequence, and synchronized monocular views. We construct the benchmark in four steps.

Step 1: Character and Animation Collection. Following the Mixamo asset collection and character refinement procedure provided with Make-It-Animatable[1] (Guo et al., 2025), we collect 95 rigged, textured characters and 2,453 source animations from Mixamo. We refine and triangulate the character meshes for motion extraction, while retaining the textured assets for rendering. We use the 44 characters that pass the texture and mesh checks to construct the final clip pool. Animations must contain enough frames to form a 32-frame sequence sampled at a stride of two.

Step 2: Motion Generation and Candidate Screening. We apply each source animation to a character rig and bake its vertex trajectories. From the resulting motion, we extract 32-frame candidates at a stride of two, advancing the starting position by 16 sampled frames. Each candidate is normalized using its sequence bounding box while preserving its original vertex indexing and face connectivity.

To select motions that expose possible sticking artifacts, we assign vertices to six coarse body regions: torso, head, left and right arms, and left and right legs. We sample up to 3,000 vertices and search their 16 nearest spatial neighbors. For pairs from different regions, excluding the torso–head boundary, we measure their distance over representative 12-frame windows. A pair is counted as a dynamic near-approach when its minimum distance lies in $(0.06, 0.12]$ and its distance changes by at least 0.08 within the window. We retain candidates with at least eight such pairs per window on average and with at least three pairs in no fewer than half of the evaluated windows. This screening criterion measures the frequency of challenging close approaches; the contact-to-separation events used for evaluation are defined separately in the next section.

We additionally require substantial non-rigid motion while maintaining clean geometry and consistent framing. Specifically, the 95th-percentile articulation displacement, normalized by body size, must be at least 0.25, while normalized centroid span must not exceed 0.50. Median and minimum

[1] https://huggingface.co/datasets/jasongzy/Mixamo

frame occupancy must be at least 0.82 and 0.60, respectively. We limit the 95th-percentile edge strain to 0.20, the maximum fraction of collapsed faces to 0.05, and the normalized 99th-percentile temporal acceleration to 1.00. Candidates containing non-finite vertices are excluded.

Step 3: Clip Selection and Rendering. Among candidates passing these criteria, we prioritize dynamic near-approach density and temporal coverage while distributing the selected clips across characters and source animations. We retain non-overlapping windows from each character–animation pair. For every selected clip, we render textured images with Blender Cycles and save the corresponding mesh trajectory. Images are produced at 256×256 resolution. The camera entries contain eight distinct near-frontal viewpoints. All views use the same mesh trajectory, timestamps, and sequence-level normalization.

Step 4: Identity-disjoint Split Construction. We partition character identities and source-animation identities independently using a fixed, deterministic ordering. The primary test split contains 256 clips whose character and source-animation identities are both absent from the training set. The final training set contains 10080 distinct samples after excluding test identities and incomplete samples. For training, we use a ground-truth contact-to-separation census to assign each clip a sampling multiplicity of one, two, or three. This yields 16223 entries in the training manifest. The resulting training and test manifests are fixed for all comparisons on MeshSticking-4D.

A.2.2 Details of Contact-to-Separation Event

We construct sticking-sensitive events from ground-truth mesh trajectories. For each pair of surface locations, we compute its reference-mesh connectivity relation and its framewise Euclidean distance. We retain pairs that are more than 20 mesh hops apart or belong to different connected components.

A contact-to-separation event is recorded when the pair first enters a near-contact state with distance below 0.02 and subsequently reaches a separated state with distance above 0.10. The event endpoint is associated with the corresponding surface locations and their ground-truth trajectories.

The event set is used for every prediction on the testset. This ensures that EETE_{25}, SFR, and MSD measure the behavior of the ground-truth contact-to-separation events.

A.2.3 Motion-80 Evaluation Protocol

Motion-80 is the public evaluation benchmark released with Motion 3-to-4 (Chen et al., 2026). We follow its reference-conditioned first-16 protocol. For each sequence, all methods receive the first-frame reference mesh and the first 16 driving frames, and predict the corresponding dynamic mesh sequence.

We apply one frame-0 similarity-ICP transform and reuse it for all evaluated frames. For each predicted and ground-truth mesh, we sample 10,000 surface points using the same sampling procedure. We report surface Chamfer Distance (CD), HD95, and F-scores at normalized distances of 0.01 and 0.02. All methods are evaluated on the same 80 sequences, frames, reference meshes, and scoring procedure.

A.2.4 Implementation of PLTA

PLTA is implemented as a local sparse topology-aware attention block adapted from AnimateAnyMesh++ (Wu et al., 2026b). Given the reference mesh connectivity, we construct a sparse graph for the sampled surface points. For each destination point i, the graph contains a self-edge and up to 16 source points collected from the reference face graph. Candidate points are ordered by mesh-hop distance, with Euclidean distance used only to break ties. Missing entries are padded with the self-edge. For an edge (i, j) with hop distance h_{ij}, we add the fixed topological bias $b_{ij} = \log(\gamma^{h_{ij}})$, where $\gamma = 0.5$.

For input features $\mathbf{x}_i$, PLTA applies pre-normalized multi-head attention restricted to the local neighborhood $\mathcal{N}(i)$:

$$\alpha_{ij}^{(r)} = \text{softmax}_{j \in \mathcal{N}(i)} \left(\frac{\left(\boldsymbol{W}_q^{(r)} \,\text{LN}(\boldsymbol{x}_i) \right)^\top \left(\boldsymbol{W}_k^{(r)} \,\text{LN}(\boldsymbol{x}_j) \right)}{\sqrt{d_h}} + b_{ij} \right). \tag{30}$$

where r indexes the attention head. The messages from all heads are concatenated, projected, and added to the input through a residual connection. A second residual connection then applies a LayerNorm–linear–GELU–linear feed-forward network. The default configuration uses 12 attention heads and zero dropout. PLTA is inserted at the decoder-synchronization stage: it first enriches the dense reference-point features, after which FPS selects 1024 mesh-supported anchors from the enriched feature field. The same features are therefore used to construct dense queries and anchor representations.

A.3 Supplementary Experimental Settings

In this section, we supplement the evaluation metrics and the implementation details of the baseline, positional encoding and ablation experients. Unless otherwise specified, all methods are evaluated with the same reference meshes and driving videos.

A.3.1 Evaluation Metrics

Evaluation on MeshSticking-4D. We evaluate motion fidelity using three correspondence-based metrics. Let $\widehat{v}_{t,i}$ and $v_{t,i}$ denote the predicted and ground-truth positions of vertex i at frame t, respectively. Euclidean Distance (EucD) measures the average vertex reconstruction error over all frames and vertices. Worst-Vertex Error (WVE) is the largest temporal-mean vertex error, and therefore emphasizes vertices with persistent reconstruction errors. Framewise Maximum Vertex Error (FMVE) first takes the maximum vertex error within each frame and then averages these maxima over time.

To evaluate sticking artifacts, we use the contact-to-separation events defined in Sec. A.2. Event-Endpoint Tail Error (EETE_{25}) measures the average error over the hardest 25% of event-endpoint vertices. Separation Failure Rate (SFR) is the fraction of events whose predicted separation does not reach the prescribed separation threshold. Mean Separation Deficit (MSD) measures the average shortfall between the predicted separation and the required separation at the event endpoint. Lower values indicate better performance for all six metrics.

Evaluation on Motion-80. Following the official Motion-80 protocol, we report surface Chamfer Distance (CD), the 95th-percentile Hausdorff Distance (HD95), and the F-score at normalized distances of 0.01 and 0.02. CD and HD95 measure surface-level reconstruction accuracy, while the F-scores measure the fraction of points that satisfy the corresponding distance threshold. Lower CD and HD95 and higher F-scores indicate better performance.

A.3.2 Baseline Implementations

We provide implementation details for the baseline methods used in our experiments. Unless otherwise specified, we use the official implementations or released checkpoints and keep the reference mesh, synchronized driving view, frame sampling, and output conversion consistent with TRoPE. All predicted sequences are converted to the common vertex-correspondence format before evaluation.

- **Implementation Details of Motion 3-to-4.** Motion 3-to-4 (Chen et al., 2026) is a feed-forward video-to-4D framework that encodes a monocular video and a reference mesh into compact motion tokens and predicts temporally coherent vertex trajectories. We use it as both the primary host and the content-based control. Specifically, we initialize the model from the released checkpoint and fine-tune it on MeshSticking-4D using the same training manifest, 12-frame stride-two windows, and 4,096 surface queries as TRoPE. For the content-based control, we disable PLTA, topology-derived features, topology-relative modulation, and vertex-wise CVaR, while retaining the content-only anchor attention. The resulting vertex trajectories are exported directly in the common reference-mesh correspondence format.
- **Implementation Details of Puppeteer.** Puppeteer (Song et al., 2025) combines automatic skeleton prediction, topology-aware skinning-weight estimation, and differentiable animation optimization. We reproduce the released rigging and skinning pipelines and reuse the predicted rig for all clips sharing the same object identity. For each sample, the reference mesh is converted to Puppeteer's coordinate convention and is animated using the synchronized 12-frame camera-0 video

window. Following the released animation pipeline, we use DPFlow optical flow (Morimitsu et al., 2025), Video-Depth-Anything-Large depth estimation (Chen et al., 2025b), and CoTracker3 point tracks (Karaev et al., 2024), and optimize the animation for 20 iterations at a rendering resolution of 128×128. The final linear-blend-skinning vertices are transformed back to the benchmark coordinate system and exported with the original face topology.

- **Implementation Details of ActionMesh.** ActionMesh (Sabathier et al., 2026) is a temporal 3D diffusion model that first predicts synchronized time-varying shape latents and then decodes them into deformations of a reference mesh. We use the released mesh-and-video pipeline with the benchmark reference mesh supplied directly, without the image-to-3D shape-generation stage, fine-tuning, or test-time optimization. The input consists of the same camera-0 frame window used for the other baselines; the 12 selected frames are padded by repeating the last frame to satisfy the model's 16-frame input interface. We use the official RMBG preprocessing, 30 temporal denoising steps, guidance scale 7.5, and bfloat16 inference. The fixed coordinate conversion between the benchmark and ActionMesh conventions is applied before inference and inverted before scoring. The decoded mesh sequence is mapped back to the original vertex ordering and face topology.
- **Implementation Details of R-DMesh.** R-DMesh (Wu et al., 2026a) models video-guided animation with a conditional mesh representation that separates the base mesh, relative trajectories, and a rectification offset, followed by a rectified-flow generation stage. For the MeshSticking-4D evaluation, we use the reference-conditioned R-DMesh pipeline and adapt its mesh decoder on the same weighted training manifest as Motion 3-to-4. The decoder is initialized from the released VAE checkpoint and fine-tuned for five epochs with batch size two and learning rate 10^{-4}, while the backbone remains frozen. At inference, we use 4,096 supported mesh points, 512 trajectories, and 64 rectified-flow steps for the 12-frame benchmark window. Its predicted vertices are projected to the common reference-mesh correspondence format before scoring.
- **Implementation Details of Mesh4D.** Mesh4D (Jiang et al., 2026) encodes the complete animation into a compact latent representation and predicts a deformation field conditioned on the video and the mesh reconstructed from the first frame. We use the released reference-mesh-conditioned configuration and supply the benchmark first-frame mesh directly, thereby keeping the output vertex support and topology fixed across time. The synchronized camera-0 frames are converted to the RGBA sequence layout expected by the released loader, and the official Mesh4D checkpoints are used without additional fine-tuning. The native image-to-shape generation stage is bypassed because the reference mesh is fixed by the benchmark protocol. The generated deformation sequence is exported in the reference-mesh correspondence format and evaluated with the same metrics as TRoPE.

A.3.3 Encoding and Structural Variants

We further provide implementation details for the encoding and ablation experiments. All experiments use the same Motion 3-to-4 host, checkpoint initialization, data manifests, and training budget. The common protocol uses 12-frame windows, 4096 query points, 1024 FPS anchors, and 42600 optimization steps. CVaR is disabled in these experiments unless explicitly stated otherwise.

Implementation of positional encoding experiments.

- **Implementation Details of Content-only Anchor Attention.** We remove the additional positional relation and retain content-based compatibility between dense queries and mesh-supported anchors. The anchor construction and local feature aggregation modules remain unchanged.
- **Implementation Details of Absolute Cartesian RoPE.** We use the absolute Cartesian coordinates of query and anchor locations as independent rotary inputs. The rotary frequency budget and attention architecture are matched to those of TRoPE.
- **Implementation Details of Laplacian Positional Encoding.** We inject the Laplacian eigenvectors as absolute decoder coordinates through additive feature modulation. This experiment corresponds to LapPE and uses the same spectral resolution as the default configuration (Dwivedi & Bresson, 2020).
- **Implementation Details of Absolute HKS RoPE.** We construct an absolute scalar heat-kernel-signature field from the reference mesh and use it as the rotary input for query and anchor features (Bronstein & Kokkinos, 2010; Sun et al., 2009; Su et al., 2024). The rotary channel budget is kept identical to that of TRoPE.

- **Implementation Details of Euclidean Relative RoPE.** We replace the topology-derived relation with the pairwise Euclidean distance between each query and anchor. The relation is injected through the same rotary attention operator as TRoPE.
- **Implementation Details of Geodesic Relative RoPE.** We compute the shortest-path distance on the reference mesh graph for each query–anchor pair and use it as the rotary relation. Compared to TRoPE, all remaining components are unchanged.
- **Implementation Details of Topology-Relative Additive Bias.** We use the same connectivity-derived relation, diffusion scales, spectral normalization, and component handling as TRoPE. Instead of converting the relation into a rotary phase, we add it directly to the attention logits.
- **Implementation Details of TRoPE.** TRoPE uses the proposed multi-scale mesh-connectivity relative relation as the rotary modulation of query–key compatibility. This configuration is the default positional encoding experiment used in the analysis.

Implementation of ablation studies.

- **Implementation Details of Component-only Relation.** We retain connected-component membership as the structural relation while removing within-component diffusion structure. Cross-component pairs use the same handling as the other experiments.
- **Implementation Details of Single-scale Diffusion.** We retain the topology-relative relation but use only the diffusion scale $\tau = 1$, instead of the default multi-scale set.
- **Implementation Details of Spectral Resolution.** We vary the number of retained spectral modes per connected component over $K \in \{8, 16, 32\}$. All other diffusion and rotary settings remain fixed.
- **Implementation Details of TRoPE without Local PLTA.** We remove the local topology-aware feature aggregation module while retaining the topology-relative rotary relation, anchor construction, and training protocol.

For all encoding and ablation experiments, the reference connectivity is computed once and reused across frames.

A.4 Additional Experimental Results

A.4.1 More Qualitative Comparisons

Beyond the qualitative comparisons presented in the main paper, we provide additional examples in Fig. 4 and Fig. 5 to further evaluate the effectiveness of TRoPE. All methods receive the same reference mesh and synchronized monocular driving video, and are rendered from identical reference and novel viewpoints. The results show that TRoPE consistently recovers the driving poses while preserving local surface geometry and part separation.

Across the examples in Fig. 4, TRoPE maintains coherent limb configurations and produces consistent silhouettes in both viewpoints. Several baselines exhibit insufficient pose deformation or distortions in the arms, legs, and torso, especially under large articulated movements. TRoPE more faithfully reduces the sticking artifacts of nearby body parts while following the motion shown in the driving video. The examples in Fig. 5 further cover diverse motion patterns. TRoPE remains stable across extended limbs and bent-over poses, whereas the baseline methods may introduce pose deviations or sticking artifacts. These additional results provide further evidence that TRoPE improves 4D generation while preserving topology-consistent geometry and reducing sticking artifacts.

A.4.2 Limitations

Although TRoPE substantially improves robustness to sticking artifacts, it still has limitations in challenging motion reconstruction. First, when the driving video contains severe self-contact, occlusion, or highly ambiguous motion evidence, the topology-relative relation cannot fully disambiguate the motion sources, and residual sticking artifacts may persist, as illustrated in Fig. 6(a). Second, suppressing cross-part motion leakage does not guarantee accurate local deformation. When a large limb displacement exists, the model may preserve part separation while assigning an inaccurate local trajectory, resulting in limb deformation, as shown in Fig. 6(b).



Figure 4: More qualitative results using TRoPE.



Figure 5: More qualitative results using TRoPE.

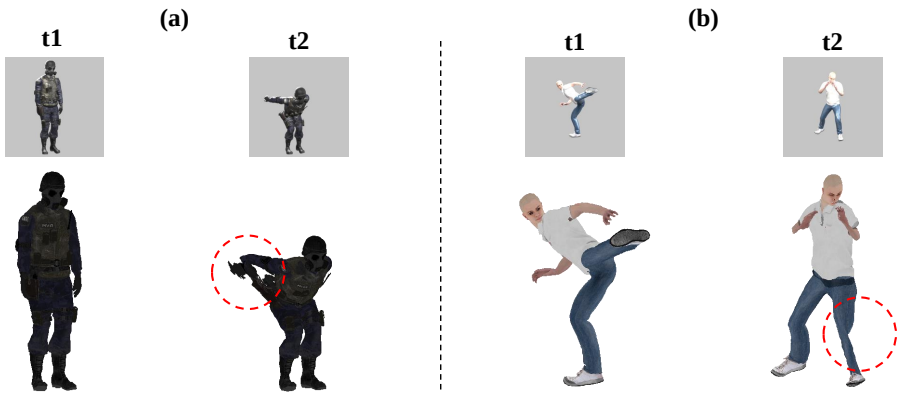

Figure 6: **Limitations.** (a) In challenging cases, sticking artifacts are mitigated to some extent but still persist. (b) In some cases, although sticking artifacts are suppressed, the limbs become deformed.