SPECTRALWEAVE: A SHARED SPECTRAL FRAMEWORK FOR NATIVE 4D MESH GENERATION

Anonymous authors
Paper under double-blind review

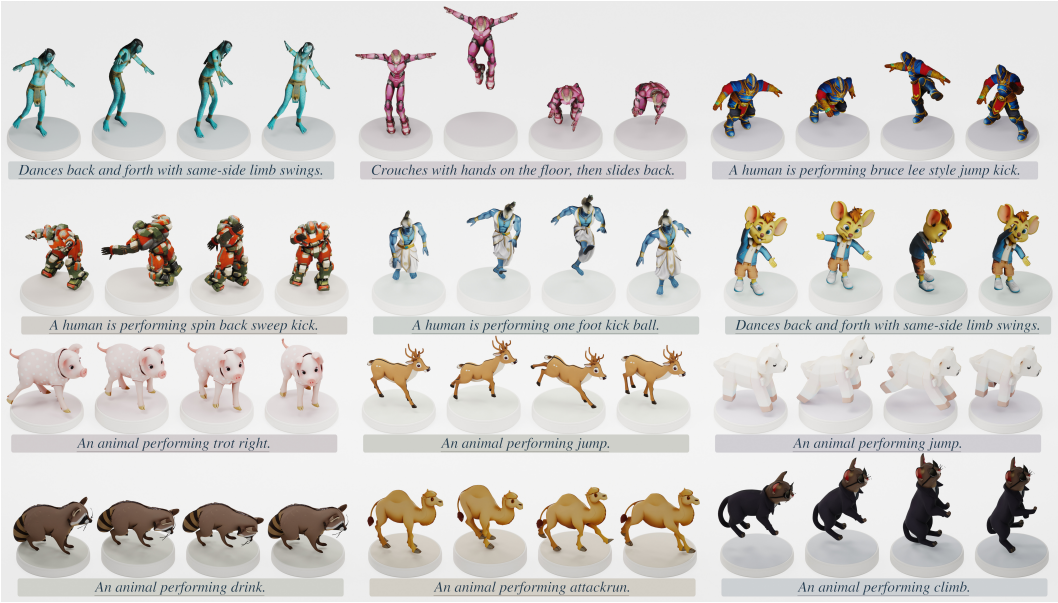


Figure 1: **SpectralWeave** animates diverse **unrigged** humanoid and animal meshes from text while preserving their native topology. Input meshes are generated by TRELLIS (Xiang et al., 2025) from FLUX-generated images.

ABSTRACT

Native 4D mesh generation animates an unrigged input mesh from a semantic action condition while preserving its vertex indexing and connectivity. Sharing motion across independently meshed inputs requires common learning coordinates that retain reconstruction on each input topology. We introduce **SpectralWeave**, a framework for native 4D mesh generation that builds its spectral motion representation from the geometric structure inherent in the input mesh. A native Laplacian basis computed from each mesh remains fixed throughout the sequence, representing motion through compact coefficient trajectories that decode directly on the input vertices without a learned mesh decoder. However, native coefficients remain difficult to compare across identities because their bases differ. We align native spectral coordinates through jointly estimated adapters, establishing shared coordinates while preserving each mesh's native reconstruction subspace. We then use partial surface correspondences for alignment and condition motion diffusion on the action and input morphology. We decode the generated shared coefficients and refine the sequence to preserve local details. Experiments in separate animal and human domains demonstrate better motion accuracy and surface quality on new identities than the compared methods. Ablations support shared spectral coordinates, morphology conditioning, and geometric refinement. See the Project page for more demos: https://anoymouscell.github.io/spectralweave.

1 INTRODUCTION

Native 4D mesh generation directly animates a static input mesh from a semantic action condition while preserving its vertex indexing and connectivity. Unlike video-mediated pipelines, which obtain dense motion evidence in image space before reconstructing or fitting a dynamic 3D representation (Jiang et al., 2024; Uzolas et al., 2025; Sabathier et al., 2026), a native generator must infer

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.

a distribution of surface trajectories from the input mesh and action condition alone. This setting calls for a motion representation that supports learning across independently meshed inputs and reconstruction on their original vertices. The input geometry provides a natural basis for constructing such a representation.

Recent native generators learn compact motion representations through mesh autoencoders or deformation controls. AnimateAnyMesh (Wu et al., 2025) learns a dynamic mesh autoencoder, BiMotion (Wang et al., 2026) encodes B-spline trajectories with a VAE, and TapMo (Zhang et al., 2024) learns handles and skinning weights. Spectral representations provide another approach. We investigate how mesh-specific Laplacian bases can support motion learning across independently meshed inputs while retaining direct reconstruction on each input mesh.

We therefore introduce **SpectralWeave**, a spectral motion representation framework for native 4D generation across independently meshed identities. Our design connects shared motion learning with direct reconstruction on each input mesh. We compute a native Laplacian basis from the input geometry and represent surface displacements through coefficient trajectories. The basis remains fixed throughout the sequence, so each coefficient weights the same spatial pattern at every frame. This provides compact motion coordinates and direct reconstruction on the input vertices without a learned mesh decoder.

However, native bases differ in sign, eigenspace orientation, and spatial structure, so their displacement coefficients cannot be used directly as shared motion coordinates. We connect these coordinate systems through identity-specific adapters constrained by pairwise functional maps. To share motion across identities while retaining their own geometry, we use partial correspondences without requiring every surface region to have a counterpart. Vertex-area weights account for uneven mesh sampling, while commonness scores reduce the influence of unreliable matches. At inference, we connect a new input mesh to the shared coordinate system using its geometry, enabling 4D motion generation adapted to that identity. We then condition the diffusion model on the requested action and morphology tokens derived from the input mesh. We decode the generated shared coefficients on input mesh and refine the sequence to preserve local details (Fig. 2).

Experiments in separate animal and human domains demonstrate our performance on new identities. Compared with the strongest baseline (Bimotion finetune) for trajectory error in our evaluation, we reduce V-RMSE by 36.0% on Animal and 40.1% on Human.

Our main contributions are as follows.

- We introduce a spectral representation framework for native 4D motion generation across independently meshed identities, supporting shared motion learning with direct reconstruction on each input topology.
- We connect shared motion coordinates to each mesh's native coefficients through jointly estimated adapters constrained by partial functional maps, preserving its native reconstruction subspace.
- We demonstrate motion generation ability in both animal and human domains, with ablations supporting shared spectral coordinates, morphology conditioning, and geometric refinement.

2 Related Work

4D Mesh Generation. DNF (Zhang et al., 2025b) models dynamic shapes unconditionally, while Motion2VecSets (Cao et al., 2024) reconstructs and tracks surfaces from point cloud sequences. Given an input mesh and action condition, TapMo (Zhang et al., 2024) uses learned handles and skinning weights, AnimateAnyMesh (Wu et al., 2025) learns a dynamic mesh autoencoder, and BiMotion (Wang et al., 2026) encodes B-spline control points with a VAE. These native generators learn deformation controls or mesh decoders from data. *SpectralWeave uses the geometric structure inherent in the input mesh to construct a native Laplacian basis for both motion representation and surface reconstruction.* Methods using video obtain dense motion evidence or pretrained video priors. SC4D (Wu et al., 2024) and DreamMesh4D (Li et al., 2024) optimize dynamic representations from monocular video, while GVFDiffusion (Zhang et al., 2025a) and Mesh4D (Jiang et al., 2026) learn to generate or reconstruct dynamic 3D content from video. Animate3D (Jiang et al., 2024) synthesizes videos from multiple views given a mesh and text, MotionDreamer (Uzolas et al., 2025) uses video diffusion features, and ActionMesh (Sabathier et al., 2026) generates animated meshes from video, with extensions conditioned on text and input assets. *SpectralWeave instead generates mesh*

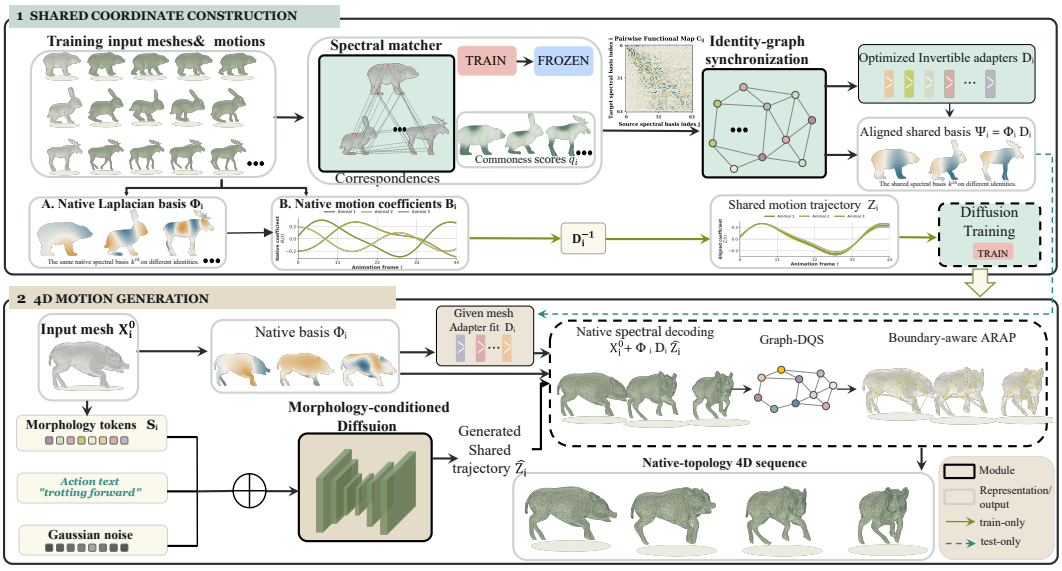


Figure 2: **SpectralWeave pipeline.** Joint adapter synchronization connects native displacement coefficients to shared coordinates. Diffusion conditioned on action and morphology tokens generates trajectories. The adapter and native basis decode them, preserving vertex indexing and connectivity.

trajectories directly from the input mesh and action condition, without relying on video observations, multiview video synthesis, or video diffusion guidance.

Motion Representations Across Identities. SMPL (Loper et al., 2015) and SMAL (Zuffi et al., 2017) provide compact pose spaces through templates and rigs. Automatic rigging adapts a generic skeleton (Baran & Popovic, 2007) or predicts joints, hierarchy, and skinning weights (Xu et al., 2020). Puppeteer (Song et al., 2025) combines rigging with animation, AnyTop (Gat et al., 2025) models motion over heterogeneous skeleton graphs, and SkelMo (Tao et al., 2026) generates motion for rigged shapes using video guidance. These methods share motion through explicit skeletal representations. Sparse controls include embedded deformation graphs (Sumner et al., 2007), TapMo's learned handles, and SMF's kinetic keypoints (Muralikrishnan et al., 2025). AnimateAnyMesh and BiMotion use learned latent spaces and decoders to share motion across meshes. *SpectralWeave instead builds its motion representation from each input mesh's native Laplacian basis. Aligning the displacement coefficients provides shared coordinates for learning motion across independently meshed identities, while morphology conditioning adapts these shared motion patterns to each input shape.*

Spectral Motion Representation. Laplacian eigenfunctions provide compact coordinates for geometry and deformation (Vallet & Lévy, 2008). SpecTrHuMS (Lemeunier et al., 2023) predicts and completes registered human mesh sequences, while SMD (Xue et al., 2025) generates Laplacian coefficient trajectories on a common SMPL topology. Both require a common mesh topology and cannot directly generate motion on independently meshed inputs with different vertex counts and connectivity. DiffusionNet (Sharp et al., 2022) learns surface descriptors, and diffusion methods predict or regularize functional maps (Zhuravlev et al., 2025; Pierson et al., 2025). SignNet and BasisNet (Lim et al., 2022) support predictions invariant to basis ambiguities, while SyncSpecCNN (Yi et al., 2017) aligns spectral features for prediction across meshes. *SpectralWeave connects shared displacement coordinates to each mesh's native motion space through adapters. These coordinates support conditioned trajectory generation while preserving the native reconstruction subspace.*

3 Proposed Method

Problem Setup. For identity i, let $\mathcal{M}_i = (\mathbf{X}_i^0, \mathbf{F}_i)$ denote its input mesh, where $\mathbf{X}_i^0 \in \mathbb{R}^{N_i \times 3}$ contains the rest-pose vertex positions and $\mathbf{F}_i$ is the triangular face set. Given $\mathcal{M}_i$ and a semantic action condition, we generate an L-frame mesh sequence $\hat{\mathbf{X}}_i \in \mathbb{R}^{L \times N_i \times 3}$ with the same vertex indexing and connectivity as the input mesh. We index animation frames by $\ell \in \{0, \ldots, L-1\}$. To describe how the given shape deforms over time, we parameterize motion by displacement

from the input mesh. For an observed training frame, $\Delta\mathbf{X}_{i,\ell} = \mathbf{X}_{i,\ell} - \mathbf{X}_i^0$; for a generated frame, $\Delta\hat{\mathbf{X}}_{i,\ell} = \hat{\mathbf{X}}_{i,\ell} - \mathbf{X}_i^0$. The generated sequence starts from the input mesh, which we enforce by fixing its first-frame displacement to zero. Training uses input meshes paired with motion sequences. At inference, we generate motion for a new identity from its input mesh and requested action.

Overview. We propose SpectralWeave, a spectral representation framework illustrated in Fig. 2 for native 4D mesh generation. We represent displacements in each input mesh's native Laplacian basis and use an identity-specific adapter to connect its coefficients to shared coordinates. we constrain the adapters using area- and commonness-weighted partial correspondences, without requiring every region to match. Action- and morphology-conditioned diffusion generates shared coefficient trajectories, which the input mesh's adapter and native basis decode before geometric refinement preserves local shape .

3.1 Native Spectral Motion Coordinates

We represent surface displacement as a weighted sum of spatial patterns defined on the input vertices (Rustamov, 2007). Low-frequency Laplacian eigenfunctions vary smoothly over the surface, allowing us to approximate spatially smooth displacements with $K \ll N_i$ patterns. For identity i, let $\mathbf{M}_i = \mathrm{diag}(\boldsymbol{m}_i)$ be the lumped mass matrix with vertex-wise Voronoi areas, and $\mathbf{L}_i$ the cotangent stiffness matrix. Both are computed from the normalized input mesh. We obtain the first K eigenfunctions, ordered by increasing eigenvalue, from

$$\mathbf{L}_i\mathbf{\Phi}_i = \mathbf{M}_i\mathbf{\Phi}_i\mathbf{\Lambda}_i, \qquad \mathbf{\Phi}_i^\mathsf{T}\mathbf{M}_i\mathbf{\Phi}_i = \mathbf{I}, \tag{1}$$

where $\mathbf{\Phi}_i \in \mathbb{R}^{N_i \times K}$ contains these spatial patterns and $\mathbf{\Lambda}_i = \mathrm{diag}(\lambda_{i,1}, \ldots, \lambda_{i,K})$ are eigenvalues. We keep this basis fixed throughout the sequence, so each eigenfunction defines the same spatial pattern at every frame.

We approximate the displacement at frame ℓ as a linear combination of the first K Laplacian eigenfunctions. The native displacement coefficients $\mathbf{B}_{i,\ell} \in \mathbb{R}^{K\times 3}$ weight these eigenfunctions along the three coordinate axes, giving

$$\Delta\mathbf{X}_{i,\ell}^{(K)} = \mathbf{\Phi}_i\mathbf{B}_{i,\ell}. \tag{2}$$

The displacement of each frame is represented by $K \times 3$ coefficients regardless of vertex count and reconstructed on the input mesh. During training, we compute these coefficients from observed displacements by solving the weighted reconstruction problem in Sec. 3.2.

However, native bases and their spectral coefficients are not directly comparable across meshes due to sign ambiguities, eigenspace rotations, and differences in spatial structure. The model must then account for coordinate differences alongside action dynamics, making training more difficult and hindering the learning of shared motion patterns that generalize to new identities. We therefore align the native representations in a shared coordinate system. Details are described in Appendices A and B.

3.2 Cross-Identity Spectral Alignment

To enable shared motion learning while retaining native reconstruction, we introduce an invertible adapter $\mathbf{D}_i \in \mathbb{R}^{K\times K}$. Its inverse converts native displacement coefficients $\mathbf{B}_{i,\ell}$ to shared displacement coefficients $\mathbf{Z}_{i,\ell} \in \mathbb{R}^{K\times 3}$:

$$\mathbf{Z}_{i,\ell} = \mathbf{D}_i^{-1}\mathbf{B}_{i,\ell}, \qquad \mathbf{B}_{i,\ell} = \mathbf{D}_i\mathbf{Z}_{i,\ell}. \tag{3}$$

During training, $\mathbf{D}_i^{-1}$ converts native coefficients to shared coefficients; during generation, $\mathbf{D}_i$ converts shared coefficients back to native coefficients for reconstruction. Substituting this conversion into Eq. (2) gives the aligned basis $\mathbf{\Psi}_i$ and displacement reconstruction:

$$\mathbf{\Psi}_i = \mathbf{\Phi}_i\mathbf{D}_i, \qquad \Delta\mathbf{X}_{i,\ell}^{(K)} = \mathbf{\Psi}_i\mathbf{Z}_{i,\ell}. \tag{4}$$

Since $\mathbf{D}_i$ is invertible, the aligned basis preserves all displacements representable by the native basis. We solve for the adapters using surface correspondences, encouraging aligned basis functions to describe similar spatial patterns on matched regions. This helps the model learn shared motion patterns across identities.

Correspondence Constraints for Alignment. We estimate alignment constraints from partial surface correspondences, using vertex areas and commonness scores to weight their contribution. We train a DiffusionNet matcher (Sharp et al., 2022; Cao et al., 2023) to predict vertex descriptors and

commonness scores $\boldsymbol{q}_i \in (0,1)^{N_i}$. We weight each vertex by $m_i(v)q_i(v)$ and collect these weights in the diagonal matrix $\mathbf{W}_i = \mathrm{diag}(\boldsymbol{m}_i \odot \boldsymbol{q}_i)$. Vertex areas account for uneven mesh sampling, while commonness scores reduce the influence of vertices without reliable counterparts. Using descriptor similarity and these vertex weights, we estimate partial surface correspondences, represented by a soft correspondence matrix $\mathbf{\Pi}_{ij} \in \mathbb{R}^{N_j \times N_i}$. We express these correspondences in the native bases through a functional map $\mathbf{C}_{ij}$ (Ovsjanikov et al., 2012), which represents the transported source basis functions in mesh j's native basis. We use the target vertex weights $\mathbf{W}_j$ to estimate this map by solving

$$\mathbf{C}_{ij} = \underset{\mathbf{C}\in\mathbb{R}^{K\times K}}{\arg\min} \left\{ \left\| \mathbf{W}_j^{1/2} \left(\mathbf{\Phi}_j \mathbf{C} - \mathbf{\Pi}_{ij}\mathbf{\Phi}_i\right) \right\|_F^2 + \eta \|\mathbf{C}\|_F^2 \right\}. \tag{5}$$

The weighted error emphasizes reliably matched surface area, and the ridge term η stabilizes the solve. The resulting functional maps provide constraints for adapter estimation.

Joint Adapter Synchronization. Using consistency across functional maps, we jointly solve for the adapters that define the shared motion coordinates. Each functional map provides a pairwise constraint $\mathbf{C}_{ij}\mathbf{D}_i \approx \mathbf{D}_j$. To reduce the influence of unreliable functional maps, we assign each pair a confidence weight ω_{ij}. We compute the reverse map $\mathbf{C}_{ji}$ and evaluate consistency by mapping coefficients from mesh i to mesh j and back through $\mathbf{C}_{ji}\mathbf{C}_{ij}$, with smaller errors giving larger weights. We combine these weighted constraints over the matched identity pairs and solve

$$\min_{\{\mathbf{D}_i\}} \quad \mathcal{L}_{\mathrm{sync}} = \sum_{(i,j)\in\mathcal{E}_{\mathrm{id}}} \omega_{ij}\, \rho(\mathbf{C}_{ij}\mathbf{D}_i, \mathbf{D}_j)\,, \text{ subject to } \sigma_{\mathrm{lo}} \le \sigma_k(\mathbf{D}_i) \le \sigma_{\mathrm{hi}}, \forall i, k. \tag{6}$$

Here, ρ applies a Huber penalty to the normalized difference, limiting the influence of unreliable pairs. The positive singular value bounds keep the adapters invertible and limit coefficient scaling.

Shared Motion Targets. With the matcher and adapters fixed, we compute native displacement coefficients $\mathbf{B}_{i,\ell}$ by weighted projection of each identity's observed displacements, then convert them to shared displacement coefficients $\mathbf{Z}_{i,\ell}$ as training targets.

$$\mathbf{B}_{i,\ell} = \underset{\mathbf{B}\in\mathbb{R}^{K\times 3}}{\arg\min} \left\{ \left\| \mathbf{W}_i^{1/2} \left(\mathbf{\Phi}_i \mathbf{B} - \Delta\mathbf{X}_{i,\ell}\right) \right\|_F^2 + \eta_{\mathrm{ridge}} \|\mathbf{B}\|_F^2 \right\}, \quad \mathbf{Z}_{i,\ell} = \mathbf{D}_i^{-1}\mathbf{B}_{i,\ell}. \tag{7}$$

The ridge term η_{ridge} stabilizes the solve. Decoding the shared coefficients gives the same displacement as decoding the native coefficients: $\mathbf{\Psi}_i\mathbf{Z}_{i,\ell} = \mathbf{\Phi}_i\mathbf{B}_{i,\ell}$.

Connecting a new Identity. At inference, we compute the new mesh's adapter using functional maps from the frozen matcher and the same alignment constraints. The adapter and native basis then decode generated shared coefficients into motion on the input mesh. Alignment details are given in Appendix A.

3.3 Morphology-Conditioned Motion Diffusion

Alignment provides a shared coordinate system, but the displacements reconstructed from shared coefficients still rely on the input mesh. The same action can require different amplitudes and combinations of shared coefficients across identities. We therefore condition the diffusion model on the action and the input mesh's geometry and spectral properties.

Text Condition. We represent the requested action using a learned action embedding and a text embedding from a frozen CLIP encoder (Radford et al., 2021). We apply a trainable projection to the CLIP embedding and combine it with the learned action embedding. The resulting action condition guides the denoiser through cross attention.

Morphology Condition. To describe the input geometry in shared coordinates, we project the input vertex positions $\mathbf{X}_i^0$ onto the native basis and convert the resulting geometry coefficients:

$$\mathbf{S}_i^{\mathrm{geo}} = \mathbf{D}_i^{-1} \left(\mathbf{\Phi}_i^{\mathsf{T}}\mathbf{M}_i\mathbf{\Phi}_i + \eta_{\mathrm{ridge}}\mathbf{I}\right)^{-1} \mathbf{\Phi}_i^{\mathsf{T}}\mathbf{M}_i\mathbf{X}_i^0. \tag{8}$$

Here, $\mathbf{S}_i^{\mathrm{geo}} \in \mathbb{R}^{K\times 3}$ contains the shared geometry coefficients of the static input vertex positions. We use the mass matrix $\mathbf{M}_i$ without commonness weighting, so that regions with low commonness scores still contribute to the morphology condition. Aligned basis functions can differ in spatial variation and displacement scale across identities. We therefore augment the shared geometry coefficients with the Dirichlet energies and area-weighted norms of the aligned basis functions to form morphology tokens for the denoiser:

$$\mathbf{S}_i = \left[\mathbf{S}_i^{\mathrm{geo}},\ \mathrm{diag}(\mathbf{\Psi}_i^{\mathsf{T}}\mathbf{L}_i\mathbf{\Psi}_i),\ \sqrt{\mathrm{diag}(\mathbf{\Psi}_i^{\mathsf{T}}\mathbf{M}_i\mathbf{\Psi}_i)}\right] \in \mathbb{R}^{K\times 5}. \tag{9}$$

Here, diag extracts a matrix's diagonal as a column vector. The Dirichlet energy describes spatial variation, while the area-weighted norm measures the displacement magnitude produced by a unit shared coefficient. These properties relate shared coefficients to the displacements reconstructed on the input mesh. All morphology tokens are computed from the input mesh and its adapter, including for new identities.

Denoising Objective. We train the denoiser with v prediction (Salimans & Ho, 2022). Let $\mathbf{V}$ and $\hat{\mathbf{V}}$ denote the denoising target and prediction for one frame. The denoising loss is

$$\mathcal{L}_{\text{vpred}} = \text{MSE}(\hat{\mathbf{V}}, \mathbf{V}). \tag{10}$$

We recover and denormalize the shared coefficients for displacement reconstruction.

The same error in shared coefficients can produce different displacement errors across identities. We therefore supervise displacement reconstruction on each input mesh using its aligned basis. For one frame, the predicted and target shared coefficients decode to $\Delta\hat{\mathbf{X}}^{(K)} = \mathbf{\Psi}_i\hat{\mathbf{Z}}$ and $\Delta\mathbf{X}^{(K)} = \mathbf{\Psi}_i\mathbf{Z}$. The surface loss is

$$\mathcal{L}_{\text{surf}} = \left\| \mathbf{W}_i^{1/2} \left(\Delta\hat{\mathbf{X}}^{(K)} - \Delta\mathbf{X}^{(K)} \right) \right\|_F^2 = \left\| \mathbf{W}_i^{1/2} \mathbf{\Psi}_i \left(\hat{\mathbf{Z}} - \mathbf{Z} \right) \right\|_F^2 . \tag{11}$$

The loss sums these squared displacement errors using the vertex weights $\mathbf{W}_i$, directly supervising reconstruction on the input mesh. Denoiser and loss details are given in Appendix A.

3.4 Deterministic Geometry Refinement and Inference

For a new input mesh, we compute its adapter and morphology tokens, generate shared coefficients for the requested action, and denormalize them. We decode the shared displacement coefficients through the adapter and native basis, then add the reconstructed displacements to the input vertex positions:

$$\hat{\mathbf{X}}_{i,\ell} = \mathbf{X}_i^0 + \mathbf{\Phi}_i \mathbf{D}_i \hat{\mathbf{Z}}_{i,\ell}. \tag{12}$$

To refine the local details, we use a deformation graph to define local surface regions on the input mesh. For each frame, we solve for a rigid transformation of each region, aligning its input vertex positions in $\mathbf{X}_i^0$ with its decoded vertex positions in $\hat{\mathbf{X}}_{i,\ell}$. Dual quaternion skinning (DQS) (Kavan et al., 2007) blends these transformations and applies them to the input mesh to obtain $\mathbf{X}_{i,\ell}^{\text{dqs}}$. We then use as rigid as possible (ARAP) optimization to preserve local edge geometry while keeping vertex positions close to $\mathbf{X}_{i,\ell}^{\text{dqs}}$:

$$\left(\mathbf{X}_{i,\ell}^{\text{refine}}, \{\hat{\mathbf{R}}_v\} \right) = \underset{\mathbf{X},\, \{\mathbf{R}_v \in \text{SO}(3)\}}{\arg\min} \left[\mathcal{L}_{\text{ARAP}}(\mathbf{X}, \{\mathbf{R}_v\}) + \mathcal{L}_{\text{att}}(\mathbf{X}; \mathbf{X}_{i,\ell}^{\text{dqs}}) \right]. \tag{13}$$

Here, $\mathbf{R}_v$ is the local rotation at vertex v, and the ARAP term assigns higher weights to edges between different surface regions. Optimizing vertex positions directly allows corrections beyond the subspace spanned by the native basis. The refined frames $\mathbf{X}_{i,\ell}^{\text{refine}}$ form the final sequence, preserving the input mesh's vertex indexing and connectivity.

4 Experiments

4.1 Experimental Setup

Datasets. To evaluate the generality of our framework, we use animal and human datasets totaling approximately 260k frames. Animal uses data from DeformingThings4D (DT4D) (Li et al., 2021), while Human uses mesh animations processed from Mixamo (Adobe, n.d.). We train a separate model for each dataset using the same representation, architecture, and objective.

Baselines and evaluation protocol. We compare SpectralWeave with the direct mesh generators TapMo (Zhang et al., 2024), AnimateAnyMesh (Wu et al., 2025), and BiMotion (Wang et al., 2026), and the video-mediated generators Animate3D (Jiang et al., 2024) and ActionMesh (Sabathier et al., 2026). We evaluate two settings. Held-out ID–Action tests identity–action combinations withheld from training for identities included in the training set. New-ID tests identities excluded from all training stages, including matcher and motion-model training. Baselines use official implementations and released checkpoints. We additionally fine-tune BiMotion on our training split using its official interface and the same training budget as SpectralWeave, providing a strong baseline adapted to our datasets. All quantitative 3D metrics use the same dataset test cases and reference motions. Complete

Table 1: **Quantitative comparison on held-out identity–action combinations and held-out (New-ID) identities.** Held-out ID–Action evaluates withheld combinations for training identities; New-ID evaluates identities excluded from all training stages. Surface metrics use a topology-agnostic comparison on New-ID. Bold marks the best result within each dataset. † denotes video-mediated methods.

Method	Held-Out Motion	New-ID Motion	New-ID Surface			
	V-RMSE↓	V-RMSE↓	CD-L2↓	HD95↓	Normal↑	F@2%↑
Animal dataset						
TapMo(ICLR'24)	15.660	15.350	6.620	14.270	64.590	37.740
AnimateAnyMesh(ICCV'25)	12.900	12.970	5.980	13.050	70.730	48.980
BiMotion(CVPR'26)	12.470	12.300	5.940	12.990	72.930	49.810
BiMotion Finetune	12.055	11.838	5.075	11.034	68.563	49.978
Animate3D(NeurIPS'24)†	14.030	13.810	6.320	13.730	68.530	43.980
ActionMesh(CVPR'26)†	15.910	15.140	6.380	13.790	65.910	40.520
SpectralWeave(Ours)	**6.660**	**7.580**	**3.550**	**7.530**	**76.450**	**62.990**
Human dataset						
TapMo(ICLR'24)	25.100	24.580	13.690	26.710	55.130	26.620
AnimateAnyMesh(ICCV'25)	20.280	20.500	10.600	21.740	56.880	32.130
BiMotion(CVPR'26)	19.250	19.580	9.360	19.750	58.400	36.230
BiMotion Finetune	6.454	8.904	4.864	9.876	60.378	59.524
Animate3D(NeurIPS'24)†	20.760	21.220	10.750	21.150	55.160	27.200
ActionMesh(CVPR'26)†	20.560	19.740	10.360	20.600	55.030	27.300
SpectralWeave(Ours)	**2.640**	**5.330**	**2.470**	**5.190**	**68.010**	**72.310**

sequences are linearly interpolated to 45 frames in normalized time. Image-generated meshes (Xiang et al., 2025) are used for qualitative comparisons.

Evaluation metrics. We evaluate agreement with the reference trajectories using mass-weighted vertex trajectory root-mean-square error (V-RMSE) and motion cosine (Cos.). Surface metrics include Chamfer distance (CD-L2) and 95th-percentile Hausdorff distance (HD95) for surface distance, normal consistency (Normal) for orientation, and F-score (F@2%) for surface overlap. Representation ablations use the motion and surface metrics reported in their tables. We also assess rendered video quality using three reference-free VBench (Huang et al., 2024) metrics: subject consistency (SC), motion smoothness (MS), and dynamic degree (DD). Motion and surface metrics are displayed at $100\times$ their original scale. VBench and human rankings assess rendered motion and perceived quality.

Implementation. Unless otherwise specified, we use $K = 32$ motion modes and 50 DDIM sampling steps. Experiments run on NVIDIA RTX A6000 GPUs. Training settings are given in Appendix B.

4.2 Comparison with Prior Work

Reference-trajectory agreement. We achieve the lowest V-RMSE on both datasets under the Seen-ID & Unseen-Motion and New-ID settings (Table 1). On New-ID, we reduce V-RMSE by 36.0% on Animal and 40.1% on Human compared with BiMotion Finetune, the best baseline on this metric. These results show better agreement with reference trajectories on unseen meshes than a strong baseline adapted to the same training data.

Surface accuracy. We achieve the best results on all four New-ID surface metrics in both datasets (Table 1). Compared with BiMotion Finetune, the best baseline for CD-L2, we reduce CD-L2 by 30.0% on Animal and 49.2% on Human. Lower HD95, higher normal consistency, and higher F@2% further indicate closer agreement with the reference surfaces in distance, orientation, and overlap.

Reference-free video evaluation. To complement reference-trajectory metrics, we evaluate rendered animations using three reference-free VBench metrics: subject consistency (SC), motion smoothness (MS), and dynamic degree (DD). SpectralWeave achieves the highest SC and DD on both datasets (Table 2). DD measures the proportion of rendered videos classified as dynamic; our scores exceed BiMotion Finetune by 5.75 and 19.23 percentage points on Animal and Human, respectively. These higher dynamic scores are accompanied by high motion smoothness: SpectralWeave ranks first on Human and is comparable to AnimateAnyMesh on Animal (99.56 vs. 99.58).

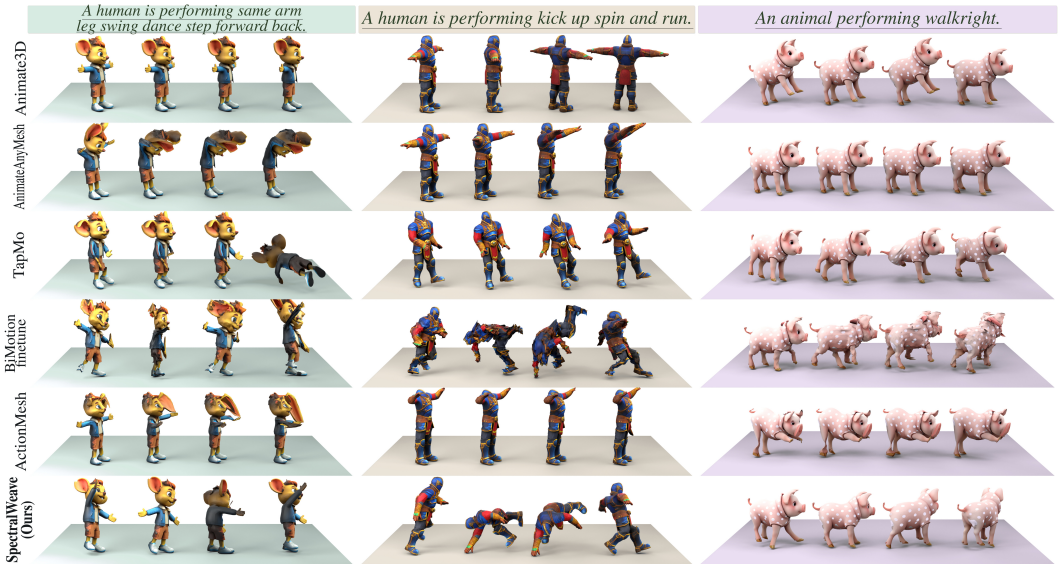


Figure 3: **Qualitative comparisons.** Input meshes are generated by image-to-3d method TRELLIS (Xiang et al., 2025) from FLUX-generated images.

User study. We ask 20 participants to rank five of seven anonymized methods on 30 cases (20 human and 10 animal), separately for action alignment, motion amplitude, and subject consistency. Table 13 in Appendix D reports the percentage of rankings placing each method first. SpectralWeave receives the largest share for action alignment (75.3%) and subject consistency (43.3%), and is most often ranked first for motion amplitude (43.8%). The amplitude criterion measures movement extent rather than overall quality.

Table 2: VBench evaluation on the same test sequences, with four views per sequence. SC: subject consistency; MS: motion smoothness; DD: dynamic degree. Larger DD indicates more detected motion.

	Animal			Human		
Method	SC ↑	MS ↑	DD ↑	SC ↑	MS ↑	DD ↑
Animate3D	94.69	99.51	3.46	93.32	99.52	4.54
AnimateAnyMesh	98.13	**99.58**	6.26	94.99	99.56	4.76
TapMo	90.98	98.81	52.54	90.04	98.98	45.60
BiMotion Finetune	98.56	99.39	53.02	95.31	99.14	49.83
ActionMesh	88.86	99.44	10.38	89.98	99.37	9.12
SpectralWeave(Ours)	**98.69**	99.56	**58.77**	**97.71**	**99.62**	**69.06**

Qualitative comparison. Fig. 3 compares methods on meshes generated from images, using the same input mesh and action prompt within each group. SpectralWeave produces articulated body and limb motion in both human and animal examples while preserving the input shapes. Some baselines show limited pose changes or visible distortions around the head and torso. Additional comparisons and generations are provided in Appendix C, and the user study protocol in Appendix D.

4.3 Analysis of the Spectral Representation

Spectral alignment. We compare five alignment settings on unseen test identities (Table 3). The *w/o adapter* variant sets $\mathbf{D}_i = \mathbf{I}$ while retaining native bases and commonness weights; *w/o spectral alignment* also removes the matcher and commonness weighting. *Single-reference alignment* fixes the adapter of one training mesh to its value from joint synchronization and estimates the remaining adapters independently using only constraints with this reference mesh. *Orthogonal synchronization* retains the identity graph and matching constraints but restricts adapters to $\mathbf{D}_i^\top \mathbf{D}_i = \mathbf{I}$. Each setting retrains the same generator architecture with recomputed motion coefficients, morphology tokens, and normalization statistics, using the same training budget, sampling, and refinement. Full achieves the best values across all four metrics on both datasets. Relative to single-reference alignment, it reduces V-RMSE by 13.4% on Animal and 7.4% on Human; relative to orthogonal synchronization, the reductions are 5.4% and 10.3%. These comparisons support joint constraints and bounded scaling, respectively, under the evaluated setting.

Table 3: **Spectral alignment ablation on unseen identities from the dataset test splits.** Bold indicates the best result within each dataset.

Setting	V-RMSE↓	Cos.↑	CD-L2↓	HD95↓
Animal test dataset				
w/o adapter ($D_i = I$)	10.651	45.701	4.914	10.537
w/o spectral alignment	11.251	41.912	5.142	11.181
Single-reference alignment	8.748	59.735	4.014	8.600
Orthogonal synchronization	8.015	62.427	3.974	8.122
Full (Ours)	**7.580**	**67.360**	**3.550**	**7.530**
Human test dataset				
w/o adapter ($D_i = I$)	7.995	85.545	3.660	7.931
w/o spectral alignment	19.922	31.963	12.067	21.619
Single-reference alignment	5.753	93.430	2.650	5.535
Orthogonal synchronization	5.944	92.604	2.890	5.807
Full (Ours)	**5.330**	**94.490**	**2.470**	**5.190**

Table 4: **Geometric refinement ablation on new identities.** Jerk is the third-difference coordinate RMS.

Animal test dataset				
Stage	CD-L2↓	HD95↓	Jerk↓	Edge-P95↓
Spectral decoding	3.641	7.709	1.358	42.703
+ Geometry refinement	**3.550**	**7.530**	**0.429**	**22.907**
Human test dataset				
Stage	CD-L2↓	HD95↓	Jerk↓	Edge-P95↓
Spectral decoding	2.501	5.210	2.666	53.197
+ Geometry refinement	**2.470**	**5.190**	**1.100**	**24.043**

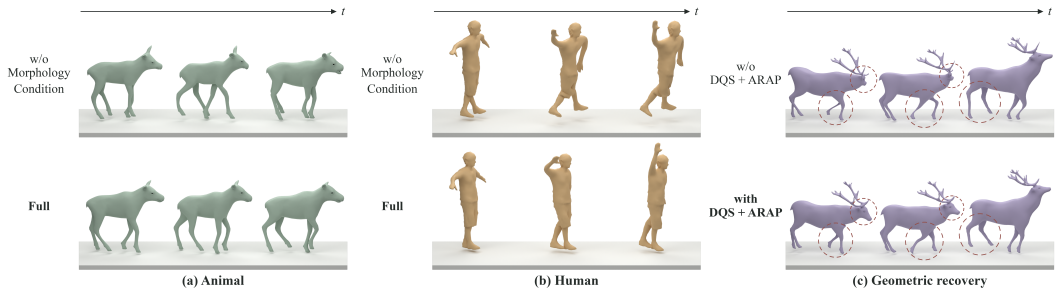


Figure 4: **Morphology conditioning and geometric refinement. (a,b)** Animal and human examples without morphology conditioning (top) and with the full procedure (bottom). **(c)** Before (top) and after (bottom) DQS and ARAP refinement. Dashed circles highlight local differences.

Partial correspondence. We compare partial correspondence with a forced full-correspondence control that requires all surface regions to participate in matching and functional-map estimation. Partial correspondence reduces all reported errors on both datasets (Table 10 in Appendix D), lowering V-RMSE in low-commonness regions by 8.7% on Animal and 3.8% on Human. The larger improvement on Animal is consistent with greater structural variation across its identities.

Morphology conditioning. Without morphology conditioning, V-RMSE increases by 30.1% on Animal and 48.0% on Human, and both CD-L2 and HD95 increase (Table 11 in Appendix D). These results support using the input shape to condition motion generation. Figs. 4(a) and (b) show qualitative comparisons.

Geometric refinement. Table 4 compares direct spectral decoding with DQS and ARAP refinement on new identities. Refinement reduces Edge-P95 by 46.4% on Animal and 54.8% on Human, while also lowering jerk in both datasets. Fig. 4 (c) illustrates the corresponding improvements in local details around the head and legs. Ablation settings are given in Appendix D.

5 Conclusion and Discussion

We presented SpectralWeave, a spectral motion representation that combines native mesh reconstruction with shared coefficient learning. Jointly estimated invertible adapters align native coordinates through partial functional-map constraints while preserving each mesh's reconstruction subspace. With morphology-conditioned diffusion and geometric refinement, the framework improves reference-trajectory and surface agreement on held-out identities in separate animal and human evaluations.

Increasing the number of modes does not consistently improve generation on new identities, motivating multiscale representations that combine shared motion patterns with finer deformations. Our evaluation covers animal and human motion in sequences of fixed length. Future work could explore broader shapes and actions, as well as longer sequences with transitions between actions.

AI USE STATEMENT

We used generative AI tools solely to improve the language and readability of the manuscript. We reviewed all AI-assisted edits to ensure that they preserved the intended meaning and take full responsibility for the final content of this work.

REFERENCES

Adobe. Mixamo, n.d. URL https://www.mixamo.com/. Accessed September 16, 2026.

Ilya Baran and Jovan Popovic. Automatic rigging and animation of 3d characters. *ACM Trans. Graph.*, 26(3):72, 2007.

Dongliang Cao, Paul Roetzer, and Florian Bernard. Unsupervised learning of robust spectral shape matching. *arXiv preprint arXiv:2304.14419*, 2023.

Wei Cao, Chang Luo, Biao Zhang, Matthias Nießner, and Jiapeng Tang. Motion2vecsets: 4d latent vector set diffusion for non-rigid shape reconstruction and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20496–20506, 2024.

Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of computation*, 87(314):2563–2609, 2018.

Inbar Gat, Sigal Raab, Guy Tevet, Yuval Reshef, Amit Haim Bermano, and Daniel Cohen-Or. Anytop: Character animation diffusion with any topology. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–10, 2025.

Qixing Huang, Fan Wang, and Leonidas Guibas. Functional map networks for analyzing and exploring large shape collections. *ACM Transactions on Graphics (ToG)*, 33(4):1–11, 2014.

Ruqi Huang, Jing Ren, Peter Wonka, and Maks Ovsjanikov. Consistent zoomout: Efficient spectral map synchronization. In *Computer Graphics Forum*, volume 39, pp. 265–278. Wiley Online Library, 2020.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024.

Yanqin Jiang, Chaohui Yu, Chenjie Cao, Fan Wang, Weiming Hu, and Jin Gao. Animate3d: Animating any 3d model with multi-view video diffusion. *Advances in Neural Information Processing Systems*, 37:125879–125906, 2024.

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

Ladislav Kavan, Steven Collins, Jiří Žára, and Carol O'Sullivan. Skinning with dual quaternions. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pp. 39–46, 2007.

Artiom Kovnatsky, Michael M Bronstein, Alexander M Bronstein, Klaus Glashoff, and Ron Kimmel. Coupled quasi-harmonic bases. In *Computer Graphics Forum*, volume 32, pp. 439–448. Wiley Online Library, 2013.

Clément Lemeunier, Florence Denis, Guillaume Lavoué, and Florent Dupont. Spectrhums: Spectral transformer for human mesh sequence learning. *Computers & Graphics*, 115:191–203, 2023.

Yang Li, Hikari Takehara, Takafumi Taketomi, Bo Zheng, and Matthias Nießner. 4dcomplete: Non-rigid motion estimation beyond the observable surface. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12686–12696. IEEE, 2021.

Zhiqi Li, Yiming Chen, and Peidong Liu. Dreammesh4d: Video-to-4d generation with sparse-controlled gaussian-mesh hybrid representation. *Advances in Neural Information Processing Systems*, 37:21377–21400, 2024.

Derek Lim, Joshua Robinson, Lingxiao Zhao, Tess Smidt, Suvrit Sra, Haggai Maron, and Stefanie Jegelka. Sign and basis invariant networks for spectral graph representation learning. *arXiv preprint arXiv:2202.13013*, 2022.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):248, 2015.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Sanjeev Muralikrishnan, Niladri Shekhar Dutt, and Niloy J Mitra. Smf: Template-free and rig-free animation transfer using kinetic codes. *ACM Transactions on Graphics (TOG)*, 44(6), 2025.

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

Emery Pierson, Lei Li, Angela Dai, and Maks Ovsjanikov. Diffumatch: Category-agnostic spectral diffusion priors for robust non-rigid shape matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5745–5756, October 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Raif M Rustamov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Symposium on geometry processing*, volume 257, pp. 225–233, 2007.

Remy Sabathier, David Novotny, Niloy J Mitra, and Tom Monnier. Actionmesh: Animated 3d mesh generation with temporal 3d diffusion. *arXiv preprint arXiv:2601.16148*, 2026.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

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

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. *arXiv preprint arXiv:2508.10898*, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Robert W Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. In *ACM siggraph 2007 papers*, pp. 80–es. 2007.

Ye Tao, Yuxin Yao, Kendong Liu, Dapeng Wu, and Junhui Hou. Skelmo: Universal skeletal motion generation for 3d rigged shapes. *arXiv preprint arXiv:2606.01518*, 2026.

Lukas Uzolas, Elmar Eisemann, and Petr Kellnhofer. Motiondreamer: Exploring semantic video diffusion features for zero-shot 3d mesh animation. In *2025 International Conference on 3D Vision (3DV)*, pp. 893–904. IEEE, 2025.

Bruno Vallet and Bruno Lévy. Spectral geometry processing with manifold harmonics. In *Computer Graphics Forum*, volume 27, pp. 251–260. Wiley Online Library, 2008.

Miaowei Wang, Qingxuan Yan, Zhi Cao, Yayuan Li, Oisin Mac Aodha, Jason J Corso, and Amir Vaxman. Bimotion: B-spline motion for text-guided dynamic 3d character generation. *arXiv preprint arXiv:2602.18873*, 2026.

Zijie Wu, Chaohui Yu, Yanqin Jiang, Chenjie Cao, Fan Wang, and Xiang Bai. Sc4d: Sparse-controlled video-to-4d generation and motion transfer. In *European Conference on Computer Vision*, pp. 361–379. Springer, 2024.

Zijie Wu, Chaohui Yu, Fan Wang, and Xiang Bai. Animateanymesh: A feed-forward 4d foundation model for text-driven universal mesh animation. *arXiv preprint arXiv:2506.09982*, 2025.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21469–21480. IEEE, 2025.

Zhan Xu, Yang Zhou, Evangelos Kalogerakis, Chris Landreth, and Karan Singh. Rignet: Neural rigging for articulated characters. *ACM Transactions on Graphics*, 39(4), 2020. ISSN 0730-0301.

Kebing Xue, Hyewon Seo, Cedric Bobenrieth, and Guoliang Luo. Shape-conditioned human motion diffusion model with mesh representation. In *Computer Graphics Forum*, volume 44, pp. e70065. Wiley Online Library, 2025.

Li Yi, Hao Su, Xingwen Guo, and Leonidas J Guibas. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2282–2290, 2017.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12502–12513, 2025a.

Jiaxu Zhang, Shaoli Huang, Zhigang Tu, Xin Chen, Xiaohang Zhan, Gang Yu, and Ying Shan. Tapmo: Shape-aware motion generation of skeleton-free characters. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Xinyi Zhang, Naiqi Li, and Angela Dai. Dnf: Unconditional 4d generation with dictionary-based neural fields. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26047–26056, 2025b.

Aleksei Zhuravlev, Zorah Lähner, and Vladislav Golyanik. Denoising functional maps: Diffusion models for shape correspondence. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26899–26909, 2025.

Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 5524–5532. IEEE, 2017.

APPENDIX CONTENTS

A Method Details **14**
A.1 Notation . . . 14
A.2 Native Spectral Motion Coordinates . . . 14
A.3 Cross-Identity Spectral Alignment . . . 14
A.4 Morphology-Conditioned Motion Diffusion . . . 15
A.4.1 Complete Surface-Aware Objective . . . 16

B Implementation Details **16**
B.1 Coordinate conventions . . . 16
B.2 Training and Inference . . . 16

C Additional Results **17**
C.1 Additional Comparisons . . . 17
C.2 Additional SpectralWeave Generations . . . 18
C.3 Motion Diversity . . . 19
C.4 Efficiency . . . 20

D EXperiments **20**
D.1 Ablation Configurations . . . 21
D.2 User Study . . . 23

A METHOD DETAILS

A.1 NOTATION

We index identities by i, animation frames by $\ell \in \{0, \dots, L-1\}$, and diffusion steps by t. Table 5 summarizes the main variables, their dimensions, and their roles.

Table 5: Key notation used in the Method and Appendix.

Symbol	Dimension	Role
$\mathbf{X}_i^0$	$N_i \times 3$	input mesh vertices
$\mathbf{\Phi}_i$	$N_i \times K$	native motion basis used for decoding
$\mathbf{\Psi}_i$	$N_i \times K$	aligned spectral basis
$\mathbf{D}_i$	$K \times K$	shared-to-native identity adapter
$\mathbf{\Pi}_{ij}$	$N_j \times N_i$	full mesh vertex map from i to j
$\mathbf{C}_{ij}$	$K \times K$	functional map from i to j
$\mathbf{B}_{i,\ell}$	$K \times 3$	native displacement coefficients
$\mathbf{Z}_i$	$L \times K \times 3$	shared displacement trajectory
$\mathbf{Y}_i$	$L \times K \times 3$	normalized shared trajectory
$\mathbf{\Sigma}$	$K \times 3$	fixed coefficient scales
$\mathbf{S}_i^{\text{geo}}$	$K \times 3$	shared coefficients of the input geometry
$\mathbf{S}_i$	$K \times 5$	morphology tokens

A.2 NATIVE SPECTRAL MOTION COORDINATES

Relation to mass projection. When $\boldsymbol{q}_i = \mathbf{1}$ and $\eta_{\text{ridge}} = 0$, mass orthonormality reduces Eq. 7 to

$$\mathbf{B}_{i,\ell} = \mathbf{\Phi}_i^{\mathsf{T}} \mathbf{M}_i \Delta \mathbf{X}_{i,\ell}, \qquad \Delta \mathbf{X}_{i,\ell}^{(K)} = \mathbf{\Phi}_i \mathbf{B}_{i,\ell}. \tag{14}$$

A.3 CROSS-IDENTITY SPECTRAL ALIGNMENT

Descriptors and commonness. The DiffusionNet matcher (Sharp et al., 2022) encodes vertex positions, normals, heat kernel signatures (HKS), and wave kernel signatures (WKS) into unit-normalized descriptors $\boldsymbol{d}_i(v)$ and commonness scores $q_i(v)$.

Partial surface correspondences. To allow regions without reliable counterparts to remain unmatched, we use unbalanced Sinkhorn transport (Chizat et al., 2018) with a dustbin on each mesh. The cost of matching source vertex u to target vertex v is $1 - \boldsymbol{d}_j(v)^{\mathsf{T}} \boldsymbol{d}_i(u)$. Source masses are $m_i(u) q_i(u) / \sum_{u'} m_i(u')$, with target masses defined analogously and sums over the vertices used for matching. Vertex areas account for uneven sampling; commonness reduces the mass assigned to unreliable matches. The remaining mass goes to the dustbins. After removing them, row normalization gives $\mathbf{\Pi}_{ij}$, with target vertices indexing rows.

Functional maps for alignment. We express $\mathbf{\Pi}_{ij} \mathbf{\Phi}_i$ in the target native basis by solving Eq. 5, giving $\mathbf{C}_{ij} \in \mathbb{R}^{K \times K}$. Matcher training uses sampled vertices and corresponding basis rows and weights; final maps use all vertices. Swapping source and target gives $\mathbf{C}_{ji}$.

Confidence weights. We give more consistent map pairs greater influence on synchronization. For the K-dimensional maps, the round-trip error and confidence weight are

$$r_{ij}^{\text{map}} = \frac{\|\mathbf{C}_{ji}\mathbf{C}_{ij} - \mathbf{I}_K\|_F}{\sqrt{K}}, \qquad \omega_{ij} = \text{clip}_{[0.1,1]}\left(\frac{0.1}{\max(r_{ij}^{\text{map}}, 10^{-8})}\right). \tag{15}$$

Smaller round-trip errors give larger weights. We compute these weights from the frozen maps and keep them fixed during adapter synchronization.

Matcher training objective. We combine correspondence supervision, map consistency, and commonness regularization:

$$\begin{aligned}\mathcal{L}_{\text{match}} = \mathcal{L}_{\text{intra}} + \omega_{ij}^{\text{train}}(\mathcal{L}_{\text{descOT}} + 0.5\mathcal{L}_{\text{map-cyc}} + 0.1\mathcal{L}_{\text{orth}} + 0.1\mathcal{L}_{\text{commute}}) \\ + 0.05\mathcal{L}_{q-\text{smooth}} + 0.1\mathcal{L}_{q-\text{mass}}.\end{aligned} \tag{16}$$

$\mathcal{L}_{\text{intra}}$ supervises known matches within an identity and is zero across identities; $\mathcal{L}_{\text{descOT}}$ encourages descriptor agreement at matched vertices. Cycle consistency penalizes round-trip deviations from $\mathbf{I}_K$; orthogonality penalizes $\mathbf{C}^\mathsf{T}\mathbf{C} - \mathbf{I}_K$ and $\mathbf{C}\mathbf{C}^\mathsf{T} - \mathbf{I}_K$ in both directions. Commutativity discourages mixing spectral frequencies. Commonness regularization encourages similar neighboring scores and penalizes area-weighted means outside $[0.55, 0.90]$.

For pairs across identities, $\omega_{ij}^{\text{train}}$ transitions from one to the confidence in Eq. 15, evaluated on the current maps. Same-identity pairs retain unit weight. After training, the matcher and synchronization weights remain fixed.

Joint adapter synchronization. We use a fixed, connected graph of training identities, with selected neighbors per mesh and both map directions included in $\mathcal{E}_{\text{id}}$. The normalized residual in Eq. 6 is

$$r_{ij} = \frac{\|\mathbf{C}_{ij}\mathbf{D}_i - \mathbf{D}_j\|_F}{\sqrt{\max(\|\mathbf{C}_{ij}\mathbf{D}_i\|_F, 10^{-12})\max(\|\mathbf{D}_j\|_F, 10^{-12})}}. \tag{17}$$

The Huber penalty is $\rho = \frac{1}{2}r_{ij}^2$ for $r_{ij} \le 0.25$ and $\rho = 0.25(r_{ij} - 0.125)$ otherwise. Initialization takes one $K \times K$ block per identity from the K smallest eigenvectors of the block matrix for $\sum_{(i,j)\in\mathcal{E}_{\text{id}}} \omega_{ij}\|\mathbf{C}_{ij}\mathbf{D}_i - \mathbf{D}_j\|_F^2$. We then use projected gradient descent with backtracking, clipping singular values to $[0.2, 2.0]$ at initialization and after each update. The matcher, maps, and confidence weights remain fixed.

Shared motion targets. Substituting $\mathbf{B} = \mathbf{D}_i\mathbf{Z}$ into Eq. 7 gives

$$\mathbf{Z}_{i,\ell} = \arg\min_{\mathbf{Z}\in\mathbb{R}^{K\times 3}} \left\{ \left\| \mathbf{W}_i^{1/2}(\mathbf{\Psi}_i\mathbf{Z} - \Delta\mathbf{X}_{i,\ell}) \right\|_F^2 + \eta_{\text{ridge}}\|\mathbf{D}_i\mathbf{Z}\|_F^2 \right\}. \tag{18}$$

The regularizer acts on native displacement coefficients $\mathbf{D}_i\mathbf{Z}$ after conversion to shared coordinates.

Connecting a new identity. The frozen matcher computes maps between the new mesh i and training anchors. Each anchor j provides an estimate $\mathbf{C}_{ji}\mathbf{D}_j$, with confidence ω_{ji} computed from both map directions. We initialize $\mathbf{D}_i$ by their confidence-weighted mean, then recompute it with weights $\omega_{ji}\min(1, 0.25/\max(r_{ji}, 10^{-12}))$. Singular values are clipped to $[0.2, 2.0]$ at every step. Updates stop after 100 iterations or when the Frobenius change divided by $\sqrt{K}$ is at most 10^{-8}. Only $\mathbf{D}_i$ is updated; training adapters remain fixed, preserving the generator's shared coordinates.

A.4 Morphology-Conditioned Motion Diffusion

Coefficient normalization. We stack shared displacement coefficients into $\mathbf{Z}_i \in \mathbb{R}^{L\times K\times 3}$ and balance their scales across modes and coordinate channels before diffusion:

$$\mathbf{Y}_i = \mathbf{Z}_i \oslash \mathbf{\Sigma}, \qquad \mathbf{Z}_i = \mathbf{\Sigma} \odot \mathbf{Y}_i, \qquad \mathbf{\Sigma} \in \mathbb{R}_{>0}^{K\times 3}. \tag{19}$$

The same scales apply at every frame. They are estimated from training data and remain fixed thereafter.

Morphology tokens. The first three token entries are the geometry coefficients in Eq. 8. The remaining entries follow from

$$\mathbf{\Psi}_i^\mathsf{T}\mathbf{L}_i\mathbf{\Psi}_i = \mathbf{D}_i^\mathsf{T}\mathbf{\Lambda}_i\mathbf{D}_i, \qquad \mathbf{\Psi}_i^\mathsf{T}\mathbf{M}_i\mathbf{\Psi}_i = \mathbf{D}_i^\mathsf{T}\mathbf{D}_i. \tag{20}$$

Their diagonals give the Dirichlet energies and squared area-weighted norms of the aligned basis functions. The energies need not be Laplacian eigenvalues because alignment mixes native eigenfunctions.

Conditioning and denoiser architecture. The action condition combines a learned action embedding with a projected frozen CLIP embedding (Radford et al., 2021). Projected morphology tokens share mode embeddings with motion tokens, each containing three coefficient channels at a frame–mode pair. Blocks apply attention across modes and frames, cross attention to conditions, and feed-forward layers, using pre-normalization and residual connections.

Diffusion target and sampling. At diffusion step t, the noisy normalized trajectory and v target are

$$\mathbf{Y}_i^{[t]} = \sqrt{\bar{\alpha}_t}\,\mathbf{Y}_i + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_i, \qquad \mathbf{V}_i^{[t]} = \sqrt{\bar{\alpha}_t}\,\boldsymbol{\epsilon}_i - \sqrt{1-\bar{\alpha}_t}\,\mathbf{Y}_i. \tag{21}$$

Here $\bar{\alpha}_t$ is the cumulative signal power and $\boldsymbol{\epsilon}_i$ is standard Gaussian noise on noninitial frames. Sampling starts from Gaussian noise and uses 50 DDIM steps (Song et al., 2020). Initial-frame coefficients remain zero during training and sampling.

A.4.1 Complete Surface-Aware Objective

Denoising and surface losses. The denoising loss averages squared v-prediction errors over coefficient entries. The surface loss in Eq. 11 can be evaluated in shared coordinates:

$$\mathbf{G}_i^{\text{shared}} = \boldsymbol{\Psi}_i^{\mathsf{T}} \mathbf{W}_i \boldsymbol{\Psi}_i, \qquad \mathcal{L}_{\text{surf}} = \operatorname{tr}\left[(\hat{\mathbf{Z}} - \mathbf{Z})^{\mathsf{T}} \mathbf{G}_i^{\text{shared}} (\hat{\mathbf{Z}} - \mathbf{Z}) \right]. \tag{22}$$

Temporal and amplitude losses. $\mathcal{L}_{\text{vel}}$ and $\mathcal{L}_{\text{acc}}$ apply the same surface metric to first- and second-difference errors, using unit frame spacing. Each window receives its smallest positive frame weight, or zero if none is positive. Let $A_i(\mathbf{Z}_i)$ be the square root of the mean of $\operatorname{tr}(\mathbf{Z}_{i,\ell}^{\mathsf{T}} \mathbf{G}_i^{\text{shared}} \mathbf{Z}_{i,\ell})$ over frames. The amplitude loss averages

$$\text{SmoothL1}_{\beta=0.1}\left(\frac{1}{2} \log \frac{A_i(\hat{\mathbf{Z}}_i)^2 + 10^{-12}}{A_i(\mathbf{Z}_i)^2 + 10^{-12}},\ 0 \right) \tag{23}$$

over sequences. The complete objective is

$$\mathcal{L} = \mathcal{L}_{\text{vpred}} + 0.10\mathcal{L}_{\text{surf}} + 0.05\mathcal{L}_{\text{vel}} + 0.02\mathcal{L}_{\text{acc}} + 0.10\mathcal{L}_{\text{amp}}. \tag{24}$$

Geometric refinement is applied after generation.

B Implementation Details

We use the same representation, denoiser architecture, training objective, and sampling procedure for Animal and Human, with separately trained matchers, coordinate systems, and diffusion models. The ablation variants are described in Appendix D.

B.1 Coordinate conventions.

Mesh normalization. For each input mesh, we compute a robust length scale as the area-weighted 95th percentile of vertex distances from its surface centroid. This scale and the body orientation remain fixed throughout the sequence. Animal sequences are centered at their area-weighted surface centroid in each frame before projection, giving root-relative motion. Human sequences instead use the input-mesh centroid for every frame, preserving global translation in the displacement trajectory. These conventions apply to both training targets and evaluation references.

B.2 Training and Inference

Matcher training. We train for 40 epochs with AdamW (Loshchilov & Hutter, 2017), learning rate 10^{-4}, weight decay 10^{-4}, and gradient clipping at 1.0. The matcher is frozen during inference and new identities contribute only input geometry to correspondence and adapter estimation.

Diffusion training. The denoiser uses eight blocks of width 128, with four attention heads and a feed-forward expansion ratio of four. Animal and Human models train for 80,000 and 35,000 optimizer steps with global batch sizes of 64. We train with AdamW using a learning rate of 2×10^{-4}, weight decay 0.01, and 2000 warm-up steps followed by cosine learning-rate decay. Training runs on NVIDIA RTX A6000 GPUs; separate runtime measurements use an RTX 3090 (Appendix C.4).

BiMotion fine-tuning. We fine-tune BiMotion using its official interface on our training splits for 80,000 Animal steps and 35,000 Human steps, with global batch sizes of 64, and a learning rate of 2×10^{-4}.

C Additional Results

C.1 Additional Comparisons

Fig. 5 extends the main comparisons to additional human and animal cases, using the same input mesh and action for all methods. Selected frames show action realization and local deformation.



Figure 5: **Additional comparisons of sequence generation**. Each column group specifies one input mesh and action, with method names on the left and four selected frames progressing from left to right. All these input meshes are generated by image-to-3D method.

C.2 Additional SpectralWeave Generations

Fig. 6 shows two requested actions for each human or animal input mesh. The examples illustrate different motions on the same geometry while retaining its vertex indexing and connectivity.

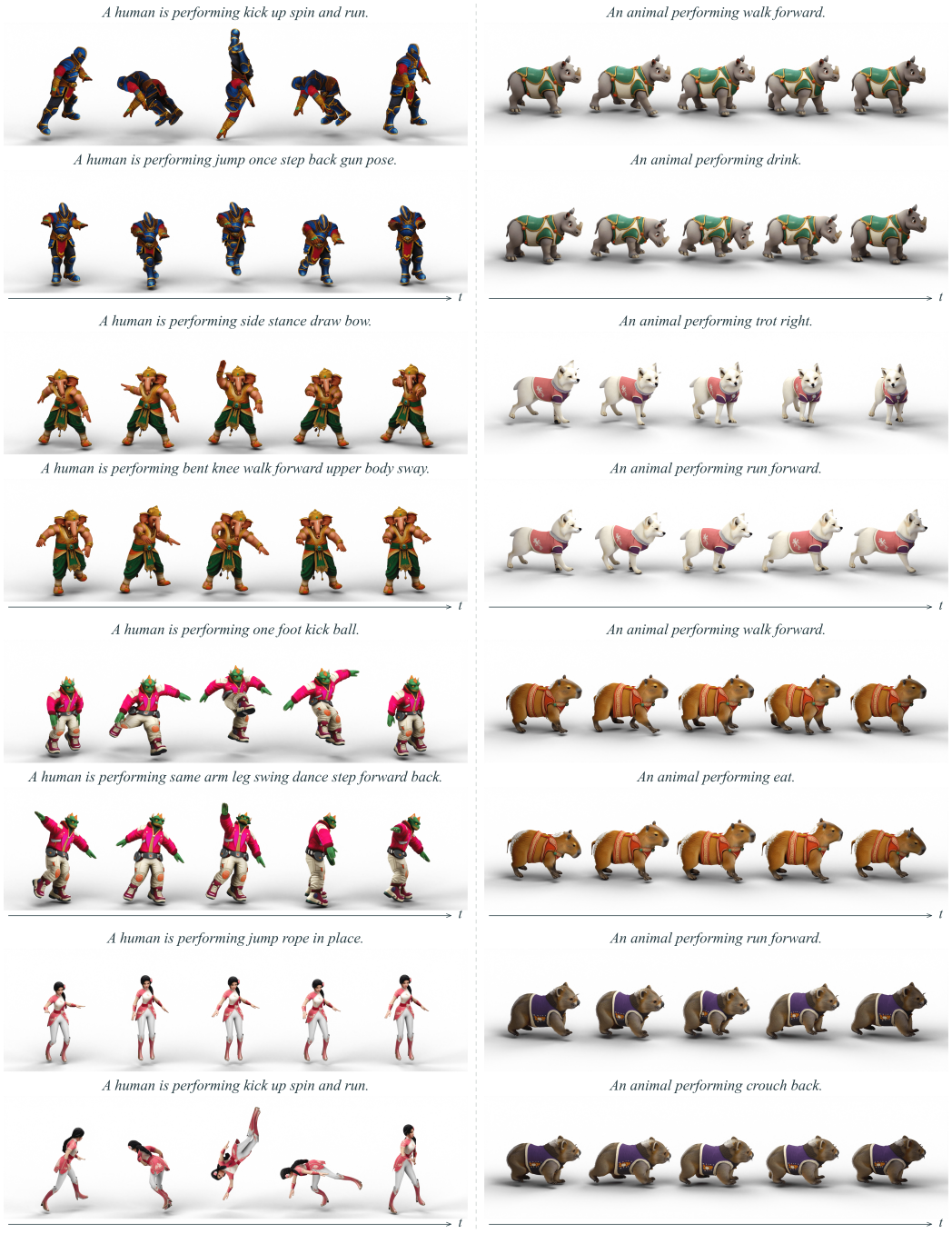


Figure 6: **Additional SpectralWeave generations**. Each pair of rows uses the same input mesh with two requested actions. Five selected frames progress from left to right, with the action description above each sequence.

C.3 MOTION DIVERSITY

Fig. 7 shows different realizations of running and attacking on the same input meshes, with variations in body pose and limb movement.



Figure 7: **Motion diversity.** Each row shows generated sequence, with its action description below. Examples are grouped by input mesh and action.

C.4 EFFICIENCY

Table 6 reports runtimes on external cases. We use one RTX 3090, batch size one, one warm-up, and three repetitions per case. Timing includes generation and refinement, excluding loading, preprocessing, and visualization. Direct-baseline timers exclude loading; video-mediated timers include additional initialization and native export. Sequence lengths and timer scopes differ, so these measurements are not a controlled speedup comparison.

Table 6: Runtime on 12 external cases. D and V denote direct and video-mediated generation. SpectralWeave includes geometric refinement. Sequence lengths and timer scopes differ across methods.

Method	Track	Frames	Gen. params. (M)	Time (s)↓
TapMo	D	120	27.03	42.31
AnimateAnyMesh	D	16	114.73	3.71
BiMotion	D	46	139.35	4.25
Animate3D	V	16	1609.86	1531.35
ActionMesh	V	16	3315.14	532.02
SpectralWeave	D	45	6.85	6.46

D EXPERIMENTS

Data splits. We first partition identities into a development set and a disjoint New Identity test set, using 40/10 identities for Animal and 80/20 for Human. Within the development set, complete identity–action sequences are assigned to training, validation, and Held-out Identity–Action testing using an action-stratified split. Every development identity retains training sequences, and all validation and test action labels occur in training. The validation and held-out test sets cover 25/26 Animal identities and 65/66 Human identities, respectively. These identity sets may overlap, but their identity–action combinations and complete sequences are mutually disjoint. Validation is used for checkpoint selection, whereas both test sets are reserved for final evaluation. New Identity meshes are excluded from matcher and motion-model training; only their static geometry is provided at inference. Exact identity and frame counts are reported in Table 7. For Human, the same source animation clips are retargeted across characters and reused across splits.

Evaluation protocol. Vertex trajectory metrics compare predictions and references on the input mesh's fixed vertex indexing. All methods are evaluated in the same coordinate frame, with orientation and scale fixed from the input mesh.

ActionMesh configuration. We evaluate ActionMesh with its official pretrained weights, conditioned jointly on the input mesh and a generated driving video. Following ActionMesh's 3D-and-text-to-animation pipeline, we generate the driving video from an input-mesh rendering and the action description, using Stable Diffusion 1.5 with AnimateDiff and IP-Adapter as the video frontend. Each video contains 16 frames at 512×512 resolution, with its first frame replaced by the input-mesh rendering. No ground-truth motion video is used. ActionMesh predicts per-vertex displacements on the input mesh in its second stage, preserving the input vertex correspondence and connectivity in the exported animation.

Distinction with SMD. SMD (Xue et al., 2025) generates shape-conditioned spectral motion on a shared SMPL topology. Its original formulation therefore does not address incompatible native coordinates across independently meshed inputs. SpectralWeave establishes shared displacement coordinates while retaining each input mesh's native basis for reconstruction, supporting different vertex counts and connectivity without a common template. We reimplement its spectral mesh formulation and adapt it to the Human benchmark by using the lowest-frequency $1{,}024$ eigenvectors of each identity's own combinatorial graph Laplacian. Each eigenvector's largest-magnitude entry is fixed to be positive; no spectral alignment is performed across identities. The model represents local vertex positions by spectral coefficients and separately predicts global axis-angle rotation and translation. Training rigid motions are estimated against each character's own rest mesh. This adaptation uses no SpectralWeave matcher, commonness weighting, adapters, or geometric refinement. SpectralWeave instead learns in shared displacement coordinates and decodes through each mesh's native basis. Table 8 compares our full pipeline with this adapted SMD implementation, which has higher V-RMSE, CD-L2, and HD95 and lower motion cosine similarity.

Table 7: Dataset split statistics. Frame counts are approximate; k denotes one thousand frames. Identity counts indicate the identities present in each split and are not additive across rows. Held-out Identity–Action evaluates combinations absent from training; New Identity contains entirely held-out identities.

Dataset	Split	Identities	Frames
Animal	Training	40	33k
	Validation	25	3k
	Held-out Identity–Action	26	3k
	New Identity	10	8k
Human	Training	80	145k
	Validation	65	16k
	Held-out Identity–Action	66	16k
	New Identity	20	32k

SMD configuration. SMD implementation uses spectral convolutions and learned pooling for encoding and decoding, with an eight-layer Transformer of width 256, eight attention heads, and feed-forward width 1,024. The contrastively pretrained shape encoder is frozen during diffusion training and supplies shape conditioning through FiLM. Shape pretraining and diffusion training use 10,000 and 50,000 updates, respectively, with effective batch size 32 and Adam at learning rate 10^{-4}. Diffusion uses x_0 prediction and a 2,000-step cosine DDPM schedule.

Table 8: **Comparison with SMD.**

Method	V-RMSE↓	Cos.↑	CD-L2↓	HD95↓
SMD Xue et al. (2025)	21.513	53.185	6.822	13.773
SpectralWeave	**5.330**	**94.490**	**2.470**	**5.190**

D.1 Ablation Configurations

Ablations use the main model training budgets and the same evaluation protocol. Each changed spectral representation is used to recompute training coefficients, morphology tokens, and normalization statistics before retraining.

Comparison with existing spectral representations. CQHB (Kovnatsky et al., 2013) constructs compatible bases for correspondence and pose transfer, while FMN (Huang et al., 2014) and Consistent ZoomOut (CZO) (Huang et al., 2020) establish consistent functional maps across shape collections. Building on this line of work, SpectralWeave constructs shared displacement coordinates for learning motion across independently meshed identities and decoding it on each input mesh. We retain each mesh's native Laplacian basis and jointly estimate invertible adapters from fixed partial functional maps. The partial maps encode matching constraints on reliably corresponding regions, while the adapters define the shared-to-native coordinate transformations used for generation. Bounding the adapter singular values maintains invertibility and controls coefficient scaling while permitting non-orthogonal transformations. This construction accommodates partial correspondences without requiring the correspondence maps themselves to be invertible, and retains each mesh's selected native reconstruction subspace.

We evaluate this coordinate construction against CQHB, FMN, and CZO adapted to the same motion-generation task. For each representation, we recompute motion targets, morphology tokens, and training-set normalization statistics, then retrain the same diffusion architecture with matched training budgets. Sampling and geometric refinement remain unchanged. Under this controlled setting, SpectralWeave achieves the best results on all four reported metrics in both datasets (Table 9), providing empirical support for the proposed representation in cross-identity motion generation.

CZO adaptation. We initialize functional maps from normalized rest-coordinate matches supplemented by HKS/WKS descriptors, without our learned matcher or commonness weighting. CZO refines these maps over a symmetric six-neighbor graph augmented by a minimum spanning tree. We extract 32 consistent latent directions and multiply each native Laplacian basis by the corresponding

CZO coefficient matrix to construct its decoding basis. Observed displacements are encoded by mass-weighted least squares. Generated coefficients are denormalized and decoded through the same basis, then added to the input vertex positions.

For a new identity, we select five training anchors using normalized log-spectral descriptors and keep their latent bases fixed. We refine the bidirectional maps to these anchors and estimate the new identity's latent basis coefficients by least squares using both map directions. This constructs its decoder in the fixed training coordinate system.

Table 9: **Comparison of spectral representations adapted to motion generation on unseen test identities.** All variants use the same generator architecture and evaluation protocol with full refinement. Bold indicates the best result within each dataset.

Representation	V-RMSE↓	Cos.↑	CD-L2↓	HD95↓
Animal New-ID test dataset				
CQHB Kovnatsky et al. (2013)	8.759	61.425	3.949	8.721
FMN Huang et al. (2014)	8.981	60.906	4.953	9.463
CZO Huang et al. (2020)	19.688	14.147	8.273	16.790
SpectralWeave (Ours)	**7.580**	**67.360**	**3.550**	**7.530**
Human New-ID test dataset				
CQHB Kovnatsky et al. (2013)	6.247	92.275	2.969	5.611
FMN Huang et al. (2014)	6.509	92.080	2.706	5.645
CZO Huang et al. (2020)	27.645	30.461	12.521	26.793
SpectralWeave (Ours)	**5.330**	**94.490**	**2.470**	**5.190**

Spectral alignment. In Table 3, *w/o adapter* sets $\mathbf{D}_i = \mathbf{I}$ while retaining native bases and commonness weights. *W/o spectral alignment* also removes the matcher and commonness, recomputes bases without sign fixing, and uses area-only projection and supervision. *Single-reference alignment* fixes one training mesh's adapter from joint synchronization and estimates the others using only bidirectional constraints with that reference. *Orthogonal synchronization* retains the graph and maps but imposes $\mathbf{D}_i^{\mathsf{T}}\mathbf{D}_i = \mathbf{I}$.

Table 10: Partial versus forced full correspondence on unseen test identities. Low-q is the same region for both variants, covering the 20% of rest-surface area with the lowest commonness scores. Bold indicates the best result within each dataset.

Setting	V-RMSE↓	CD-L2↓	HD95↓	Low-q V-RMSE↓
Animal test dataset				
Forced full correspondence	8.3581	3.8940	8.2981	7.3905
Partial correspondence (Ours)	**7.580**	**3.550**	**7.530**	**6.7470**
Human test dataset				
Forced full correspondence	5.5635	2.5858	5.3864	4.5053
Partial correspondence (Ours)	**5.330**	**2.470**	**5.190**	**4.3329**

Table 11: Morphology ablation on new identities. "No morphology" removes the shape condition. All variants use full refinement, with a fixed recipe within each domain.

Domain	Variant	V-RMSE↓	CD-L2↓	HD95↓
Animal	Full	**7.580**	**3.550**	**7.530**
	No morphology	9.865	4.422	9.488
Human	Full	**5.330**	**2.470**	**5.190**
	No morphology	7.886	3.519	7.675

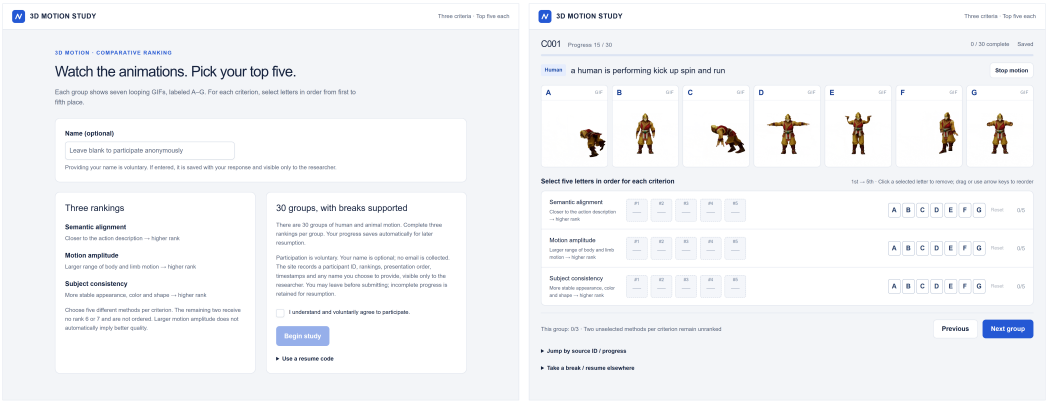
Table 12: Shared spectral dimension on unseen-identity test sets. All settings use full refinement, with a fixed recipe within each domain, against the same GT.

	Animal		Human	
Modes	CD-L2↓	HD95↓	CD-L2↓	HD95↓
$K = 16$	3.860	8.240	2.691	5.727
$K = 32$	**3.550**	**7.530**	**2.470**	**5.190**
$K = 64$	4.004	8.534	2.859	6.009
$K = 128$	4.334	9.305	3.213	6.872

Morphology and spectral dimension. The morphology ablation (Table 11) removes morphology conditioning while retaining the action condition and motion targets. For the spectral dimension ablation (Table 12), each model trains at its respective dimension and is evaluated against the same

ground-truth motions on unseen test identities from both datasets. The default $K = 32$, fixed before examining the test results, gives the lowest CD-L2 and HD95 among the tested dimensions; increasing K to 64 or 128 raises both errors.

D.2 User Study



(a) Study instructions (b) Ranking interface

Figure 8: **User study interface.** (a) Study instructions. (b) Seven anonymized animations and controls for ranking five methods under each of the three criteria.

Table 13: User study: percentage of rankings placing each method first. Action alignment assesses agreement with the prompt; subject consistency assesses appearance and shape stability. Motion amplitude ranks movement extent, with larger motion ranked first. Bold marks the largest share in each column. A dash "–" indicates that the method received no first-place rankings for that criterion.

Method	Action alignment (%)	Motion amplitude (%)	Subject consistency (%)
Animate3D	1.5	0.5	8.3
AnimateAnyMesh	2.0	1.5	5.0
TapMo	5.8	17.7	8.3
BiMotion	–	–	25.7
BiMotion Finetune	14.3	28.7	7.3
ActionMesh	1.0	7.8	2.0
SpectralWeave (Ours)	**75.3**	**43.8**	**43.3**

Study design. Twenty participants evaluate 30 cases (20 human and 10 animal), each showing an action description and seven methods under anonymous labels A–G. Animations play synchronously; case order and method positions are randomly permuted for each participant (Fig. 8).

Ranking and scores. Participants independently rank their top five methods for action alignment, motion amplitude, and subject consistency; the other two remain unranked. These criteria assess agreement with the description, extent of movement, and stability of appearance and shape, respectively. Instructions clarify that larger amplitude does not imply higher overall quality. Each participant completes 90 rankings; median completion time is approximately 95 minutes. Table 13 reports first-place selection percentages from 600 rankings per criterion.