HoAM-4D: Holistic Aware Mapping for 4D Mesh Animation and Reconstruction

Anonymous authors
Paper under double-blind review

Abstract

While generative 3D modeling has advanced rapidly, lifting static 3D assets and monocular videos into coherent 4D mesh sequences remains challenging, as end-to-end feed-forward frameworks often lack explicit video–mesh alignment, while view-dependent optimization methods struggle to model authentic 4D motion, resulting in inconsistent image observations or disordered trajectories. To address these challenges, we present **HoAM-4D**, a multi-stage optimization framework that introduces holistic vertex awareness through 4D kinematic flow and video-to-mesh mapping for coherent 4D mesh generation. Specifically, to improve 4D motion modeling, HoAM-4D projects both the 3D mesh and video observations into 3D space and predicts geometry-motion and image-motion displacement fields. To address video–mesh misalignment, it further predicts image-to-canonical-mesh and image-to-synchronous-mesh correspondence fields. Beyond visible regions, we extend the video-to-mesh mapping to occluded areas through skeleton-based conduction. Finally, we leverage both the input video and 4D motion as supervision to optimize holistic-aware mesh deformation from the first-frame canonical mesh, enforcing articulated structural, temporal, and appearance consistency. Extensive experiments demonstrate that **HoAM-4D** outperforms existing methods in geometric accuracy, motion coherence, and video-aligned rendering quality.

1 Introduction

High-fidelity mesh animation Sabathier et al. (2026); Chen et al. (2025) stands as a cornerstone of 3D computer vision and graphics, with widespread applications such as digital humans Chai et al. (2024), virtual reality Zhen et al. (2025), and cinematic visual effects Xu et al. (2025). By representing dynamic scenes through structured polygonal meshes, this technique can achieve efficient rendering, seamless integration with physics engines, and intuitive artistic control. Reconstructing such animations from a monocular video nevertheless remains challenging, since a single view provides incomplete 3D evidence. Meshes generated independently per frame differ in scale, topology, and vertex ordering, and the image features that carry the motion signal live in a different space from those meshes, with no explicit correspondence between the two.

In this context, current work in 4D generation primarily follows two paradigms: *optimization-based* and *generation-based* methods, as illustrated in Figure 1. Optimization-based methods, such as DreamMesh4D Li et al. (2024) and V2M4 Chen et al. (2025), use rendering losses to iteratively refine dynamic 3D representations. Although this provides strong appearance supervision, the signal is view dependent and does not explicitly establish image-to-geometry correspondence in occluded or ambiguous regions. Generation-based approaches instead predict complete motion in a feed-forward manner. For example, Mesh4D Jiang et al. (2026) uses skeletal motion to deform a base mesh, while ActionMesh Sabathier et al. (2026) predicts sequence-level vertex deviations. These methods provide stronger motion regularity, but typically compress visual guidance into a global condition rather than modeling image-space motion and its relation to canonical and instantaneous 3D geometry.

Therefore, a successful system is therefore supposed to infer coherent motion from both modalities while establishing point wise connection between modalities for holistic aware supervision. However, motion and video-to-mesh mapping are heterogeneous targets, one a 3D vector field and the other a discrete assignment, and are therefore learned in different spaces with different losses. Our key observation is that this split is unnecessary. If the video is lifted into the same volume as the geometry,

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

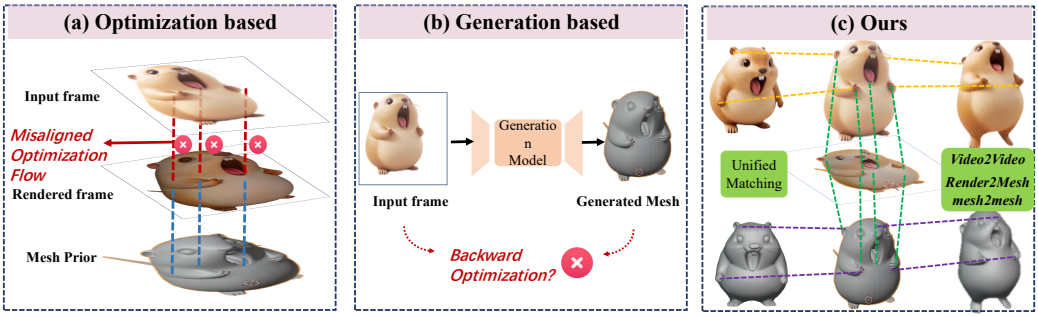


Figure 1: **Comparison of 4D reconstruction paradigms.** (a) Optimization-based methods depend on partial, view-dependent rendering supervision. (b) Generation-based methods produce coherent motion but do not explicitly recover image-to-geometry correspondence. (c) HoAM-4D predicts two temporal displacement fields and two cross-modal correspondence-offset fields, from which canonical and synchronous matching maps are derived geometrically.

a correspondence is nothing more than a displacement whose endpoint lands on the matched voxel, and both become homogeneous 3D quantities that a single regression objective can cover.

We realize this idea in **HoAM-4D**, a three-stage framework. Stage I reconstructs and rigs each frame to obtain a raw mesh sequence, a canonical skeleton with skinning weights, and a camera trajectory. Stage II lifts the video by augmenting image features with their grid coordinates to form sparse 3D image voxels, then feed this with 3D mesh voxels to a dual-branch sparse transformer with temporal and cross-modal attention to regress four displacement fields: geometry motion, image motion, and image-to-canonical and image-to-synchronous correspondence offsets. Matching maps follow from the offset endpoints instead of being classified separately, and a geometric cycle relating the two offsets to the geometry motion ties the four fields together.

Stage III optimizes the fixed-topology canonical mesh under both the predicted 4D motion and the input video. Since the canonical correspondence attaches video evidence only to vertices the camera can see, we further introduce a skeleton-based deviation conduction module. Rather than diffusing the deviation over the surface graph, which leaks across kinematically unrelated parts, it aggregates the visible deviation at each joint and redistributes it to the occluded vertices that the same joint drives, completing the supervision where direct observation is unavailable.

Through this design, HoAM-4D converts a monocular video into a topologically coherent 4D mesh sequence. Our main contributions are summarized as follows:

- We recast video-to-mesh correspondence as displacement prediction in a shared 3D volume, so that motion and matching become homogeneous quantities trained under a single regression objective and linked by a geometric cycle.
- We introduce a **Cross-Modal Motion Estimation Transformer** (Section 3.2) that lifts the video into the geometry volume as coordinate-augmented image voxels and jointly regresses all four fields, using temporal attention for the two motion heads and cross-modal attention for the two correspondence heads.
- We develop a **Skeleton-based Deviation Conduction** module (Section 3.3) that propagates video-derived supervision from visible to occluded regions, improving reconstruction completeness and temporal coherence.

2 Related Works

2.1 4D Reconstruction and Motion Estimation

Optimization-based 4D Reconstruction These methods optimize 4D latent representations or deformable meshes through rendering loss back-propagation. Dynamic representations, including Dynamic NeRF Jiang et al. (2023); Yao et al. (2025), 4D Gaussian Splatting Wang et al. (2024), and Deformable Meshes Chen et al. (2025); Sabathier et al. (2025), are typically supervised by multi-view

video and surface graph constraints. While achieving high-fidelity rendering and continuous motion flow, these approaches require significant test-time optimization and are prone to temporal geometry drift.

Generation-based 4D Reconstruction To improve efficiency, generative models produce 4D content in a single forward pass. ShapeGen4D Yenphraphai et al. (2025) introduces temporal attention to 3D generators for video-conditioned mesh prediction, while ActionMesh Sabathier et al. (2026) enforces a consistent mesh topology across sequences to enhance structural integrity. However, these methods often struggle with precise camera-view alignment and lack the motion fidelity seen in optimization-based pipelines.

2.2 3D Dense Matching

Geometric and Point-based Correspondence Early works primarily focused on local geometric features. Point-based methods, such as DCP Lang et al. (2021), utilize **DGCNN** to learn per-point embeddings via self- and cross-attention. To handle structural symmetries, SE-ORNet Deng et al. (2023) aligns orientations using a teacher-student framework. However, these approaches often treat points as isolated entities, neglecting the underlying surface topology and global graph relations.

Surface Map-based Methods To enforce topological coherence, surface-based methods map 3D manifolds to shared domains, such as meshes Aigerman et al. (2022); Lévy et al. (2023) or planar regions Aigerman et al. (2015). For example, SURFMNet Roufosse et al. (2019) extends functional maps (FMNet Litany et al. (2017)) to unsupervised settings through structural constraints. While robust for intra-category tasks, these methods struggle with the extreme geometric variations inherent in "any-to-any" cross-category matching.

Semantic and Multi-view Integration Driven by 2D foundation models like DINO Oquab et al. (2023) and Stable Diffusion Rombach et al. (2022), multi-view rendering approaches have gained prominence. Diff3F Dutt et al. (2024) aggregates 2D features across rendered views and back-projects them for dense 3D matching. However, these methods rely heavily on 2D visual cues and frequently ignore intrinsic 3D geometric information. Our model addresses this gap by integrating 3D semantic features with **positional embeddings**, ensuring both semantic alignment and local geometric continuity.

In contrast to both lines of work, HoAM-4D estimates temporal motion and cross-modal correspondence as homogeneous 3D displacement fields, so that video evidence and geometry motion are learned in a common coordinate system and combined during skeleton-constrained reconstruction.

3 Method

Problem formulation. Given a monocular video $\mathcal{I} = \{I_t\}_{t=1}^{T}$ of a single articulated object, we seek an animated mesh sequence $\{\mathcal{M}_t\}_{t=1}^{T}$ that shares the fixed connectivity of a canonical mesh $\mathcal{M}^c = (\mathcal{V}^c, \mathcal{F}^c)$, i.e. $\mathcal{M}_t = (\mathcal{V}_t, \mathcal{F}^c)$, together with a camera trajectory $\{\boldsymbol{\pi}_t\}_{t=1}^{T}$ such that rendering $\mathcal{M}_t$ from $\boldsymbol{\pi}_t$ reproduces I_t. The difficulty is that the available 3D evidence is a set of independently generated per-frame meshes with inconsistent topology and vertex ordering, while the image evidence lives in a different space and carries no explicit link to mesh vertices.

Overview. As illustrated in Figure 2, HoAM-4D first initializes frame-wise meshes, a canonical mesh, cameras, and skeletal rigs. It then represents the sequence using two complementary voxel streams: a geometry stream constructed from generated meshes and an image stream constructed by augmenting image features with 3D voxel coordinates. A dual-branch network predicts geometry and image temporal displacements. Two independent cross-modal heads predict image-to-canonical and image-to-synchronous correspondence offsets. Canonical and synchronous matching maps are derived from the endpoints of these offsets instead of being directly classified. The resulting four displacement fields are trained in a common coordinate system and jointly constrain the final skeleton-driven reconstruction.

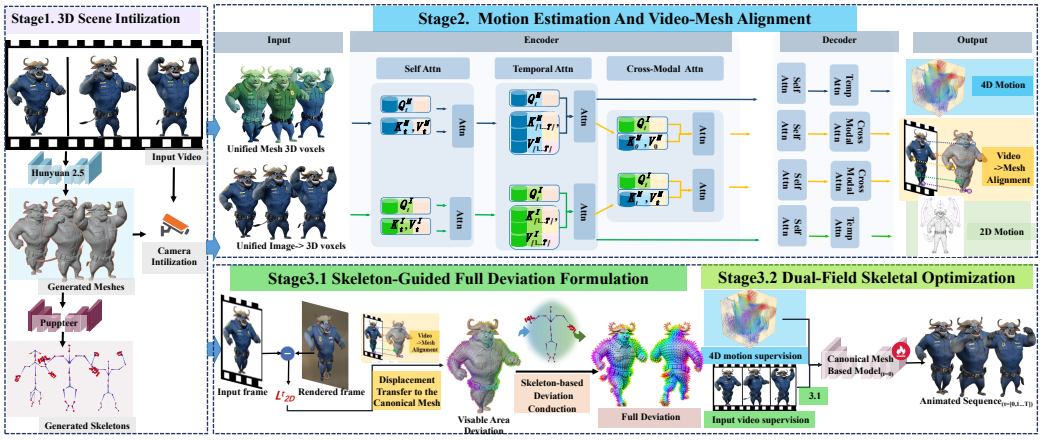


Figure 2: **Overview of HoAM-4D. (Stage I)** initializes the canonical mesh, frame-wise geometry, cameras, and skeletal rigs. **(Stage II)** encodes mesh and image voxels in two parallel branches of self- and temporal attention, then decodes four displacement fields: the two motion decoders use temporal attention, while the two correspondence decoders use cross-modal attention against first-frame and same-frame geometry, from which the matching maps are derived geometrically. **(Stage III.1)** transfers the fields to the canonical mesh and completes the visible deviation by skeleton-based conduction; **(Stage III.2)** optimizes the skeleton under both 4D motion and input video supervision.

3.1 Stage I: 3D Scene Initialization

Stage I reconstructs and rigs each frame with Hunyuan3D 2.5 and Puppeteer, selects the first-frame mesh as the fixed-topology canonical representation, normalizes the remaining observations to its volume, and estimates a temporally smooth camera trajectory using DreamSim and silhouette alignment; the complete formulation is provided in Appendix A. It produces the four quantities used throughout the rest of the paper:

$$\underbrace{\mathcal{I} = \{I_t\}_{t=1}^{T}}_{\text{video}}, \quad \underbrace{\{\widehat{\mathcal{M}}_t\}_{t=1}^{T}}_{\text{raw meshes}}, \quad \underbrace{\big(\mathcal{M}^c,\, \mathcal{J} = \{\mathbf{j}_k^c\}_{k=1}^{K},\, \mathbf{W} = [w_{ik}]\big)}_{\text{canonical mesh and rig}}, \quad \underbrace{\{\boldsymbol{\pi}_t\}_{t=1}^{T}}_{\text{cameras}}. \tag{1}$$

Here $\widehat{\mathcal{M}}_t$ are the volume-aligned per-frame meshes that supply geometry evidence, $\mathcal{M}^c = (\mathcal{V}^c, \mathcal{F}^c)$ is the first-frame canonical mesh with vertices $\mathbf{v}_i^c$, and $\mathcal{J}$ and $\mathbf{W}$ are its joints and skinning weights with $w_{ik} \geq 0$, $\sum_k w_{ik} = 1$. Stage II consumes $(\mathcal{I}, \{\widehat{\mathcal{M}}_t\})$, while Stage III optimizes $\mathcal{M}^c$ under $(\mathcal{J}, \mathbf{W}, \{\boldsymbol{\pi}_t\})$.

3.2 Stage II: Cross-Modal Motion Estimation Transformer

The key design of Stage II is to make motion and correspondence *the same kind of quantity*. Once video and geometry live in one 3D volume, an image-to-mesh match is just a displacement whose endpoint lands on the matched voxel, so all four outputs become 3D vectors trained by one regression objective. Stage II therefore tokenizes both modalities into that volume, couples them with cross-modal attention, and decodes four displacement fields.

Sparse 3D Tokenization. Both modalities are lifted into the common canonical volume, so that motion and correspondence become homogeneous 3D quantities. Each aligned mesh $\widehat{\mathcal{M}}_t$ is voxelized with only occupied cells retained, giving active coordinates $\mathbf{c}_{t,n}^G$ and TRELLIS geometry features $\mathbf{f}_t^{\text{geo}} = E_{\text{Trellis}}(\mathbf{V}_t^G)$. In parallel, the semantic features $\mathbf{f}_t^{\text{sem}} = [E_{\text{DINO}}(I_t), E_{\text{SD}}(I_t)]$ of frame I_t are assigned grid coordinates $\mathbf{c}_{t,m}^I$ and voxelized into the same space. Attaching each feature to its voxel gives two sparse token sets $\mathbf{X}_t^G$ and $\mathbf{X}_t^I$ that stay modality-specific yet share one coordinate frame:

$$\mathbf{x}_{t,n}^G = W_G\big[\mathbf{f}_{t,n}^{\text{sem}}, \mathbf{f}_{t,n}^{\text{geo}}, \mathbf{c}_{t,n}^G\big], \qquad \mathbf{x}_{t,m}^I = W_I\big[\mathbf{f}_{t,m}^{\text{sem}}, \mathbf{c}_{t,m}^I\big]. \tag{2}$$

Dual-Branch Spatiotemporal Encoder. The two branches $b \in \{G, I\}$ share one architecture with independent parameters. Each block first aggregates spatial structure within a frame, then gathers

motion context across the whole sequence by querying all frames $\mathbf{S}^b_{\text{all}} = \text{Concat}(\mathbf{S}^b_1, \ldots, \mathbf{S}^b_T)$:

$$\mathbf{S}^b_t = \mathbf{X}^b_t + \text{SelfAttn}_b\big(\text{LN}(\mathbf{X}^b_t)\big)\,, \tag{3}$$

$$\mathbf{H}^b_t = \text{Attn}\big(Q_b(\mathbf{S}^b_t), K_b(\mathbf{S}^b_{\text{all}}), V_b(\mathbf{S}^b_{\text{all}})\big) + \text{LN}(\mathbf{R}^b_t), \tag{4}$$

Here $\mathbf{R}^b_t$ is a raw motion field computed directly from the voxel features, before any learning: for each active voxel at frame t we take the cosine similarity between its feature and those of the candidate voxels at frame $t+1$, pick the arg-max index, and use the coordinate difference to that voxel as a coarse displacement. This nearest-feature match is noisy and occasionally inconsistent, so we do not treat it as a target; embedded and added after temporal attention, it serves as guidance that lets the branch refine an approximate trajectory rather than invent one from scratch. Its contribution is ablated in Table 4.

Cross-Modal Coupling. Instead of predicting dense image-to-mesh affinity matrices, we cast correspondence as 3D displacement prediction. The image tokens query geometry tokens twice: against the canonical frame to anchor identity, and against the same frame to resolve the instantaneous pose,

$$\mathbf{C}^r_t = \text{CrossAttn}_r\Big(Q = \mathbf{H}^I_t,\ K, V = \mathbf{H}^G_{\sigma(r,t)}\Big)\,, \quad \sigma(can, t) = 1,\ \ \sigma(sync, t) = t, \tag{5}$$

for $r \in \{can, sync\}$. The two resulting features are thus identity-aware and pose-aware respectively, and each feeds its own displacement head. Predicting a displacement rather than classifying a candidate is what allows the correspondence to be graded by Euclidean error and to enter the cross-field consistency terms below; Table 4 compares the two formulations directly.

Four Displacement Heads. All four decoders begin with frame-wise self-attention, but then differ according to what their target depends on. A temporal displacement is a property of a token across time, so the two motion decoders follow self-attention with temporal attention over the sequence. A correspondence offset is instead a relation between two modalities at a fixed pair of frames, so the two offset decoders follow self-attention with cross-modal attention against the corresponding geometry tokens. Each decoder ends with a linear layer regressing one 3D vector per token:

$$\widehat{\mathbf{D}}^b = \mathcal{D}_b(\mathbf{H}^b), \quad b \in \{G, I\}, \qquad \widehat{\mathbf{O}}^r = \mathcal{D}_r(\mathbf{C}^r), \quad r \in \{can, sync\}. \tag{6}$$

The outputs are the geometry and image temporal displacement fields and the image-to-canonical and image-to-synchronous correspondence offsets, all expressed as 3D vectors in the shared volume.

Unified Displacement Supervision. Because the four outputs are homogeneous, a single regression objective covers all of them. Writing $\mathbf{U}^h$ for the field of head $h \in \{G, I, can, sync\}$, i.e. $\mathbf{U}^{G,I} = \mathbf{D}^{G,I}$ and $\mathbf{U}^{can,sync} = \mathbf{O}^{can,sync}$,

$$\mathcal{L}_{disp} = \sum_h \left\| \widehat{\mathbf{U}}^h - \mathbf{U}^{h,*} \right\|_1 . \tag{7}$$

The targets come from the paired sequence: $\mathbf{D}^{G,*}$ from mesh trajectories, $\mathbf{O}^{can,*}$ and $\mathbf{O}^{sync,*}$ from the coordinate differences between an image voxel and its matched geometry voxel in the canonical and current frames, and $\mathbf{D}^{I,*}$ from the geometry trajectory transferred through the synchronous relation, which avoids defining an incompatible image-space flow target.

Cross-Field Consistency. The four fields are geometrically related rather than independent, and we constrain them accordingly:

$$\mathcal{L}_{network} = \mathcal{L}_{disp} + \mathcal{L}_{map} + \mathcal{L}_{cyc} + \mathcal{L}_{mm} + \mathcal{L}_{temp}. \tag{8}$$

Here $\mathcal{L}_{map}$ is a KL term that aligns the endpoint-derived soft matching maps with the ground-truth correspondence distributions, $\mathcal{L}_{cyc}$ enforces the geometric cycle $\mathbf{O}^{sync} - \mathbf{O}^{can} = \mathbf{D}^G$ between the two correspondence offsets and the geometry motion, $\mathcal{L}_{mm}$ requires the image motion to agree with the geometry motion transferred through the synchronous relation, and $\mathcal{L}_{temp}$ penalizes second-order temporal variation of all displacement fields. Balancing weights are omitted for clarity; detailed derivations are given in Appendix C.

3.3 Stage III: Matching-Guided Skeleton-Constrained Optimization

Stage III turns the four voxel-level fields into a single articulated animation. Two of them are used directly: the geometry motion states where vertices should go, and the canonical correspondence carries the video evidence onto the mesh. The skeleton then serves twice, as the deformation model and as the channel that extends video supervision into occluded regions.

Displacement Transfer to the Canonical Mesh. Stage III animates $\mathcal{M}^c$ by linear blend skinning with the rig $(\mathcal{J}, \mathbf{W})$ of Equation 1, driven by per-joint rigid transforms $\mathbf{T}_{t,k} = (\mathbf{R}_{t,k}, \mathbf{t}_{t,k}) \in SE(3)$:

$$\mathbf{v}_{t,i} = \underbrace{\sum_{k=1}^{K} w_{ik} \left(\mathbf{R}_{t,k}\mathbf{v}_i^c + \mathbf{t}_{t,k}\right)}_{\text{skeletal}} + \underbrace{\mathbf{r}_{t,i}}_{\text{free}}, \tag{9}$$

where $\{\mathbf{T}_{t,k}\}$ carry the articulated motion and the per-vertex residual $\mathbf{r}_{t,i}$ is kept small, contributing about a tenth of the total displacement. The skeletal term guarantees kinematically valid, globally coherent deformation, while the residual lets the surface depart slightly from a purely rigid blend so that local shape and texture detail can follow the video. To keep the effective degrees of freedom close to those of the skeleton alone, we penalize $\|\mathbf{r}_{t,i}\|_2$ together with its Laplacian over the mesh and its second-order temporal difference, so the residual stays spatially smooth, temporally stable, and an order of magnitude below the skeletal displacement. Connectivity $\mathcal{F}^c$ is preserved by construction in either case.

Both fields reach these vertices by weighted averaging. The geometry field is interpolated with normalized trilinear weights $\eta_{i,n}$ over the canonical voxels adjacent to $\mathbf{v}_i^c$, giving $\widehat{\mathbf{D}}^G_{t,i} = \sum_n \eta_{i,n}\widehat{\mathbf{D}}^G_{t,n}$. The video constraint needs no extra machinery either, since the image branch already predicts $\widehat{\mathbf{D}}^I$ on image voxels in the same volume: composing the canonical matching map with the same weights gives the responsibility $a_{t,m\to i} = \sum_j \widehat{M}^{can}_{t,mj}\eta_{i,j}$ of image voxel m for vertex i, and hence the vertex-level deviation $\widehat{\mathbf{D}}^{I\to c}_{t,i} = \sum_m a_{t,m\to i}\widehat{\mathbf{D}}^I_{t,m} / \sum_m a_{t,m\to i}$. Image voxels are instantiated only from observed pixels, so self-occluded vertices accumulate no responsibility mass; let $\mathcal{V}_t^{vis}$ be those that do.

Skeleton-based Deviation Conduction. The deviation above covers $\mathcal{V}_t^{vis}$ only, leaving occluded regions unconstrained by the video. Rather than diffusing it over the surface graph, which ignores articulation and leaks across kinematically unrelated parts, we conduct it through the skeleton: vertices sharing a joint move under one rigid transform, so a deviation seen on a visible vertex is evidence about that joint and hence about every vertex it drives. Using $\mathbf{W}$ as the conduction kernel, joint k collects a correction $\mathbf{\Delta}_{t,k} = \sum_{i\in\mathcal{V}_t^{vis}} w_{ik}\widehat{\mathbf{D}}^{I\to c}_{t,i} / \sum_{i\in\mathcal{V}_t^{vis}} w_{ik}$ from its visible vertices and reports the observed fraction of its skinning mass, $\beta_{t,k} = \sum_{i\in\mathcal{V}_t^{vis}} w_{ik} / \sum_i w_{ik}$. A fully hidden joint has $\beta_{t,k} = 0$ and no direct evidence, so we blend it by coverage with its closest observed ancestor $\pi(k)$ on the kinematic tree and redistribute the result to occluded vertices:

$$\widetilde{\mathbf{\Delta}}_{t,k} = \beta_{t,k}\mathbf{\Delta}_{t,k} + (1-\beta_{t,k})\mathbf{\Delta}_{t,\pi(k)}, \qquad \widehat{\mathbf{D}}^{\text{full}}_{t,i} = \begin{cases} \widehat{\mathbf{D}}^{I\to c}_{t,i}, & i \in \mathcal{V}_t^{vis}, \\ \sum_k w_{ik}\widetilde{\mathbf{\Delta}}_{t,k}, & \text{otherwise.} \end{cases} \tag{10}$$

A well-observed joint thus keeps its own estimate while an occluded one inherits from the nearest observed joint. Since every step is a weighted sum with fixed weights, the conduction is differentiable and the video gradient reaches occluded vertices through the joints that drive them.

Dual-Field Skeletal Optimization. With the network frozen, we optimize the joint transforms and the residuals of Equation 9 against the two recovered fields:

$$\mathcal{L}_{opt} = \sum_{t,i} \left\| \mathbf{v}_{t,i} - (\mathbf{v}_i^c + \widehat{\mathbf{D}}^G_{t,i}) \right\|_1 + \sum_{t,i} \left\| \mathbf{v}_{t,i} - (\mathbf{v}_i^c + \widehat{\mathbf{D}}^{\text{full}}_{t,i}) \right\|_1. \tag{11}$$

The first term is the 4D motion loss, which holds the trajectory near the geometry motion predicted in Stage II; the second is the video-derived deviation loss, which pulls the surface toward the image evidence over the whole mesh rather than only its visible part. Roughly 90% of the resulting displacement is produced by skeletal propagation and the remaining 10% by the free residual, so the animation stays articulated and temporally stable while still admitting the small non-rigid changes needed to keep texture aligned with the video. Balancing weights are again omitted for clarity.

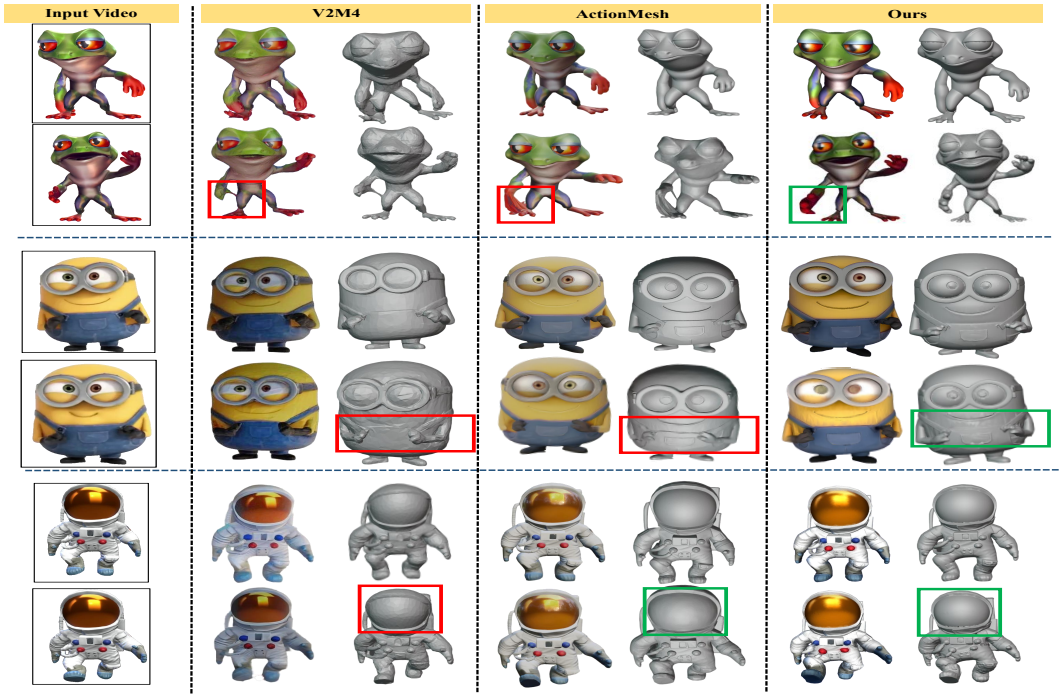


Figure 3: **Qualitative Evaluation of Different Methods.** We show both the color rendering and mesh geometry of the animated meshes. Complete animated sequences and multi-view comparisons are provided in the supplementary material.

4 Experiments

4.1 Experimental Settings

Datasets and supervision. We train on the 32-frame subset of DyMesh Wu et al. (2025), whose sequential dynamic meshes with fixed per-sequence topology and front-view renderings supply exactly the paired mesh–video evidence our four targets require, after filtering objects with negligible motion or severe distortion. For evaluation we deliberately move off the training distribution: 64 paired mesh–video sequences from ActionBench Sabathier et al. (2026) with large articulation and non-rigid deformation, plus 32 high-resolution video-only sequences including Consistent4D Jiang et al. (2023). Geometric metrics use the 64 paired sequences and rendering metrics all 96, so all numbers reflect cross-dataset generalization.

Baselines. We compare HoAM-4D with four state-of-the-art 4D generation methods: V2M4 Chen et al. (2025), ActionMesh Sabathier et al. (2026), LIM Sabathier et al. (2025), and DM4D Li et al. (2024). These methods have different geometric input requirements: DM4D and ActionMesh use a single textured mesh, whereas V2M4 uses a temporal sequence of textured meshes. We derive the required priors from the same input sequence and use the same first-frame reconstruction whenever a canonical mesh is required, so that the comparison focuses on temporal reconstruction rather than differences in initialization quality.

Evaluation Metrics We measure appearance with CLIP Radford et al. (2021), LPIPS Zhang et al. (2018), and FVD Unterthiner et al. (2018), and geometry and motion with CD_{3D}, CD_{4D}, and CD_M Sabathier et al. (2026). Two further metrics target what these miss: CD_{occ} restricts CD_{4D} to vertices invisible from the estimated camera, and Acc reports the percentage of image voxels whose predicted canonical endpoint lands within one voxel of the ground truth, measuring correspondence directly rather than through the mesh. Full definitions are in Appendix B.1.

Implementation Details. We use frozen Stable Diffusion and DINOv2 features for image voxels and TRELLIS features for geometry voxels. At a nominal resolution of 128^3, two temporal heads predict geometry and image displacements, while two independent cross-modal heads predict canonical

and synchronous correspondence offsets. All four heads use Smooth-L_1 displacement regression. We additionally supervise the endpoint-derived soft maps and impose geometric-cycle, matched-motion, and temporal-smoothness consistency. We first warm up the geometry temporal head, introduce the two correspondence heads and the image temporal head, and then jointly fine-tune the complete network. The displacement-to-matching derivation is given in Appendix C; architectural, optimization, and hardware details are reported in Appendix B.2.

4.2 Performance Comparisons

Table 1: Geometric and motion comparison with state-of-the-art methods. Lower is better for all metrics, and the best results are shown in bold.

Method	CD-3D ↓	CD-4D ↓	CD-M ↓
LIM Sabathier et al. (2025)	0.095	0.127	0.258
DM4D Li et al. (2024)	0.095	0.140	0.247
V2M4 Chen et al. (2025)	0.063	0.223	0.500
ActionMesh Sabathier et al. (2026)	0.050	0.069	0.137
Ours	**0.048**	**0.062**	**0.118**

Table 2: Video-aligned rendering comparison. ↑ indicates higher is better, and ↓ indicates lower is better.

Method	CLIP ↑	LPIPS ↓	FVD ↓
DM4D Li et al. (2024)	0.8291	0.0835	871.36
V2M4 Chen et al. (2025)	0.9006	0.0812	554.26
ActionMesh Sabathier et al. (2026)	0.9145	0.0766	503.49
Ours	**0.9312**	**0.0702**	**471.28**

Quantitative Evaluation We measure geometry and motion on the labeled 4D point-cloud sequences of ActionBench Sabathier et al. (2026) against all four baselines, and video-aligned rendering against the textured outputs of DM4D, V2M4, and ActionMesh. HoAM-4D is best on every metric of both groups. Table 1 reduces CD-3D, CD-4D, and CD-M by 4.0%, 10.1%, and 13.9% relative to the strongest baseline ActionMesh, with the gains concentrated in CD-4D and CD-M: all-frame geometry reasoning recovers coherent trajectories, while the derived synchronous relation keeps the independently predicted image displacement from introducing temporal drift. Table 2 improves CLIP from 0.9145 to 0.9312 and reduces LPIPS from 0.0766 to 0.0702 and FVD from 503.49 to 471.28, i.e. relative changes of 1.8%, 8.4%, and 6.4%, indicating that the image displacement supplies frame-specific visual motion while the derived maps attach it to the correct 3D regions without sacrificing global coherence. All four fields are thus assessed through their downstream effect rather than by treating matching as an independent output.

Qualitative Evaluation Figure 3 compares textured and normal renderings on Consistent4D Jiang et al. (2023) and our additional collection. HoAM-4D preserves articulated structure and temporal texture consistency while staying aligned with the observed silhouette, whereas the baselines show blurred texture, local shape artifacts, or temporal drift. The two strongest fail in complementary ways: ActionMesh produces coherent motion but lacks frame-wise correspondence, leaving a visible gap to the input video, while V2M4 fits appearance more directly but accumulates shape changes. HoAM-4D avoids both, as the temporal displacements model motion while the two correspondences maintain identity and resolve per-frame alignment. Animations and multi-view renderings are in the supplementary material.

4.3 Ablation Studies

Overall Ablation. The upper block of Table 3 runs the two pretrained priors on their own. Puppeteer animates a rigged mesh without video-derived motion and is weakest overall, while frame-wise 3D generation follows appearance but yields topologically inconsistent meshes, leaving CD_M and CD_{occ} undefined; both stay far from the full framework.

Removing one component at a time, volume alignment proves necessary because the four regression targets and the endpoint-distance kernel must share a common coordinate system, so dropping it costs accuracy everywhere despite an unchanged network. Without motion supervision the optimizer recovers trajectories from the video deviation and the skeletal prior alone, and CD-4D degrades the most, from 0.062 to 0.142. Without 2D deviation supervision, which is what dropping either the correspondence offsets or the conduction module amounts to, coherent geometry motion still drives the mesh but no image evidence reaches it: CD-4D stays moderate at 0.089 while CD_{occ} rises from 0.134 to 0.208. The contrast between these two rows reflects the intended division of labor between temporal coherence and frame-specific visual grounding.

Table 3: Overall ablation: pretrained priors alone, then one component removed at a time. CD_{occ} covers vertices invisible from the input view.

Variant	CLIP↑	FVD↓	CD-4D ↓	CD-M ↓	CD_{occ} ↓
Pure Puppeteer	0.7204	673.51	0.248	0.217	0.264
Pure 3D generation	0.8346	612.73	0.195	N/A	N/A
w/o volume alignment	0.9254	503.78	0.071	0.156	0.168
w/o motion sup.	0.9186	519.47	0.142	0.165	0.183
w/o 2D deviation sup.	0.9124	536.85	0.089	0.131	0.208
Full HoAM-4D	**0.9312**	**471.28**	**0.062**	**0.118**	**0.134**

Table 4: Stage II ablation, covering the head formulation, architecture, and objective.

Variant	Acc ↑	CLIP ↑	CD-4D ↓	CD-M ↓
Full model	**81.59**	**0.9312**	**0.062**	**0.118**
corr. as classification	71.93	0.8875	0.083	0.151
adjacent-frame attn.	74.26	0.8932	0.078	0.142
w/o raw motion $\mathbf{R}_t^b$	77.38	0.9064	0.071	0.133
shared branch weights	78.55	0.9127	0.068	0.127
w/o $\mathcal{L}_{cyc} + \mathcal{L}_{mm}$	79.04	0.9153	0.067	0.135
w/o $\mathcal{L}_{map}$	80.12	0.9088	0.065	0.124

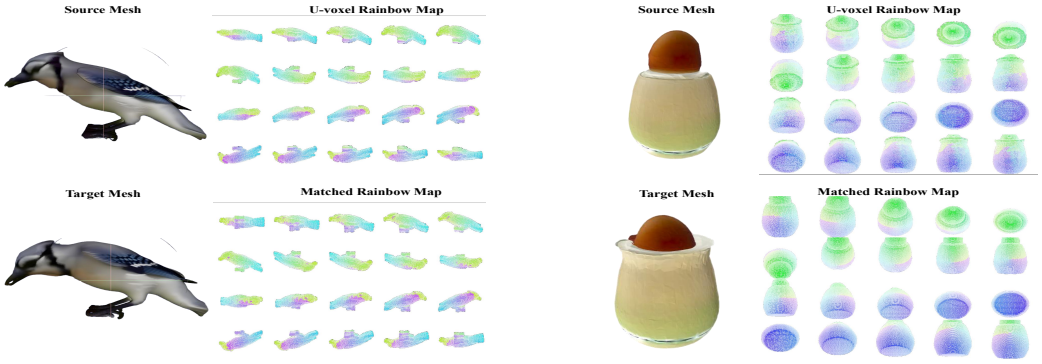


Figure 4: Predicted 4D motion field. The source mesh uses a spatial pseudocolor map and the target is colored by the predicted voxel-to-vertex motion, so consistent colors indicate coherent propagation.

Displacement Heads vs. Classification Heads. Replacing only the two correspondence-offset heads with softmax classifiers over the same candidates gives the weakest variant in the table, since cross-entropy penalizes a neighboring candidate and one on the opposite limb equally while displacement regression is graded by Euclidean error. A discrete assignment also cannot enter the geometric cycle $\mathbf{O}^{sync} - \mathbf{O}^{can} = \mathbf{D}^G$, leaving $\mathcal{L}_{cyc}$ and $\mathcal{L}_{mm}$ inapplicable and correspondence untied from motion, which is why the degradation concentrates in CD-M.

Stage II Architecture and Objectives. The remaining rows ablate the network and the objective on ActionBench Sabathier et al. (2026). Restricting temporal attention to adjacent frames removes long-range motion context and is the most damaging architectural change. Dropping $\mathbf{R}_t^b$ costs more than any single objective term, since a coarse but geometrically grounded initial correspondence is easier to refine than an unconstrained search over the volume. Sharing weights across the two branches confirms that the modalities need independent parameters despite the identical architecture. On the objective side, $\mathcal{L}_{map}$ disambiguates nearby candidates, while $\mathcal{L}_{cyc}$ and $\mathcal{L}_{mm}$ keep the fields consistent; removing them lets the image displacement drift from the geometry trajectory, again showing up in CD-M. Input features are ablated in Appendix D.1.

4D Motion Field Visualization. Under the rapid non-rigid deformation of Figure 4, the predicted displacements track the bird's wing extremities across frames, supplying the spatially coherent supervision that Stage III needs to avoid frame-wise drift.

5 Conclusion

We have presented **HoAM-4D**, which reconstructs animated meshes from monocular video by predicting four three-dimensional displacement fields: geometry and image temporal motion together with canonical and synchronous correspondence offsets. Turning the latter into endpoints makes image-to-geometry matching a differentiable consequence of displacement prediction and lets all four heads share one regression objective, while derived-map and cycle consistency connect the fields and skeleton-constrained optimization preserves fixed topology and articulated motion. Experiments demonstrate improved geometric accuracy, motion coherence, and video-aligned rendering quality over existing 4D reconstruction methods.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. In this study, no human subjects or animal experimentation was involved. All datasets used, including DyMesh Wu et al. (2025), ActionBench Sabathier et al. (2026), and Consistent4D Jiang et al. (2023), were sourced in compliance with relevant usage guidelines, ensuring no violation of privacy. We have taken care to avoid any biases or discriminatory outcomes in our research process. No personally identifiable information was used, and no experiments were conducted that could raise privacy or security concerns. We are committed to maintaining transparency and integrity throughout the research process.

REPRODUCIBILITY STATEMENT

We have made every effort to ensure that the results presented in this paper are reproducible. The code, the trained weights, and the evaluation split will be released publicly immediately upon acceptance to facilitate replication and verification. The experimental setup, including training steps, model configurations, and hardware details, is described in detail in Appendix B.2. We have also provided a full description of the scene initialization procedure in Appendix A, the displacement-to-matching derivation in Appendix C, and the evaluation protocol including the occluded-region metric CD_{occ} in Appendix B.1, to assist others in reproducing our experiments.

Additionally, all pretrained components and datasets used in the paper, namely Hunyuan3D 2.5, Puppeteer, TRELLIS, Stable Diffusion 1.5, and DINOv2 for the former and DyMesh, ActionBench, and Consistent4D for the latter, are publicly available, ensuring consistent and reproducible evaluation results. Our filtering criteria and the composition of the 96-sequence evaluation suite are stated in Section 4.

We believe these measures will enable other researchers to reproduce our work and further advance the field.

LLM USAGE

Large Language Models (LLMs) were used to aid in the writing and polishing of the manuscript. Specifically, we used an LLM to assist in refining the language, improving readability, and ensuring clarity in various sections of the paper. The model helped with tasks such as sentence rephrasing, grammar checking, and enhancing the overall flow of the text.

It is important to note that the LLM was not involved in the ideation, research methodology, or experimental design. All research concepts, ideas, and analyses were developed and conducted by the authors. The contributions of the LLM were solely focused on improving the linguistic quality of the paper, with no involvement in the scientific content or data analysis.

The authors take full responsibility for the content of the manuscript, including any text generated or polished by the LLM. We have ensured that the LLM-generated text adheres to ethical guidelines and does not contribute to plagiarism or scientific misconduct.

REFERENCES

Noam Aigerman, Roi Poranne, and Yaron Lipman. Seamless surface mappings. *ACM Transactions on Graphics (TOG)*, 34(4):1–13, 2015.

Noam Aigerman, Kunal Gupta, Vladimir G Kim, Siddhartha Chaudhuri, Jun Saito, and Thibault Groueix. Neural jacobian fields: learning intrinsic mappings of arbitrary meshes. *ACM Transactions on Graphics (TOG)*, 41(4):1–17, 2022.

Zenghao Chai, Chen Tang, Yongkang Wong, and Mohan Kankanhalli. Star: Skeleton-aware text-based 4d avatar generation with in-network motion retargeting. *arXiv preprint arXiv:2406.04629*, 2024.

Jianqi Chen, Biao Zhang, Xiangjun Tang, and Peter Wonka. V2m4: 4d mesh animation reconstruction from a single monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11643–11653, 2025.

Jiacheng Deng, Chuxin Wang, Jiahao Lu, Jianfeng He, Tianzhu Zhang, Jiyang Yu, and Zhe Zhang. Se-ornet: Self-ensembling orientation-aware network for unsupervised point cloud shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5364–5373, 2023.

Niladri Shekhar Dutt, Sanjeev Muralikrishnan, and Niloy J Mitra. Diffusion 3d features (diff3f): Decorating untextured shapes with distilled semantic features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4494–4504, 2024.

Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. *Advances in Neural Information Processing Systems*, 36:50742–50768, 2023.

Yanqin Jiang, Li Zhang, Jin Gao, Weimin Hu, and Yao Yao. Consistent4d: Consistent 360 {\deg} dynamic object generation from monocular video. *arXiv preprint arXiv:2311.02848*, 2023.

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.

Zeqiang Lai, Yunfei Zhao, Haolin Liu, Zibo Zhao, Qingxiang Lin, Huiwen Shi, Xianghui Yang, Mingxin Yang, Shuhui Yang, Yifei Feng, et al. Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details. *arXiv preprint arXiv:2506.16504*, 2025.

Itai Lang, Dvir Ginzburg, Shai Avidan, and Dan Raviv. Dpc: Unsupervised deep point correspondence via cross and self construction. In *2021 International Conference on 3D Vision (3DV)*, pp. 1442–1451. IEEE, 2021.

Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérome Maillot. Least squares conformal maps for automatic texture atlas generation. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pp. 193–202. 2023.

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.

Or Litany, Tal Remez, Emanuele Rodola, Alex Bronstein, and Michael Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE international conference on computer vision*, pp. 5659–5667, 2017.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

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.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1617–1627, 2019.

Remy Sabathier, Niloy J Mitra, and David Novotny. Lim: Large interpolator model for dynamic reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 6154–6164, 2025.

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

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.

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

Yikai Wang, Xinzhou Wang, Zilong Chen, Zhengyi Wang, Fuchun Sun, and Jun Zhu. Vidu4d: Single generated video to high-fidelity 4d reconstruction with dynamic gaussian surfels. *Advances in Neural Information Processing Systems*, 37:131316–131343, 2024.

Zijie Wu, Chaohui Yu, Fan Wang, and Xiang Bai. Animateanymesh: A feed-forward 4d foundation model for text-driven universal mesh animation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Zhen Xu, Zhengqin Li, Zhao Dong, Xiaowei Zhou, Richard Newcombe, and Zhaoyang Lv. 4dgt: Learning a 4d gaussian transformer using real-world monocular videos. 2025.

Chun-Han Yao, Yiming Xie, Vikram Voleti, Huaizu Jiang, and Varun Jampani. Sv4d 2.0: Enhancing spatio-temporal consistency in multi-view video diffusion for high-quality 4d generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13248–13258, 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. *arXiv preprint arXiv:2510.06208*, 2025.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. Learning 4d embodied world models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5337–5347, 2025.

A Scene Initialization Details

To obtain a consistent initialization, we assume that intrinsic object properties such as volumetric scale and skeletal structure remain stable across non-rigid motion, while the extrinsic camera parameters may vary over time. We therefore adopt a volume-static and camera-dynamic initialization strategy. Given an input video $\mathcal{I} = \{I_t\}_{t=1}^{T}$, Hunyuan3D 2.5 produces a frame-wise coarse mesh and Puppeteer initializes its skeleton and skinning weights:

$$\mathcal{M}_t^{raw} = \mathcal{G}_{\theta_g}(I_t), \quad (\mathcal{J}_t, \mathbf{W}_t) = \mathcal{S}_{\theta_s}(\mathcal{M}_t^{raw}), \tag{12}$$

where $\mathcal{G}_{\theta_g}$ is the 3D generation backbone and $\mathcal{S}_{\theta_s}$ is the rigging and skinning module. We keep the first-frame rig as the canonical one, i.e. $\mathcal{J} = \mathcal{J}_1$ and $\mathbf{W} = \mathbf{W}_1$.

Because independently generated meshes need not share topology or vertex order, we take the first frame as the canonical mesh $\mathcal{M}^c = (\mathcal{V}^c, \mathcal{F}^c)$ whose connectivity the final animation retains, and convert the remaining frames into sparse voxel observations that contribute geometric, appearance, and scale evidence without ever being deformed themselves.

We align the volumetric scale of every observation to the first-frame reference. Let $V_{ref} = V(1)$ and let $V_{total}(t)$ be the estimated volume of the raw mesh $\mathcal{M}_t^{raw}$ at time t. The rescaling factor is

$$\Gamma(t) = \left(\frac{V_{ref}}{V_{total}(t)} \right)^{1/3}, \tag{13}$$

and its vertices $\mathcal{V}_t^{raw}$ are updated by $\Gamma(t)(\mathcal{V}_t^{raw} - \bar{\mathbf{v}}_t) + \bar{\mathbf{v}}_t$, where $\bar{\mathbf{v}}_t$ is the mesh centroid. The rescaled meshes are the volume-aligned observations $\widehat{\mathcal{M}}_t$ of Equation 1.

For camera initialization, we follow the view search of V2M4 and add silhouette-based scale control. The first-frame camera $\boldsymbol{\pi}_1$, parameterized by elevation ϵ, azimuth ϕ, and radius ρ, is selected as

$$\boldsymbol{\pi}_1^* = \arg\max_{\epsilon,\phi,\rho} \left[\mathcal{Q}_{DreamSim}(\mathcal{R}(\widehat{\mathcal{M}}_1, \boldsymbol{\pi}_1), I_1) + \alpha \ \mathrm{IoU}(\Omega_1, \hat{\Omega}_1) \right], \tag{14}$$

where $\mathcal{R}$ is the renderer, $\mathcal{Q}_{DreamSim}$ is the cosine similarity between DreamSim Fu et al. (2023) embeddings, Ω_t is the foreground mask of I_t, and $\hat{\Omega}_t$ is the silhouette rendered from $\widehat{\mathcal{M}}_t$ under $\boldsymbol{\pi}_t$. For $t > 1$, we additionally penalize temporal camera variation:

$$\boldsymbol{\pi}_t^* = \arg\max_{\boldsymbol{\pi}_t} \left[\mathcal{Q}_{DreamSim}(\mathcal{R}(\widehat{\mathcal{M}}_t, \boldsymbol{\pi}_t), I_t) + \alpha \ \mathrm{IoU}(\Omega_t, \hat{\Omega}_t) - \mathcal{S}_{smooth}(\boldsymbol{\pi}_t, \boldsymbol{\pi}_{t-1}) \right], \tag{15}$$

where $\mathcal{S}_{smooth}$ is the squared geodesic distance between the two camera rotations plus the squared difference of their radii. We retain the first-frame camera as the canonical view and repose each observation according to the deviation between $\boldsymbol{\pi}_1^*$ and $\boldsymbol{\pi}_t^*$.

B Additional Experimental Details

B.1 Evaluation Protocol

We evaluate visual fidelity through a video-to-rendering protocol that compares the input frames with outputs rendered from the aligned viewpoints. CLIP Radford et al. (2021) measures semantic similarity, LPIPS Zhang et al. (2018) measures perceptual distance, and Frechet Video Distance (FVD) Unterthiner et al. (2018) measures sequence-level visual quality. For geometric and kinematic accuracy, we report 3D Chamfer Distance (CD_{3D}), 4D Chamfer Distance (CD_{4D}), and Motion Chamfer Distance (CD_M) Sabathier et al. (2026). These metrics quantify spatial and trajectory deviations between predicted and ground-truth representations after ICP alignment.

We additionally define an occluded-region metric CD_{occ} to evaluate the surface that the input view cannot constrain. For each frame we rasterize the posed mesh from the estimated camera $\boldsymbol{\pi}_t$ and mark a vertex as occluded when it is either back-facing or fails the depth test, which is the same visibility criterion that determines $\mathcal{V}_t^{vis}$ in Section 3.3. CD_{occ} is then the CD_{4D} accumulated over occluded vertices only, averaged across frames and sequences. A vertex occluded in one frame may be visible in another, so CD_{occ} measures how well the reconstruction behaves precisely where direct video evidence is momentarily unavailable.

B.2 Implementation Details

We employ Hunyuan3D 2.5 Lai et al. (2025) for frame-wise mesh generation and Puppeteer Song et al. (2025) to initialize the canonical skeleton and skinning weights. The TRELLIS 1.0 VAE supplies structural features for the geometry branch, while Stable Diffusion 1.5 and DINOv2 ViT-L supply appearance and semantic features for the image branch; all feature extractors remain frozen. Geometry and image features are discretized on a common nominal grid of 128^3, and only occupied voxels are instantiated. Image features are augmented directly with their x, y, and z grid coordinates rather than constructed through a separate depth-based back-projection.

Each branch uses six encoder blocks with a token width of 256 and eight attention heads. A block first performs within-frame self-attention and then all-frame cross-attention, using the current frame as queries and the concatenated sequence as keys and values. The branch-specific raw motion field is obtained by matching each voxel to the most cosine-similar candidate in the next frame and taking the coordinate difference; it is embedded by a linear layer to the token width and added after cross-attention as a condition, while the output heads directly predict complete displacements rather than residuals. Every decoder starts with frame-wise self-attention. The two motion decoders then apply temporal attention over the sequence and output the geometry and image displacements, while the two correspondence decoders instead apply cross-modal attention against the first-frame and same-frame geometry tokens respectively and output the canonical and synchronous offsets. Each decoder uses two such blocks followed by a linear regression layer. Soft matching maps are computed from offset endpoints with temperature τ and local valid-candidate masks. We train all four heads with Smooth-L_1 displacement regression and use the derived-map KL, geometric-cycle, matched-motion, and temporal-smoothness objectives in Equation 8.

We set $\alpha = 0.2$ in the camera objective and smooth camera poses across adjacent frames. The geometry temporal head is first warmed up on paired mesh trajectories, followed by the two correspondence-offset heads and the image temporal head; all modules are then jointly fine-tuned. The network is trained with AdamW. For each test scene, root poses, joint rotations, and per-vertex residuals are optimized for 200 iterations using a OneCycle schedule with a peak learning rate of 0.02, driven entirely by the two vertex-level displacement targets without any rendering loss. The residual is regularized so that its magnitude stays near a tenth of the skeletal displacement, which keeps the deformation dominated by articulated motion. All experiments run on one NVIDIA A100 GPU, and the complete pipeline takes approximately 30 minutes per 32-frame scene, broken down by stage in Table 6.

C Displacement-to-Matching Derivation

This section derives the canonical and synchronous matching maps from the two predicted correspondence-offset fields. Let $\mathbf{c}^I_{t,m}$ be image voxel m at time t, let $\mathbf{c}^G_{1,j}$ be canonical geometry candidate j, and let $\mathbf{c}^G_{t,n}$ be synchronous geometry candidate n. During training, the paired sequence provides a canonical index $j^*(t,m)$ and a synchronous index $n^*(t,m)$ for every valid image voxel. The two ground-truth offset targets are therefore

$$\mathbf{O}^{can,*}_{t,m} = \mathbf{c}^G_{1,j^*(t,m)} - \mathbf{c}^I_{t,m}, \qquad \mathbf{O}^{sync,*}_{t,m} = \mathbf{c}^G_{t,n^*(t,m)} - \mathbf{c}^I_{t,m}. \tag{16}$$

For surface samples whose voxelization distributes mass over several neighboring cells, the discrete indices are replaced by the corresponding barycentric or trilinear weighted coordinates.

The predicted offsets define two endpoints:

$$\widehat{\mathbf{p}}^{can}_{t,m} = \mathbf{c}^I_{t,m} + \widehat{\mathbf{O}}^{can}_{t,m}, \qquad \widehat{\mathbf{p}}^{sync}_{t,m} = \mathbf{c}^I_{t,m} + \widehat{\mathbf{O}}^{sync}_{t,m}. \tag{17}$$

We convert the canonical endpoint to a probability distribution over first-frame geometry voxels using a temperature-controlled distance kernel:

$$\widehat{M}^{can}_{t,mj} = \frac{\exp\big(-\|\widehat{\mathbf{p}}^{can}_{t,m} - \mathbf{c}^G_{1,j}\|_2^2/\tau\big)}{\sum_{k\in\mathcal{C}_1(t,m)} \exp\Big(-\|\widehat{\mathbf{p}}^{can}_{t,m} - \mathbf{c}^G_{1,k}\|_2^2/\tau\Big)}, \quad j \in \mathcal{C}_1(t,m), \tag{18}$$

where $\mathcal{C}_1(t,m)$ contains valid canonical candidates. Synchronous matching is obtained analogously:

$$\widehat{M}^{sync}_{t,mn} = \frac{\exp\big(-\|\widehat{\mathbf{p}}^{sync}_{t,m} - \mathbf{c}^G_{t,n}\|_2^2/\tau\big)}{\sum_{k\in\mathcal{C}_t(m)} \exp\Big(-\|\widehat{\mathbf{p}}^{sync}_{t,m} - \mathbf{c}^G_{t,k}\|_2^2/\tau\Big)}, \quad n \in \mathcal{C}_t(m). \tag{19}$$

Candidates are restricted to occupied voxels within a local radius around the predicted endpoint for sparse computation. Invalid or occluded observations are excluded by the supervision mask and assigned to an unmatched bin. At inference, a hard correspondence can be recovered by the nearest endpoint candidate, but all training losses use the differentiable soft distributions in Equations 18 and 19.

The image temporal target is expressed in the same coordinate system as geometry motion by transferring the ground-truth geometry displacement through the synchronous relation:

$$\mathbf{D}^{I,*}_{t,m} = \sum_{n} M^{sync,*}_{t,mn} \mathbf{D}^{G,*}_{t,n}. \tag{20}$$

This avoids defining an incompatible image-space flow target. The four predictions also form a geometric cycle. For a correct canonical–synchronous pair,

$$\mathbf{c}^{I}_{t,m} + \mathbf{O}^{can,*}_{t,m} = \mathbf{c}^{G}_{1,j^*}, \tag{21}$$

$$\mathbf{c}^{I}_{t,m} + \mathbf{O}^{sync,*}_{t,m} = \mathbf{c}^{G}_{t,n^*}. \tag{22}$$

Subtracting the two equalities yields

$$\mathbf{O}^{sync,*}_{t,m} - \mathbf{O}^{can,*}_{t,m} = \mathbf{c}^{G}_{t,n^*} - \mathbf{c}^{G}_{1,j^*} = \mathbf{D}^{G,*}_{t,j^*}. \tag{23}$$

Replacing the hard correspondence with its soft expectation gives the geometric-cycle and matched-motion losses used in Equation 8. Thus, matching remains probabilistic and differentiable, while every learned head is supervised primarily as a three-dimensional displacement regressor.

D Additional Ablations and Robustness Analysis

D.1 Input Feature Ablation

Table 5 varies the frozen encoders that build the two token streams, keeping the architecture and all objectives fixed. DINO provides the strongest semantic cue to the image branch, and removing it costs the most on both correspondence accuracy and rendering quality, since without it the image voxels carry little that can be matched against geometry. TRELLIS supplies complementary geometric structure that the image features cannot recover, so its removal mainly affects the geometry branch. Stable Diffusion adds appearance detail and mainly improves image motion, giving the smallest individual contribution of the three encoders. Positional embeddings are indispensable for endpoint-based correspondence, because the matching maps are derived from distances between predicted endpoints and candidate voxel coordinates; without coordinate information the network cannot place an endpoint in the volume consistently.

Table 5: Ablation of the frozen input features used to build geometry and image tokens.

Variant	Acc ↑	CLIP↑	CD-4D↓	CD-M↓
Full model	**81.59**	**0.9312**	**0.062**	**0.118**
w/o DINO	68.84	0.7853	0.072	0.138
w/o TRELLIS	74.82	0.8614	0.068	0.128
w/o pos. embed.	78.15	0.8847	0.065	0.124
w/o Stable Diff.	78.91	0.8985	0.064	0.122

D.2 Runtime Analysis

Table 6 reports the runtime–quality tradeoff on a single NVIDIA A100 GPU for a 32-frame sequence. HoAM-4D is substantially faster than V2M4 while achieving lower CD-4D and CD-M. The difference does not come from a cheaper initialization, since we adopt the same DreamSim-based view search as V2M4 and pay the same cost for it. It comes from the final stage: because the four displacement fields already supply a learned motion prior, the scene-specific optimization only has to refine joint transforms, whereas V2M4 must search for motion from rendering losses alone. HoAM-4D remains slower than the feed-forward ActionMesh baseline, which performs no test-time optimization at all, but this additional computation is what yields the more accurate geometry, motion, and input-view alignment reported in Tables 1 and 2. Network inference itself is negligible, since Stage II is a single forward pass over the sequence.

Table 6: Runtime and quality comparison on one NVIDIA A100 GPU for a 32-frame sequence. Our total is broken down by stage; the camera search is the same procedure as in V2M4.

Method	Runtime/scene↓	CD-4D↓	CD-M↓
V2M4	187 min	0.223	0.500
ActionMesh	**3 min**	0.069	0.137
Ours (total)	30 min	**0.062**	**0.118**
Stage I: mesh generation & rigging	8 min	–	–
Stage I: camera search	35 s	–	–
Stage II: network inference	$<$1 min	–	–
Stage III: mesh optimization	21 min	–	–

D.3 Camera Sensitivity

We perturb the initial camera pose before refinement to evaluate robustness. As shown in Table 7, moderate perturbations cause limited performance variation, and refinement recovers a pose close to the reference view. Camera estimation therefore supplies a stable canonical frame, while the four displacement fields, endpoint-derived matching, and skeleton-constrained optimization absorb modest remaining errors.

Table 7: Sensitivity analysis of camera initialization.

Initial Noise	Pose Error	Search Time	CLIP↑	FVD↓	CD-4D↓
$[-45°, 45°]$	$[-5°, 5°]$	35s	0.9098	537.35	0.068
$[-15°, 15°]$	$[-1°, 1°]$	35s	0.9207	483.12	0.063
No perturbation	$\sim 0°$	35s	**0.9312**	**471.28**	**0.062**

E Limitations and Future Directions

HoAM-4D assumes a fixed canonical mesh and therefore cannot handle extreme topology changes, such as object fragmentation, structural breakage, or the appearance of new objects during a sequence. These events create or remove surface components and invalidate the fixed connectivity, correspondence, and skinning relationships used by our skeleton-guided deformation. Extending the framework with topology-adaptive or compositional representations is an important direction for future work.