Neu-PiG: Neural Preconditioned Grids for Fast Dynamic Surface Reconstruction on Long Sequences

Julian Kaltheuner Hannah Dröge Markus Plack Patrick Stotko Reinhard Klein
University of Bonn
{kaltheun,droege,mplack,stotko,rk}@cs.uni-bonn.de

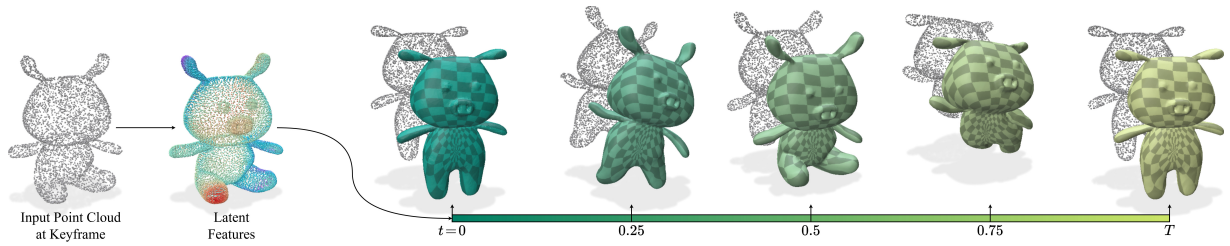

Figure 1. We present Neu-PiG, a method that learns spatially smooth and temporally coherent deformations from dynamic point clouds. Starting from input point clouds (left), a unified latent space parameterized by an initial reference mesh (middle) is optimized through multi-scale grids to produce high-fidelity deformations (right) without relying on strong priors or correspondences.

Abstract

Temporally consistent surface reconstruction of dynamic 3D objects from unstructured point cloud data remains challenging, especially for very long sequences. Existing methods either optimize deformations incrementally, risking drift and requiring long runtimes, or rely on complex learned models that demand category-specific training. We present Neu-PiG*, a fast deformation optimization method based on a novel preconditioned latent-grid encoding that distributes spatial features parameterized on the position and normal direction of a keyframe surface. Our method encodes entire deformations across all time steps at various spatial scales into a multi-resolution latent grid, parameterized by the position and normal direction of a reference surface from a single keyframe. This latent representation is then augmented for time modulation and decoded into per-frame 6-DoF deformations via a lightweight multi-layer perceptron (MLP). To achieve high-fidelity, drift-free surface reconstructions in seconds, we employ Sobolev preconditioning during gradient-based training of the latent space, completely avoiding the need for any explicit correspondences or further priors. Experiments across diverse human and animal datasets demonstrate that Neu-PiG outperforms state-the-art approaches, offering both superior accuracy and scalability to long sequences while running at least 60× faster than existing training-free methods and achieving inference speeds on the same order as heavy pre-trained models.*

1. Introduction

Understanding and modeling how shapes in the surrounding world evolve over time is fundamental to perceiving and interacting with dynamic environments. Whether observing a human performing complex motions, an animal moving through its habitat, or an object undergoing physical transformations, capturing such spatiotemporal deformations is key to reasoning about the physical and semantic structure of the world. This capability underpins a wide range of applications in augmented and virtual reality (AR/VR), robotics, autonomous systems, and computer vision, including realistic motion capture, scene reconstruction, and human–robot interaction.

A common strategy to represent deformable shapes is through parametric models, which provide a controllable yet expressive space of possible deformations. Prominent examples include SMPL [40] and SMPL-X [53] for human bodies, FLAME [34] for facial geometry, and MANO [57] for hands. While these models offer compact and interpretable control, they are inherently limited to the object categories for which they were designed. To move beyond category-specific representations, template-free methods have been proposed, which can broadly be divided into optimization-based and learning-based approaches. Optimization-based methods (*e.g.*, Dyno-

Surf [71], PDG [25]) directly optimize surface deformations for each sequence, producing high-quality results but at the cost of long runtimes. In contrast, learning-based approaches amortize reconstruction through category-specific priors, achieving fast inference but struggling to generalize beyond their training domain. Despite their complementary strengths, a general, efficient, and category-agnostic approach for modeling dynamic shape deformations remains an open challenge.

To address these limitations, we propose *Neu-PiG*, a novel method which introduces **Neu**ral **P**recondi**t**ioned **G**rids as an efficient representation for fast estimation of spatially smooth and temporally coherent deformations from dynamic point cloud sequences. Starting from an initial mesh estimate as a canonical reference, we learn the complete deformation across all time steps via a novel surface encoding that is parameterized by vertex positions and normal directions. To enable this without relying on correspondences or restricting the method's generalizability by strong priors, our key insight lies in the geometric structure and the associated optimization procedure of the latent embedding. We store learnable feature vectors at different spatial scales in a multi-resolution voxel grid and aggregate them for a given query point via trilinear interpolation and averaging. This ensures that we learn the absolute deformations at each level rather than a decomposition of them which would be sensitive to small deviations and thus significantly less stable during optimization. We further stabilize this process by applying Sobolev preconditioning to the gradient-based training of the latent grid. Consequently, this approach allows us to obtain a unified latent space which can be augmented for time-modulation and then decoded into high-fidelity deformations via a lightweight MLP. As illustrated in Fig. 1 and demonstrated in extensive experiments, Neu-PiG achieves superior reconstruction accuracy at an order-of-magnitude lower runtimes than state-of-the-art optimization-based approaches and comparable to the inference times of pretrained models.

In summary, our key contributions are:

- We propose a fast optimization-based method that estimates temporally coherent deformations of arbitrary subjects, including humans and animals, from sequential point cloud data.
- We introduce a novel preconditioned surface encoding based on vertex positions and normal directions of a reference mesh that captures the full deformations across all time steps into a unified latent space.
- We design a multi-scale latent grid representation to enforce spatial consistency both on global and local scales, enabling fast decoding via a lightweight MLP.

Code is available at: https://github.com/vc-bonn/neu-pig

2. Related Work

2.1. Parametric Template Models

Category-specific reconstruction has historically relied on deforming a canonical mesh to match observations [2, 34, 40, 53, 68, 69]. Such models represent facial and body geometry within learned low-dimensional manifolds, enabling expressive control of identity and pose [6, 39, 59]. Extensions further incorporate clothing, soft tissue, or body-shape variability [1, 5, 65], making parametric systems dominant for structured capture tasks. Despite their success, the linear subspace assumption of these models restricts generalization beyond the training domain [40, 69]. Recent neural formulations replace explicit templates with implicit shape functions that learn deformation spaces directly from data [34, 68]. Current state-of-the-art methods decouple skeletal bases from surface-shape representations [51], or body-shape from clothing [14], while complementary approaches bridge parametric and implicit formulations by learning MLP weights for an SDF-based body model [43].

Our approach follows this trend toward implicit representations but removes category dependence entirely: Neu-PiG reconstructs dynamic surfaces from unstructured sequences using a latent spatial grid and a lightweight decoder without predefined templates.

2.2. Deformation Field Estimation

Non-rigid motion recovery aims to describe how points or surfaces evolve across time [46, 62]. Early methods estimated dense warp fields via optimization, *e.g.* through embedded graphs [62] or volumetric tracking [46]. Subsequent dynamic radiance formulations employ Gaussian primitives for compact 4D mapping [27, 42] or separate static and moving components for SLAM-style pipelines [33, 73].

Neural deformation models shifted the focus from explicit alignment to learned motion inference [8, 9, 35]. Hierarchical and continuous formulations parameterize motion as neural vector fields [16, 22, 23, 48, 64] or as spatio-temporal surfaces [49]. Probabilistic variants treat deformation as a distribution over 4D geometry [11], while canonical-space techniques couple shape and motion within rendering frameworks [32, 38, 52, 54, 72]. Optimization-only strategies such as DynoSurf [71] achieve supervision-free tracking but remain computationally demanding.

Latent and Implicit Representations. Recent advances move beyond explicit displacement fields by storing deformation cues in spatially distributed latent representations that are decoded into motion [52, 54]. Dense voxel features paired with compact decoders provide locality but can be memory-intensive at high resolutions [35, 71]. As a foundational alternative for neural fields, multi-resolution hash encodings supply compact capacity for static recon-

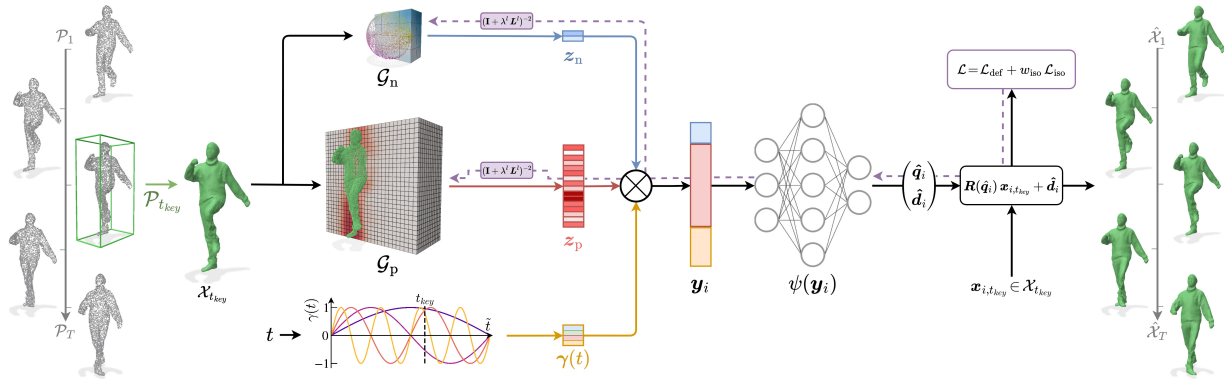

Figure 2. Overview of Neu-PiG. A reference surface $\mathcal{X}_{t_{\text{key}}}$ is first generated from the input point cloud at a keyframe $\mathcal{P}_{t_{\text{key}}}$. Position and normal-direction-embedded latent features $\boldsymbol{z}_{\text{p}}$ and $\boldsymbol{z}_{\text{n}}$ are then sampled from multi-resolution preconditioned grids and combined with a time embedding $\boldsymbol{\gamma}(t)$. A lightweight MLP ψ predicts per-frame 6-DoF deformations that are applied to $\mathcal{X}_{t_{\text{key}}}$ to reconstruct each frame $\hat{\mathcal{X}}_t$. Sobolev preconditioning in the latent space enforces spatial smoothness and temporal coherence across the sequence during optimization.

struction [44]; however, their collision-prone, non-smooth parameterization is ill-suited to coherent non-rigid motion and, to our knowledge, has not been adopted for deformation modeling. For dynamic scenes, factorized latent structures impose stronger continuity priors across space and time-tensor decompositions and separable planes yield smoother fields with lightweight decoders [10, 13, 20]. In parallel, implicit deformation fields map canonical coordinates to observations via learned warps, often combining latent features with time embeddings for scalable long-horizon tracking [11, 52, 54]. Complementary trends instantiate related ideas in alternative primitives by embedding motion in per-Gaussian codes for deformable 3D splats [3]. Following these directions, but focusing specifically on stable latent carriers for deformation, we group our design around a preconditioned voxel representation decoded by a lightweight MLP; for broader context on latent field formulations in dynamic settings, see [74].

Neu-PiG differs from explicit per timestep deformation grid formulations such as Preconditioned Deformation Grids [25] by maintaining a single preconditioned latent grid shared across all frames, yielding smooth, drift-free results on extended sequences.

2.3. Preconditioning

Stabilizing gradient-based optimization through preconditioning is well-established in numerical geometry [18, 30, 31], and while challenging in high-dimensional latent spaces, graph- and learning-based variants adapt such schemes for data-driven problems [15, 21, 37, 58, 66].

Sobolev gradient methods [45] compute updates under smoother inner products, improving convergence in shape and surface optimization [19, 25, 41, 55, 56], and have shown benefits in training scenarios [4, 17, 28, 50], *e.g.* by accelerating ReLU network convergence [50]. They have since appeared in non-rigid reconstruction [24, 61], differentiable rendering [47], and spatiotemporal filtering [12].

Neu-PiG extends this principle to latent voxel spaces, propagating Sobolev-filtered gradients through a learned feature grid to ensure smooth, coherent updates of latent codes across long dynamic sequences.

3. Method

Given a sequence of unstructured point clouds $\{\mathcal{P}_t\}_{t=1}^{T}$ where each $\mathcal{P}_t = \{\boldsymbol{p}_{i,t} \in \mathbb{R}^3\}_{i=1}^{N_t}$, we address the problem of dynamic surface reconstruction by estimating deformations from a single reference mesh defined at a keyframe $t_{\text{key}} \in \{1, \ldots, T\}$, to all other frames. An overview of this setup is illustrated in Fig. 2. The reference mesh consists of vertices and their associated vertex normals, $\mathcal{X}_{t_{\text{key}}} = \{(\boldsymbol{x}_{i,t_{\text{key}}}, \boldsymbol{n}_{i,t_{\text{key}}}) \in \mathbb{R}^3 \times \mathbb{R}^3\}_{i=1}^{N_{t_{\text{key}}}}$ (see Sec. 3.1). We encode the deformations into smooth latent features based on the vertex positions and normal directions of $\mathcal{X}_{t_{\text{key}}}$ and store them in a pair of preconditioned multi-resolution voxel grids. Our latent-grid formulation can be interpreted as a spatially coherent spatio-temporal field optimized under Sobolev regularization. Unlike implicit neural representations that encode geometry entirely in network weights, our factorized grid–decoder formulation distributes high-dimensional features over space, enabling a MLP ψ to reconstruct dynamic geometry from local content. This interpretation motivates our preconditioned latent-field design described in Sec. 3.2. These latent encodings are combined with frequency-encoded timesteps t and decoded by a lightweight MLP ψ that predicts per-frame deformations, transforming the reference mesh $\mathcal{X}_{t_{\text{key}}}$ into temporally coherent reconstructions $\{\hat{\mathcal{X}}_t\}_{t=1}^{T}$, see Secs. 3.3 and 3.4. We jointly optimize the voxel grids and network parameters using a deformation loss and an isometry loss, without any

additional regularization or priors, see Sec. 3.5.

3.1. Initialization

We begin by selecting a keyframe t_{key}, from which we can reconstruct a surface mesh with suitable topology. This frame acts as a canonical reference for all subsequent deformations. Following the method of PDG [25], we assess the spatial extent of each input point cloud $\mathcal{P}_t$, weighted by a bias term that favors the temporal midpoint of the sequence. The point cloud corresponding to the selected keyframe, $\mathcal{P}_{t_{\text{key}}}$, is then converted into an initial mesh $\mathcal{X}_{t_{\text{key}}}$ using screened Poisson surface reconstruction [26]. Note that the precise geometric alignment of $\mathcal{X}_{t_{\text{key}}}$ with $\mathcal{P}_{t_{\text{key}}}$ is not critical, as long as the topology is correct, since we estimate deformations for all frames, including t_{key}.

We initialize the latent features stored at each level of the multi-resolution voxel grid with zero vectors. Furthermore, we set the weights of all but the last layer of the MLP to random values, while zeroing the weights of the last one. This initializes the deformations as identity transformations and lets them evolve gradually during optimization, avoiding sudden jumps that could disrupt latent-space consistency and lead to incoherent results.

3.2. Multi-Resolution Latent Grids

To efficiently represent the geometric structure across spatial scales, we encode the reference surface $\mathcal{X}_{t_{\text{key}}}$ in a hierarchy of preconditioned voxel grids. Each grid level stores latent features at a distinct spatial frequency: coarser levels capture global deformation patterns, while finer levels encode high-frequency geometric detail. This hierarchical representation serves as an expressive latent structure for the deformation network introduced in Sec. 3.3.

Position and Normal-Direction Embeddings. For each vertex $\boldsymbol{v}_{i,t_{\text{key}}} = (\boldsymbol{x}_{i,t_{\text{key}}}, \boldsymbol{n}_{i,t_{\text{key}}})$ on the reference surface, we query two distinct grids: a position-based grid $\mathcal{G}_{\text{p}}$ that encodes spatial context, and a normal-direction-based grid $\mathcal{G}_{\text{n}}$ that captures local orientation information. This enables spatially adjacent regions with differing normal directions to deform independently. Each grid cell stores a learnable feature vector and we retrieve interpolated features at $\boldsymbol{x}_{i,t_{\text{key}}}$ and $\boldsymbol{n}_{i,t_{\text{key}}}$ via trilinear interpolation, yielding the latent vectors $\boldsymbol{z}_{\text{p}}(\boldsymbol{x}_{i,t_{\text{key}}})$ and $\boldsymbol{z}_{\text{n}}(\boldsymbol{n}_{i,t_{\text{key}}})$, respectively. As the position-based embedding is the most important factor in deformation, we store 30-dimensional latent features at each cell of $\mathcal{G}_{\text{p}}$, resulting in $\boldsymbol{z}_{\text{p}} \in \mathbb{R}^{30}$. In contrast, the orientation-based embedding in $\mathcal{G}_{\text{n}}$ stores 2-dimensional latent features at each grid cell, resulting in $\boldsymbol{z}_{\text{n}} \in \mathbb{R}^2$.

Grid Hierarchy. The position grid $\mathcal{G}_{\text{p}}$ is defined as a multi-resolution hierarchy with progressively increasing spatial resolution [25]. It consists of eight levels, with the coarsest grid containing 2^3 cells and the finest 32^3 cells, achieved by increasing the grid resolution by 3 elements in each level. Instead of treating the grid levels independently, we aggregate their outputs into a unified latent representation, ensuring that each point $\boldsymbol{x}_{i,t_{\text{key}}}$ is associated with a spatially coherent latent feature:

$$\boldsymbol{z}_{\text{p}}(\boldsymbol{x}_{i,t_{\text{key}}}) = \frac{1}{L}\sum_{l=1}^{L} \boldsymbol{z}_{\text{p}}^{l}(\boldsymbol{x}_{i,t_{\text{key}}}). \tag{1}$$

The normal-direction-based grid $\mathcal{G}_{\text{n}}$ is defined as a single-resolution grid of size 4^3, which is sufficient to encode local orientation changes, despite the potentially large normal variations in nearby regions.

3.3. Temporal Deformation Model

Our temporal deformation model predicts per-frame transformations that deform the reference mesh $\mathcal{X}_{t_{\text{key}}}$ into temporally varying reconstructions $\{\hat{\mathcal{X}}_t\}_{t=1}^{T}$. Each vertex in $\mathcal{X}_{t_{\text{key}}}$ is represented by two latent descriptors from the preconditioned voxel grids $\mathcal{G}_{\text{p}}$ and $\mathcal{G}_{\text{n}}$, a position-based encoding $\boldsymbol{z}_{\text{p}}(\boldsymbol{x}_{i,t_{\text{key}}})$ and a normal direction-based encoding $\boldsymbol{z}_{\text{n}}(\boldsymbol{n}_{i,t_{\text{key}}})$. For each timestep t, these descriptors are combined with a time embedding $\boldsymbol{\gamma}(t)$ to form the input vector:

$$\boldsymbol{y}_i = \big(\boldsymbol{z}_{\text{n}}(\boldsymbol{n}_{i,t_{\text{key}}}), \boldsymbol{z}_{\text{p}}(\boldsymbol{x}_{i,t_{\text{key}}}), \boldsymbol{\gamma}(t)\big)^T \in \mathbb{R}^{2+30+8}. \tag{2}$$

This input formulation provides the network with both spatial and directional context, enabling the generation of temporally coherent deformations that are locally smooth and globally consistent across the sequence.

Fourier Time Encoding. To represent time in a continuous, multi-scale manner, we normalize the timestep to the range $[0, 1]$ via $\tilde{t} = (t-1)/(T-1)$ and map it to a sinusoidal embedding following the Fourier feature formulation [63]. Specifically, we define the time embedding as:

$$\boldsymbol{\gamma}(t) = \big[\sin(\pi\nu_j\tilde{t}), \cos(\pi\nu_j\tilde{t})\big]_{j=1}^{M} \in \mathbb{R}^{2M}, \tag{3}$$

where each frequency $\nu_j = 2^{j-1}$ modulates the temporal signal at a distinct scale. This encoding allows the network to capture both low-frequency, slowly varying motions and high-frequency, transient deformations. In practice, we use $M = 4$ frequencies, resulting in an 8D time embedding.

Network Architecture. The deformation network ψ is implemented as a lightweight MLP that maps the input vector $\boldsymbol{y}$ to a transformation consisting of a rotation quaternion $\boldsymbol{q} \in \mathbb{R}^4$ and a displacement vector $\boldsymbol{d} \in \mathbb{R}^3$:

$$(\boldsymbol{q}, \boldsymbol{d})^T = \psi(\boldsymbol{y}). \tag{4}$$

The network comprises three fully connected layers, each with 512 hidden units and LeakyReLU activations. The final layer outputs a 7-dimensional transformation vector. All parameters of ψ are optimized jointly with the latent grid features during the reconstruction process.

3.4. Transformation Mapping

To apply the predictions from the network ψ while ensuring stability and smoothness during optimization, we map each component to explicit geometric transformations.

Rotation Component. Let $\boldsymbol{q} = (q_w, q_x, q_y, q_z)^T \in \mathbb{R}^4$ denote the quaternion parameters predicted by the deformation network. To enforce that a zero-valued network output corresponds to the identity rotation, we offset the scalar component q_w by 1, i.e.,

$$\hat{\boldsymbol{q}} = \frac{(1 + q_w, q_x, q_y, q_z)^T}{\|(1 + q_w, q_x, q_y, q_z)^T\|}, \tag{5}$$

and subsequently normalize the quaternion to unit length. This formulation provides a continuous, stable rotation representation, avoiding singularities in *e.g.* Euler angles.

Translation Component. To decouple rotational and displacement effects, the network predicts a displacement vector $\boldsymbol{d} \in \mathbb{R}^3$. Since all points lie within the normalized coordinate space $[-1, 1]^3$, we constrain displacements by applying a hyperbolic tangent function:

$$\hat{\boldsymbol{d}} = \tanh(\alpha\, \boldsymbol{d}), \qquad \alpha = 0.1. \tag{6}$$

This formulation prevents excessive displacements while still allowing smooth, learnable displacements.

Final Transformation. The complete transformation applied to each vertex $\boldsymbol{x}_{i,t_{\text{key}}} \in \mathcal{X}_{t_{\text{key}}}$ can now be expressed as:

$$\hat{\boldsymbol{x}}_{i,t} = \boldsymbol{R}(\hat{\boldsymbol{q}}_i)\, \boldsymbol{x}_{i,t_{\text{key}}} + \hat{\boldsymbol{d}}_i. \tag{7}$$

Here, $\boldsymbol{R}(\hat{\boldsymbol{q}})$ denotes the rotation matrix corresponding to the unit quaternion $\hat{\boldsymbol{q}}$. Together, the per-vertex quaternion-based rotations and bounded displacements form a compact and stable transformation representation, facilitating smooth temporal deformations across the sequence.

3.5. Optimization Objectives

Starting from the reference surface $\mathcal{X}_{t_{\text{key}}}$ defining the object topology, we optimize deformation fields that map it directly to each target point cloud $\mathcal{P}_t$. Our objective combines a deformation loss and an isometry loss, which together balance reconstruction accuracy and surface smoothness:

$$\mathcal{L} = \mathcal{L}_{\text{def}} + w_{\text{iso}}\, \mathcal{L}_{\text{iso}}. \tag{8}$$

As our MLP architecture inherently promotes smooth deformations, we use a relatively small isometry weight $w_{\text{iso}} = 100$ which downweights $\mathcal{L}_{\text{iso}}$ to approximately 10% of the magnitude of $\mathcal{L}_{\text{def}}$.

Preconditioning. To improve convergence stability, we follow PDG [25] and apply Sobolev preconditioning [47] to smooth gradient updates of the latent grids $\mathcal{G}_{\text{p}}$ and $\mathcal{G}_{\text{n}}$. The parameters $\boldsymbol{z}^l$ of grid level l are updated as

$$\boldsymbol{z}^l \leftarrow \boldsymbol{z}^l - \eta\, (\mathbf{I} + \lambda^l\, \boldsymbol{L}^l)^{-2}\, \frac{\partial \mathcal{L}}{\partial \boldsymbol{z}^l}, \tag{9}$$

where η denotes the learning rate, $\mathbf{I}$ is the identity matrix, $\lambda^l > 0$ controls the strength of spatial smoothing, and $\boldsymbol{L}^l$ is the Laplacian matrix at level l. Here, $(\mathbf{I} + \lambda^l\, \boldsymbol{L}^l)^{-2}$ acts as a low-pass filter on the gradient, coupling neighboring cells and promoting spatially coherent latent updates that remain temporally stable across the sequence. The bandwidth parameter λ^l can be tuned to match the decoder's effective frequency capacity [60, 70], ensuring that latent-field variations remain within the MLP's learnable range. In contrast to PDG [25], which preconditions raw deformation fields, Neu-PiG applies Sobolev preconditioning to learned latent vectors, allowing high-dimensional features to capture richer local variations, while a single MLP ensures temporally consistent behavior across space and time.

Deformation Loss. The main objective function measures the alignment between the deformed surface $\hat{\mathcal{X}}_t$ and the input point cloud $\mathcal{P}_t$ at each timestep:

$$\mathcal{L}_{\text{def}} = \frac{1}{T} \sum_{t=1}^{T} w_{\text{conf}}(t) \cdot L_{\text{CD}}(\hat{\mathcal{X}}_t, \mathcal{P}_t), \tag{10}$$

where L_{CD} is the robust Chamfer distance [71], and $w_{\text{conf}}(t)$ a time-adaptive confidence weight, originally proposed in PDG [25]. Although our method estimates the total deformation from the reference surface rather than deformation between consecutive frames, confidence weighting is beneficial, particularly when combined with our frequency-split timestep encoding, which implicitly accounts for past reconstruction accuracy. We define the weight as

$$w_{\text{conf}}(t) = \prod_{\tau=t_{\text{key}}}^{t} \omega(\tau)^{\delta}, \tag{11}$$

$$\omega(\tau) = \frac{1}{1 + \max(0, \text{cd}_\tau - \text{cd}_{t_{\text{key}}})} \in [0, 1], \tag{12}$$

where $\omega(\tau)$ represents the reconstruction performance at time step τ relative to the keyframe t_{key}. We compute this weight via the Chamfer distance $\text{cd}_\tau = L_{\text{CD}}(\text{sg}(\hat{\mathcal{X}}_\tau), \mathcal{P}_\tau)$, where sg denotes the stop-gradient operator. Note that

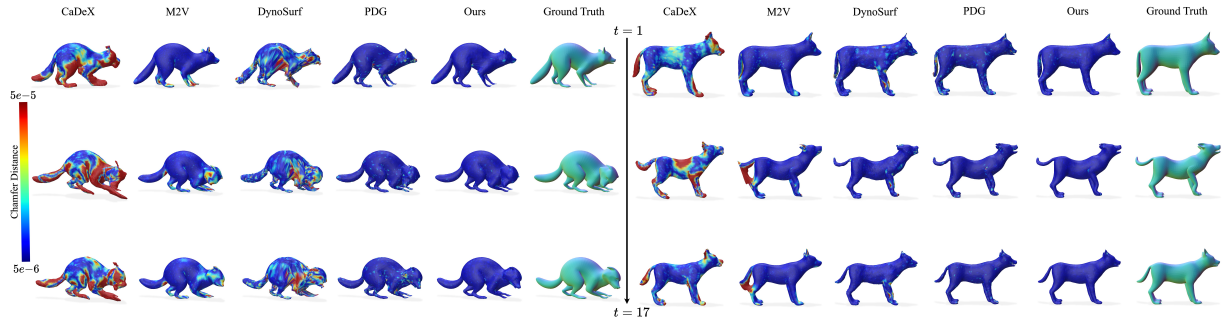

Figure 3. Qualitative reconstructions on DT4D. Neu-PiG preserves geometry and temporal consistency across complex animal motions.

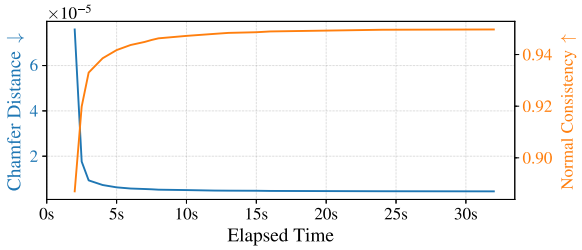

Figure 4. Convergence of Chamfer Distance and Normal Consistency on DT4D, showing high-quality reconstructions in seconds.

$\omega(t_{\text{key}}) = 1$. Additionally, to mitigate optimization deadlocks and improve convergence stability, we employ a catch-up variable $\delta = 1 - \sqrt{\bar{e}}$, which gradually increases the confidence during training. Here, $\bar{e} = e/e_{\max}$ denotes the normalized optimization epoch.

Isometry Loss. To preserve local structure, we use an isometry loss that penalizes changes in edge lengths $\hat{\boldsymbol{e}}_{ij,t} = \hat{\boldsymbol{x}}_{i,t} - \hat{\boldsymbol{x}}_{j,t}$ of the deformed meshes:

$$\mathcal{L}_{\text{iso}} = \frac{1}{T\,|\mathcal{E}|} \sum_{t=1}^{T} \sum_{(i,j)\in\mathcal{E}} \Big| \|\hat{\boldsymbol{e}}_{ij,t}\| - \text{sg}\big(\|\hat{\boldsymbol{e}}_{ij,t_{\text{key}}}\|\big) \Big|, \quad (13)$$

where $\mathcal{E}$ denotes the set of mesh edges.

Implementation Details. We jointly optimize the grid features and MLP parameters using the Adam optimizer [29] with its default parameters. The MLP uses a learning rate of 10^{-3}, while each grid level l follows a per-level rate of 0.005×2.5^{l}. The smoothness weight is defined as $\lambda = 0.4 \times 1.5^{l}$. To highlight the trade-off between accuracy and runtime, we report two configurations evaluated on a NVIDIA RTX 4090 that differ only in optimization length, 250 epochs (Ours†) and 1000 epochs (Ours).

4. Evaluation

We evaluate Neu-PiG against state-of-the-art methods, distinguishing between learning-based approaches that rely on category-specific priors and training-free methods that directly optimize deformations from point clouds without pretraining. The learning-based baselines include CaDeX [32] and M2V [11], the latter requiring predefined inter-frame correspondences. For training-free baselines, we compare against DynoSurf [71] and PDG [25]. Experiments are conducted on three public benchmarks: DFAUST [7] (human motion), AMA [67] (clothed human performance), and DT4D [36] (articulated animals). Unless stated otherwise, each sequence contains $T = 17$ frames following previous works [25, 71] and consists of 5000 points per timestep.

Table 1. Quantitative comparison on DFAUST, AMA, and DT4D. Neu-PiG achieves best accuracy and temporal coherence while being 60× faster than prior training-free methods.

		CD $[\times 10^{-5}]\downarrow$	NC ↑	F-0.5% ↑	Corr. ↓	Time ↓
DFAUST	CaDeX	3.68	0.941	0.730	0.013	**3 s**
	DynoSurf	2.13	0.953	0.980	0.010	30 min
	M2V	1.61	0.960	0.877	-	4 s
	PDG	0.46	0.956	0.987	0.017	7 min
	Ours†	0.40	0.967	**0.989**	**0.008**	8 s
	Ours	**0.39**	**0.968**	**0.989**	0.009	32 s
DT4D	CaDeX	56.51	0.870	0.386	0.050	**3 s**
	DynoSurf	15.18	0.919	0.773	**0.032**	30 min
	M2V	7.61	0.944	0.792	-	4 s
	PDG	1.67	0.956	0.948	0.043	7 min
	Ours†	0.96	**0.969**	0.962	0.034	8 s
	Ours	**0.87**	**0.969**	**0.968**	0.034	32 s
AMA	DynoSurf	1.01	0.918	0.921	0.044	30 min
	PDG	0.79	0.926	0.961	0.037	7 min
	Ours†	0.53	0.946	0.978	0.024	**8 s**
	Ours	**0.44**	**0.951**	**0.988**	**0.018**	32 s

Evaluation Metrics. We assess reconstruction quality using ℓ_2-Chamfer Distance (CD), Normal Consistency (NC), F-score, and Correspondence Error (Corr.). CD measures bidirectional surface deviations, while NC evaluates local smoothness and normal alignment. The Correspondence Error measures temporal consistency of vertex trajectories across frames and the F-score is chosen with a threshold of 0.5%. We additionally report the average per-sequence

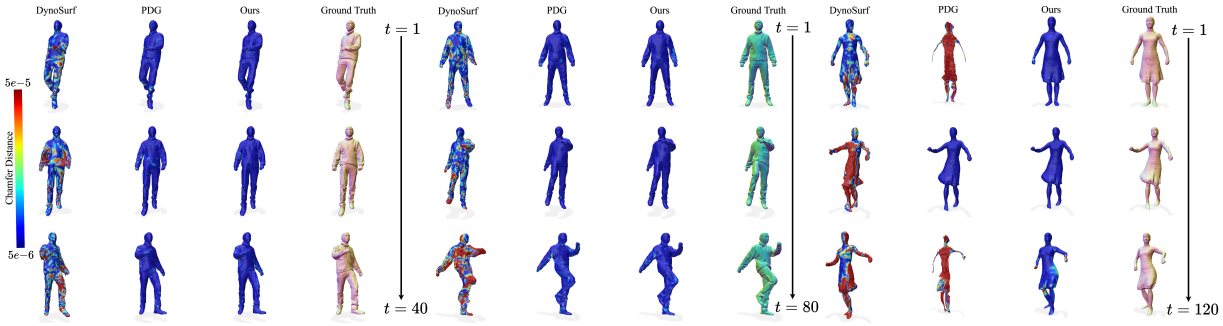

Figure 5. Performance across sequence lengths on AMA. Neu-PiG maintains accuracy and stability as duration increases.

runtime to evaluate computational efficiency.

4.1. Quantitative Comparison

We evaluate Neu-PiG on the DFAUST [7], AMA [67], and DT4D [36] benchmarks, comparing it against both learning-based and training-free baselines (see Tab. 1). Overall, both variants achieve strong performance across all datasets. Although the longer schedule increases computation time, the resulting accuracy gains are minor. On DFAUST, the differences are marginal, while the others benefit only from slight refinement. This behavior aligns with the convergence trends shown in Fig. 4, where Neu-PiG yields high-quality reconstructions within a few seconds.

Across all benchmarks, Neu-PiG achieves the lowest Chamfer Distance and Correspondence Error, as well as the highest Normal Consistency and F-score, while being more than 60× faster than prior training-free approaches. It consistently outperforms all training-free baselines on human motion datasets and maintains high fidelity on the challenging DT4D animal sequences (see Fig. 3), demonstrating reliable generalization across diverse motion types.

Scalability. Maintaining consistent correspondences over long and complex sequences is a persistent challenge in dynamic surface reconstruction, as errors tend to accumulate with sequence length. To evaluate scalability, we split the AMA dataset into sequences ranging from 40 to 120 frames and report the performance in Tab. 2 and Fig. 5. While PDG requires progressively longer runtimes and ultimately fails on very long sequences and DynoSurf shows pronounced degradation, Neu-PiG reconstructs complete sequences in under two minutes while maintaining stable correspondences and high geometric fidelity.

4.2. Ablation Study

We perform ablations to assess the impact of key design choices, including architectural components, loss weighting and temporal frequency encoding. Additional ablation results are provided in the supplementary material.

Table 2. Scalability on AMA sequences of increasing length. Neu-PiG remains stable and efficient, unlike PDG and DynoSurf.

T		CD $[\times 10^{-5}]\downarrow$	NC ↑	F-0.5% ↑	Corr. ↓	Time ↓
40	DynoSurf	4.64	0.857	0.686	0.096	35 min
	PDG	0.66	0.923	0.971	0.042	28 min
	Ours	**0.53**	**0.947**	**0.981**	**0.019**	**47 s**
60	DynoSurf	14.64	0.778	0.474	0.132	48 min
	PDG	0.72	0.919	0.960	0.054	63 min
	Ours	**0.60**	**0.945**	**0.976**	**0.020**	**63 s**
80	DynoSurf	16.32	0.759	0.431	0.163	60 min
	PDG	1.35	0.906	0.931	0.089	93 min
	Ours	**1.04**	**0.940**	**0.959**	**0.023**	**84 s**
100	DynoSurf	26.47	0.706	0.354	0.133	74 min
	PDG	8.59	0.853	0.720	0.091	124 min
	Ours	**0.95**	**0.936**	**0.933**	**0.024**	**97 s**
120	DynoSurf	39.00	0.673	0.284	0.140	96 min
	PDG	30.20	0.788	0.571	0.118	158 min
	Ours	**1.31**	**0.926**	**0.887**	**0.028**	**110 s**

Method Components. We evaluate the contribution of individual components by selectively disabling or modifying parts of Neu-PiG. Specifically, we remove the normal-direction latent encoding, omit preconditioning, replace the multi-resolution grid with the hash encoding of [44], and test a single-resolution variant matching the finest grid parameters. Each modification degrades reconstruction quality, as shown in Tab. 3, confirming that the full combination of components yields the best overall performance. In particular, hash grids are more memory-efficient but sacrifice spatial smoothness, which is important in our setting for coherent latent codes and stable deformations.

Time Frequency Function. We investigate how different time-encoding formulations affect our temporal deformation model. This module maps the normalized timestep $\tilde{t} \in [0, 1]$ into a high-dimensional representation for the deformation network. We compare several time–frequency strategies, each transforming $\tilde{t}$ by a distinct function: The first variant employs a *polynomial basis*, a simple determin-

istic encoding that represents time using monomials,

$$\boldsymbol{\gamma}_{\text{polynomial}}(t) = \left[\tilde{t}^j\right]_{j=1}^{2M}. \tag{14}$$

The second uses a *Gaussian Fourier mapping* with a standard normal distribution, $\boldsymbol{B} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$,

$$\boldsymbol{\gamma}_{\text{gaussian}}(t) = \left[\sin(2\pi \boldsymbol{B}\tilde{t}), \cos(2\pi \boldsymbol{B}\tilde{t})\right]_{j=1}^{M}. \tag{15}$$

Finally we consider a *learned embedding*, obtained from a two-layer MLP ψ_t,

$$\boldsymbol{\gamma}_{\text{learned}}(t) = \psi_t(\tilde{t}) \in \mathbb{R}^{2M}. \tag{16}$$

In addition, we compare these baselines against our proposed time-Fourier encoding (Sec. 3.3). All variants are trained under identical settings, with results in Tab. 4 showing that the Fourier formulation yields the best balance of reconstruction accuracy and temporal coherence.

Stability Function. In Sec. 3.5, we adopt the deformation stability scaling proposed in PDG [25], which stabilizes optimization by gradually increasing the confidence in later timesteps. While effective, this mechanism can also slow convergence when early-frame reconstructions remain inaccurate. To assess its influence, we ablate both components of the stability term, the catch-up variable δ and the temporal weighting ω on sequences of 40 frames, with the results presented in Tab. 5. For the catch-up variable δ, which controls the confidence increase over training epochs $\tilde{e}$, we evaluate four schedules: *Constant* ($\delta = 1$), *linear* ($\delta = 1 - \tilde{e}$), *exponential* ($\delta = \mathrm{e}^{-c\,\tilde{e}}$ with $c = 5$), *interpolated* ($w_{\text{conf}}(t) = (1 - \tilde{e})\,\omega(t) + \tilde{e}$). In addition, we test simplified formulations of the temporal weighting ω within the confidence function $w_{\text{conf}}(t)$. First, we replace the cumulative temporal product with a direct weighting:

$$w_{\text{conf}}(t) = \omega(t)^{\delta} \tag{17}$$

Then, we compare three alternative definitions of ω: *Constant* ($\omega(t) = 1$), *delta-based* ($\omega(t) = 0.5$), *single* ($\omega(t) = 1/(1 + \max(0, \mathrm{cd}_t - \mathrm{cd}_{t_{\text{key}}}))$). It isolates the effect of each stabilization term, allowing us to quantify its impact on convergence speed and reconstruction accuracy.

4.3. Limitations

While our approach achieves fast and stable reconstructions, several practical limitations remain. The method assumes a fixed topology derived from the keyframe mesh, which prevents recovery from incorrect or incomplete initial surfaces. Moreover, the representational capacity of the latent grids and network bounds the sequence length that can be modeled effectively. Finally, as correspondence is inferred implicitly through the Chamfer distance, very large motions or occlusions may reduce reconstruction fidelity, and in rare cases of extreme non-uniform deformation, such as strong local surface shrinkage, may also lead to local face flipping away from the keyframe.

Table 3. Ablation on architectural components. Removing any module reduces accuracy, confirming their complementary roles.

	CD $[\times 10^{-5}]\downarrow$	NC $\uparrow$	F-0.5% $\uparrow$	Corr. $\downarrow$
Hash Encoding	1.23	0.903	0.947	0.045
w/o Preconditioning	0.98	0.955	0.958	0.036
w/o $\boldsymbol{z}_{\text{n}}$	0.91	0.968	0.965	0.036
w/o $\mathcal{L}_{\text{iso}}$	1.03	0.961	0.959	0.044
Single Level $L = 1$	0.98	0.965	0.961	0.036
Ours	**0.87**	**0.969**	**0.968**	**0.034**

Table 4. Comparison of time encoding strategies. Our Fourier encoding achieves the best accuracy and coherence.

	CD $[\times 10^{-5}]\downarrow$	NC $\uparrow$	F-0.5% $\uparrow$	Corr. $\downarrow$
Polynomial	0.46	**0.951**	0.987	0.019
Gaussian	0.77	0.945	0.979	0.046
Learned	0.50	0.950	0.983	0.022
Ours	**0.44**	**0.951**	**0.989**	**0.017**

Table 5. Ablation of stability terms δ and ω on AMA with $T = 40$. Our formulation yields best convergence and accuracy.

		CD $[\times 10^{-5}]\downarrow$	NC $\uparrow$	F-0.5% $\uparrow$	Corr. $\downarrow$
δ	Constant	3.22	0.905	0.899	0.114
	Linear	0.60	0.946	0.971	0.025
	Exponential	0.53	0.946	**0.982**	0.030
	Interpolated	0.64	0.945	0.972	0.046
	Ours	**0.53**	**0.947**	0.981	**0.025**
$\omega(t)$	Constant	3.27	0.905	0.898	0.114
	Delta-based	2.96	0.909	0.908	0.112
	Single	1.15	0.931	0.952	0.098
	Ours	**0.53**	**0.947**	**0.981**	**0.025**

5. Conclusion

We presented Neu-PiG, a fast method for temporally consistent surface reconstruction from unstructured dynamic point clouds. By encoding deformations across all time steps into a preconditioned multi-scale latent grid parameterized by a canonical surface, our method achieved spatially smooth and temporally stable reconstructions without explicit correspondences or learned priors. Experiments demonstrated that our method attains higher fidelity and stability than existing state-of-the-art approaches while running over an order of magnitude faster. We believe Neu-PiG offers a scalable foundation for future research in real-time 4D reconstruction and dynamic scene understanding.

Acknowledgements

This research has been funded by the Federal Ministry of Educa- tion and Research under grant no. 01IS22094A WEST-AI, by the Federal Ministry of Education and Research of Germany as well as the state of North-Rhine Westphalia as part of the Lamarr-Institute for Machine Learning and Artificial Intelligence, by the Ministry of Culture and Science North Rhine-Westphalia under grant number PB22-063A (InVirtuo 4.0: Experimental Research in Virtual Envi- ronments) and by the state of North Rhine-Westphalia as part of the Excellency Start-up Center.NRW (U-BO-GROW) under grant number 03ESCNW18B.

References

[1] Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. Learning to Reconstruct People in Clothing from a Single RGB Camera. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: Shape Completion and Animation of People. *ACM Transactions on Graphics (TOG)*, 24(3), 2005. 2

[3] Jeongmin Bae, Seoha Kim, Youngsik Yun, Hahyun Lee, Gun Bang, and Youngjung Uh. Per-Gaussian Embedding-Based Deformation for Deformable 3D Gaussian Splatting. In *European Conference on Computer Vision (ECCV)*, pages 321–335. Springer, 2024. 3

[4] George Baravdish, Gabriel Eilertsen, Rym Jaroudi, B Tomas Johansson, Lukáš Malỳ, and Jonas Unger. A Hybrid Sobolev Gradient Method for Learning NODEs. In *Operations Research Forum*, page 91. Springer, 2024. 3

[5] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-Garment Net: Learning to Dress 3D People from Images. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[6] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image. In *European Conference on Computer Vision (ECCV)*, 2016. 2

[7] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J Black. Dynamic FAUST: Registering Human Bodies in Notion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6, 7

[8] Aljaz Bozic, Pablo Palafox, Michael Zollhöfer, Angela Dai, Justus Thies, and Matthias Nießner. Neural Non-Rigid Tracking. *Advanced Neural Information Processing Systems (NeurIPS)*, 33, 2020. 2

[9] Aljaz Bozic, Pablo Palafox, Michael Zollhofer, Justus Thies, Angela Dai, and Matthias Nießner. Neural Deformation Graphs for Globally-consistent Non-rigid Reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[10] Ang Cao and Justin Johnson. HexPlane: A Fast Representation for Dynamic Scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 130–141, 2023. 3

[11] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3, 6

[12] Wesley Chang, Xuanda Yang, Yash Belhe, Ravi Ramamoorthi, and Tzu-Mao Li. Spatiotemporal Bilateral Gradient Filtering for Inverse Rendering. In *SIGGRAPH Asia Conference Papers*, 2024. 3

[13] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensoRF: Tensorial Radiance Fields. In *European Conference on Computer Vision (ECCV)*, pages 333–350. Springer, 2022. 3

[14] Honghu Chen, Bo Peng, Yunfan Tao, and Juyong Zhang. D^3-Human: Dynamic Disentangled Digital Human from Monocular Video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10836–10846, 2025. 2

[15] Jie Chen. Graph Neural Preconditioners for Iterative Solutions of Sparse Linear Systems. In *International Conference on Learning Representations (ICLR)*, 2025. 3

[16] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural Ordinary Differential Equations. *Advanced Neural Information Processing Systems (NeurIPS)*, 31, 2018. 2

[17] Namkyeong Cho, Junseung Ryu, and Hyung Ju Hwang. Sobolev Training for Operator Learning. *Journal of Computational Physics*, page 114408, 2025. 3

[18] Sebastian Claici, Mikhail Bessmeltsev, Scott Schaefer, and Justin Solomon. Isometry-Aware Preconditioning for Mesh Parameterization. In *Computer Graphics Forum (CGF)*, 2017. 3

[19] Ilya Eckstein, J-P Pons, Yiying Tong, C-CJ Kuo, and Mathieu Desbrun. Generalized Surface Flows for Mesh Processing. In *Eurographics Symposium on Geometry Processing (SGP)*, 2007. 3

[20] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-Planes: Explicit Radiance Fields in Space, Time, and Appearance. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12479–12488, 2023. 3

[21] Paul Häusner, Ozan Öktem, and Jens Sjölund. Neural incomplete factorization: learning preconditioners for the conjugate gradient method. *Transactions on Machine Learning Research*, 2024. 3

[22] Jingwei Huang, Chiyu Max Jiang, Baiqiang Leng, Bin Wang, and Leonidas Guibas. MeshODE: A Robust and Scalable Framework for Mesh Deformation. *arXiv preprint arXiv:2005.11617*, 2020. 2

[23] Boyan Jiang, Yinda Zhang, Xingkui Wei, Xiangyang Xue, and Yanwei Fu. Learning Compositional Representation for 4D Captures with Neural ODE. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[24] Yucheol Jung, Hyomin Kim, Hyejeong Yoon, and Seungyong Lee. Preconditioned Single-step Transforms for Non-rigid ICP. In *Computer Graphics Forum (CGF)*, 2025. 3

[25] Julian Kaltheuner, Alexander Oebel, Hannah Dröge, Patrick Stotko, and Reinhard Klein. Preconditioned Deformation Grids. *Computer Graphics Forum (CGF)*, 2025. 2, 3, 4, 5, 6, 8

[26] Michael Kazhdan and Hugues Hoppe. Screened Poisson Surface Reconstruction. *ACM Transactions on Graphics (TOG)*, 32(3), 2013. 4

[27] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4), 2023. 2

[28] AOM Kilicsoy, J Liedmann, MA Valdebenito, F-J Barthold, and MGR Faes. Sobolev Neural Network With Residual Weighting as a Surrogate in Linear and Non-Linear Mechanics. *IEEE Access*, 2024. 3

[29] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 6

[30] Shahar Z Kovalsky, Meirav Galun, and Yaron Lipman. Accelerated Quadratic Proxy for Geometric Optimization. *ACM Transactions on Graphics (TOG)*, 35(4), 2016. 3

[31] Dilip Krishnan, Raanan Fattal, and Richard Szeliski. Efficient Preconditioning of Laplacian Matrices for Computer Graphics. *ACM Transactions on Graphics (TOG)*, 32(4), 2013. 3

[32] Jiahui Lei and Kostas Daniilidis. CaDeX: Learning Canonical Deformation Coordinate Space for Dynamic Surface Representation via Neural Homeomorphism. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 6

[33] Runfa Blark Li, Mahdi Shaghaghi, Keito Suzuki, Xinshuang Liu, Varun Moparthi, Bang Du, Walker Curtis, Martin Renschler, Ki Myung Brian Lee, Nikolay Atanasov, and Truong Nguyen. Dynagslam: Real-time gaussian-splatting slam for online rendering, tracking, motion predictions of moving objects in dynamic scenes, 2025. 2

[34] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics (TOG)*, 36 (6), 2017. 1, 2

[35] Yang Li and Tatsuya Harada. Non-rigid Point Cloud Registration with Neural Deformation Pyramid. *Advanced Neural Information Processing Systems (NeurIPS)*, 35, 2022. 2

[36] Yang Li, Hikari Takehara, Takafumi Taketomi, Bo Zheng, and Matthias Nießner. 4DComplete: Non-Rigid Motion Estimation Beyond the Observable Surface. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 6, 7

[37] Yichen Li, Peter Yichen Chen, Tao Du, and Wojciech Matusik. Learning Preconditioners for Conjugate Gradient PDE Solvers. In *International Conference on Machine Learning (ICML)*, 2023. 3

[38] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[39] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural Actor: Neural Free-view Synthesis of Human Actors with Pose Control. *ACM Transactions on Graphics (TOG)*, 40(6), 2021. 2

[40] 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), 2015. 1, 2

[41] Tobias Martin, Pushkar Joshi, Miklós Bergou, and Nathan Carr. Efficient non-linear optimization via multi-scale gradient filtering. In *Computer Graphics Forum (CGF)*, 2013. 3

[42] Hidenobu Matsuki, Gwangbin Bae, and Andrew J Davison. 4dtam: Non-rigid tracking and mapping via dynamic surface gaussians. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2

[43] Marko Mihajlovic, Siwei Zhang, Gen Li, Kaifeng Zhao, Lea Muller, and Siyu Tang. VolumetricSMPL: A Neural Volumetric Body Model for Efficient Interactions, Contacts, and Collisions. In *IEEE International Conference on Computer Vision (ICCV)*, pages 5060–5070, 2025. 2

[44] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics (TOG)*, 41(4):1–15, 2022. 3, 7

[45] JW Neuberger. Steepest descent and differential equations. *Journal of the Mathematical Society of Japan*, 37(2), 1985. 3

[46] Richard A Newcombe, Dieter Fox, and Steven M Seitz. DynamicFusion: Reconstruction and Tracking of Non-rigid Scenes in Real-Time. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[47] Baptiste Nicolet, Alec Jacobson, and Wenzel Jakob. Large Steps in Inverse Rendering of Geometry. *ACM Transactions on Graphics (TOG)*, 40(6), 2021. 3, 5

[48] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy Flow: 4D Reconstruction by Learning Particle Dynamics. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[49] Awais Nizamani, Hamid Laga, Guanjin Wang, Farid Boussaid, Mohammed Bennamoun, and Anuj Srivastava. Dynamic neural surfaces for elastic 4d shape representation and analysis, 2025. 2

[50] Jong Kwon Oh, Hanbaek Lyu, and Hwijae Son. Sobolev acceleration for neural networks. *arXiv preprint arXiv:2509.19773*, 2025. 3

[51] Jinhyung Park, Javier Romero, Shunsuke Saito, Fabian Prada, Takaaki Shiratori, Yichen Xu, Federica Bogo, Shoou-I Yu, Kris Kitani, and Rawal Khirodkar. ATLAS: Decoupling Skeletal and Shape Parameters for Expressive Parametric Human Modeling. In *IEEE International Conference on Computer Vision (ICCV)*, pages 6508–6518, 2025. 2

[52] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable Neural Radiance Fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 2, 3

[53] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and

Michael J Black. Expressive Body Capture: 3D Hands, Face, and Body from a Single Image. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[54] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 3

[55] Robert J Renka. Constructing fair curves and surfaces with a Sobolev gradient method. *Computer Aided Geometric Design*, 21(2), 2004. 3

[56] Robert J Renka and JW Neuberger. Minimal Surfaces and Sobolev Gradients. *SIAM Journal on Scientific Computing*, 16(6), 1995. 3

[57] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Graphics (TOG)*, 36(6), 2017. 1

[58] Alexander Rudikov, Vladimir Fanaskov, Ekaterina Muravleva, Yuri M Laevsky, and Ivan Oseledets. Neural operators meet conjugate gradients: The FCG-NO method for efficient PDE solving. In *International Conference on Machine Learning (ICML)*, 2024. 3

[59] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael J Black. Learning to Regress 3D Face Shape and Expression from an Image without 3D Supervision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[60] Hao Shi, Hongyi Wang, Yifan Jiang, Jie Zhang, and Jingyi Yu. Improved implicit neural representation with fourier reparameterized training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 5

[61] Miroslava Slavcheva, Maximilian Baust, and Slobodan Ilic. SobolevFusion: 3D Reconstruction of Scenes Undergoing Free Non-rigid Motion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[62] Robert W Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. *ACM Transactions on Graphics (TOG)*, 26(3), 2007. 2

[63] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains, 2020. 4

[64] Jiapeng Tang, Dan Xu, Kui Jia, and Lei Zhang. Learning Parallel Dense Correspondence from Spatio-Temporal Descriptors for Efficient and Robust 4D Reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[65] Garvita Tiwari, Bharat Lal Bhatnagar, Tony Tung, and Gerard Pons-Moll. SIZER: A Dataset and Model for Parsing 3D Clothing and Learning Size Sensitive 3D Clothing. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[66] Vladislav Trifonov, Alexander Rudikov, Oleg Iliev, Yuri M Laevsky, Ivan Oseledets, and Ekaterina Muravleva. Learning from Linear Algebra: A Graph Neural Network Approach to Preconditioner Design for Conjugate Gradient Solvers. *arXiv preprint arXiv:2405.15557*, 2024. 3

[67] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. Articulated Mesh Animation from Multi-view Silhouettes. *ACM Transactions on Graphics (TOG)*, 27(3), 2008. 6, 7

[68] Lizhen Wang, Zhiyuan Chen, Tao Yu, Chenguang Ma, Liang Li, and Yebin Liu. FaceVerse: a Fine-grained and Detail-controllable 3D Face Morphable Model from a Hybrid Dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[69] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. GHUM & GHUML: Generative 3D Human Shape and Articulated Pose Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[70] Zhenyu Xu, Yihe Ma, Zhihao Chen, Rui Yang, et al. Signal processing for implicit neural representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 5

[71] Yuxin Yao, Siyu Ren, Junhui Hou, Zhi Deng, Juyong Zhang, and Wenping Wang. DynoSurf: Neural Deformation-based Temporally Consistent Dynamic Surface Reconstruction. In *European Conference on Computer Vision (ECCV)*, 2024. 2, 5, 6

[72] Tarun Yenamandra, Ayush Tewari, Florian Bernard, Hans-Peter Seidel, Mohamed Elgharib, Daniel Cremers, and Christian Theobalt. i3DMM: Deep Implicit 3D Morphable Model of Human Heads. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[73] Jianhao Zheng, Zihan Zhu, Valentin Bieri, Marc Pollefeys, Songyou Peng, and Iro Armeni. Wildgs-slam: Monocular gaussian splatting slam in dynamic environments. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2

[74] Jiaxuan Zhu and Hao Tang. Dynamic Scene Reconstruction: Recent Advance in Real-time Rendering and Streaming. *arXiv preprint arXiv:2503.08166*, 2025. 3

Neu-PiG: Neural Preconditioned Grids for Fast Dynamic Surface Reconstruction on Long Sequences

Supplementary Material

In this supplementary document, we provide additional experiments, analyses, and visualizations that complement the results presented in the main paper. Specifically, we extend the evaluation of Neu-PiG with further quantitative and qualitative results to validate the impact of grid design choices, preconditioning strength, and input conditions.

In Figs. 9 and 10, we present extended visual results on a broader range of datasets, demonstrating that Neu-PiG maintains high reconstruction quality and temporal consistency across diverse motion types, outperforming existing state-of-the-art approaches in both accuracy and stability.

5.1. Latent Grid Design

We analyze the design choices of our preconditioned multi-resolution grid, focusing on the impact of the number of levels and the smoothness introduced by preconditioning.

Grid Levels. In Tab. 6, we evaluate the influence of the number of grid levels, described in Sec. 3.2, on reconstruction quality using the AMA dataset. We keep our default configuration and vary only the total number of levels L. The results show that even with a small number of levels, and thus reduced latent capacity, Neu-PiG achieves surprisingly strong reconstructions. However, excessively increasing the grid resolution leads to performance degradation due to overfitting, as shown in Fig. 6.

Complementing this quantitative analysis, Fig. 7 provides a qualitative view of the learned multi-resolution decomposition. Coarser levels capture low-frequency global motion, whereas finer levels recover localized high-frequency deformations, visualized using a local rigidity proxy based on Kabsch alignment.

Smoothness Weights. We investigate the effect of the smoothness parameter λ in our grid preconditioning, introduced in Sec. 3.5, on reconstruction performance. To access its influence, we vary the base smoothness value λ^1 to both smaller and larger magnitudes, while using the same increase per level as in the default configuration. The results in Tab. 7 on the AMA dataset indicate that Neu-PiG is largely robust to moderate changes in smoothness, while extreme values, either excessively high or entirely absent, lead to a clear degradation in reconstruction quality. We further visualize the learned latent spaces in Fig. 8 by performing a PCA analysis and plotting the first principal component.

Table 6. Ablation study on the number of grid levels in $\mathcal{G}_\mathrm{p}$, evaluated on the AMA dataset. Increasing the number of levels improves reconstruction quality up to $L = 8$, after which performance slightly degrades due to overfitting.

	CD $[\times 10^{-5}] \downarrow$	NC $\uparrow$	F-0.5% $\uparrow$	Corr. $\downarrow$
$L = 1$	1.96	0.850	0.810	0.063
$L = 2$	1.76	0.859	0.841	0.054
$L = 4$	0.54	0.945	0.980	0.023
$L = 6$	0.46	0.949	0.987	0.018
Ours ($L = 8$)	**0.44**	**0.951**	**0.989**	**0.017**
$L = 10$	0.45	**0.951**	**0.989**	0.021
$L = 12$	0.46	0.948	0.988	0.024

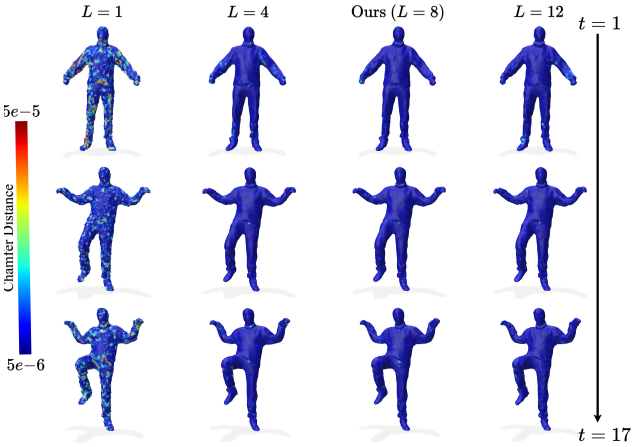

Figure 6. Qualitative analysis of the effect of the number of grid levels during optimization on reconstruction quality. Neu-PiG achieves accurate and temporally stable reconstructions even with few grid levels, while excessively increasing resolution leads to mild overfitting and loss of fine structural detail.

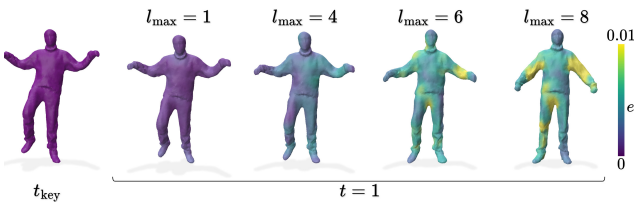

Figure 7. After optimization, we progressively restrict Neu-PiG to $l_{\max}$ grid levels and visualize the deformations, colored by a rigidity proxy. It shows that coarse levels capture smooth, global motion, whereas finer levels contribute localized, high-frequency detail, supporting the role of the multi-resolution latent grid.

Table 7. Ablation study on the smoothness weight λ of the base grid level ($l = 1$), evaluated on the AMA dataset. Neu-PiG remains robust under moderate variations of λ, while extreme values reduce reconstruction quality.

	CD $[\times 10^{-5}] \downarrow$	NC $\uparrow$	F-0.5% $\uparrow$	Corr. $\downarrow$
$\lambda^1 = 0$	0.58	0.928	0.977	0.022
$\lambda^1 = 0.08$	0.45	0.950	0.988	0.020
Ours ($\lambda^1 = 0.4$)	**0.44**	**0.951**	**0.989**	**0.017**
$\lambda^1 = 2$	0.46	0.948	0.986	0.021
$\lambda^1 = 10$	0.50	0.946	0.983	0.025

5.2. Input Conditions

We additionally study how different input conditions, such as varying number of input points and noise levels, affect the reconstruction performance of Neu-PiG.

Noise. We evaluate the robustness of Neu-PiG to noisy input data by adding Gaussian noise with varying strengths to the input points based on the size of the bounding box diagonal. As shown in Tab. 8, the method remains stable for noise levels up to 0.25–0.5%, after which the performance degrades as noise is captured in the surface geometry.

Point Cloud Resolution. We evaluate the reconstruction performance across varying input point cloud resolutions, focusing on the generalization behavior in comparison to DynoSurf, which similarly employs MLP-based transformation modeling. As shown in Tab. 9, DynoSurf exhibits a noticeable drop in performance with increasing target resolution, indicating limited scalability. In contrast, our method demonstrates consistent improvements with higher input densities, closely matching the behavior of the direct optimization approach PDG, and achieving superior reconstruction quality at all tested resolutions. The difference compared to DynoSurf is particularly noteworthy: although both methods employ MLPs to model per-point transformations, DynoSurf fails to benefit from denser point clouds. This robustness stems from our hierarchical latent representation, which efficiently captures both coarse and fine geometric structures.

5.3. Neural Model

We further analyze the design of the neural components in Neu-PiG and evaluate how different architectural choices of the decoder influence reconstruction quality and temporal stability.

Time Deformation Model. We investigate how modifications to the temporal deformation network influence reconstruction performance. This component models temporal

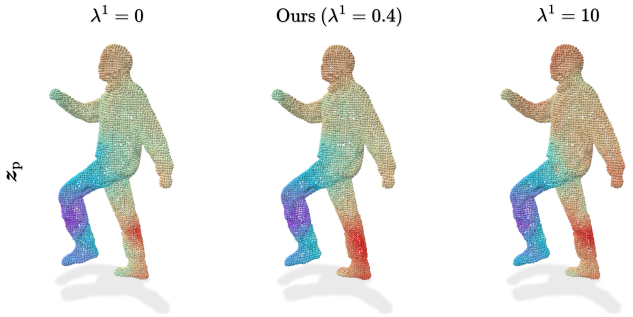

Figure 8. Visualization of the learned latent space for different smoothness weights. We perform a PCA analysis on the latent vectors and display the first principal component as a spatial field.

Table 8. Ablation study on robustness to input noise with varying strengths based on the bounding box diagonal. Neu-PiG maintains stable reconstruction quality up to 0.5% noise, after which performance degrades as noise is captured in the surface geometry.

Noise	CD $[\times 10^{-5}] \downarrow$	NC $\uparrow$	F-0.5% $\uparrow$	Corr. $\downarrow$
0.25%	0.69	0.925	0.967	0.021
0.5%	1.34	0.866	0.853	0.026
1%	4.45	0.681	0.540	0.049
2%	21.71	0.517	0.247	0.086

shape evolution conditioned on the latent embeddings introduced in Sec. 3.2. We systematically vary the dimensionality of the positional and normal latent vectors $\boldsymbol{z}_\mathrm{p}$ and $\boldsymbol{z}_\mathrm{n}$, as well as the frequency range of the time encoding $\boldsymbol{\gamma}$. In addition, we assess the impact of network capacity by adjusting the number of hidden units per layer. As shown in Tab. 10, Neu-PiG remains stable across most configurations. However, excessive reduction in model capacity leads to lower reconstruction quality, while increasing the dimensionality of $\boldsymbol{z}_\mathrm{p}$ or $\boldsymbol{\gamma}$ causes the network to overfit more easily.

Rotation Modeling. Our deformation model predicts parameters that are mapped to 3D rotations and deformations, as described in Sec. 3.4. Several representations exist for mapping parameters to rotations, including quaternions, the Cayley transform, and the exponential map, and their choice can affect optimization stability, convergence behavior, and final reconstruction accuracy. To study this influence, we run our reconstruction using each representation under identical settings and compare performance over different amounts of maximal optimization epochs. The resulting metrics are summarized in Tab. 11. The results show that while all parameterizations achieve comparable accuracy, quaternions converge slightly faster and yield marginally better performance during early training.

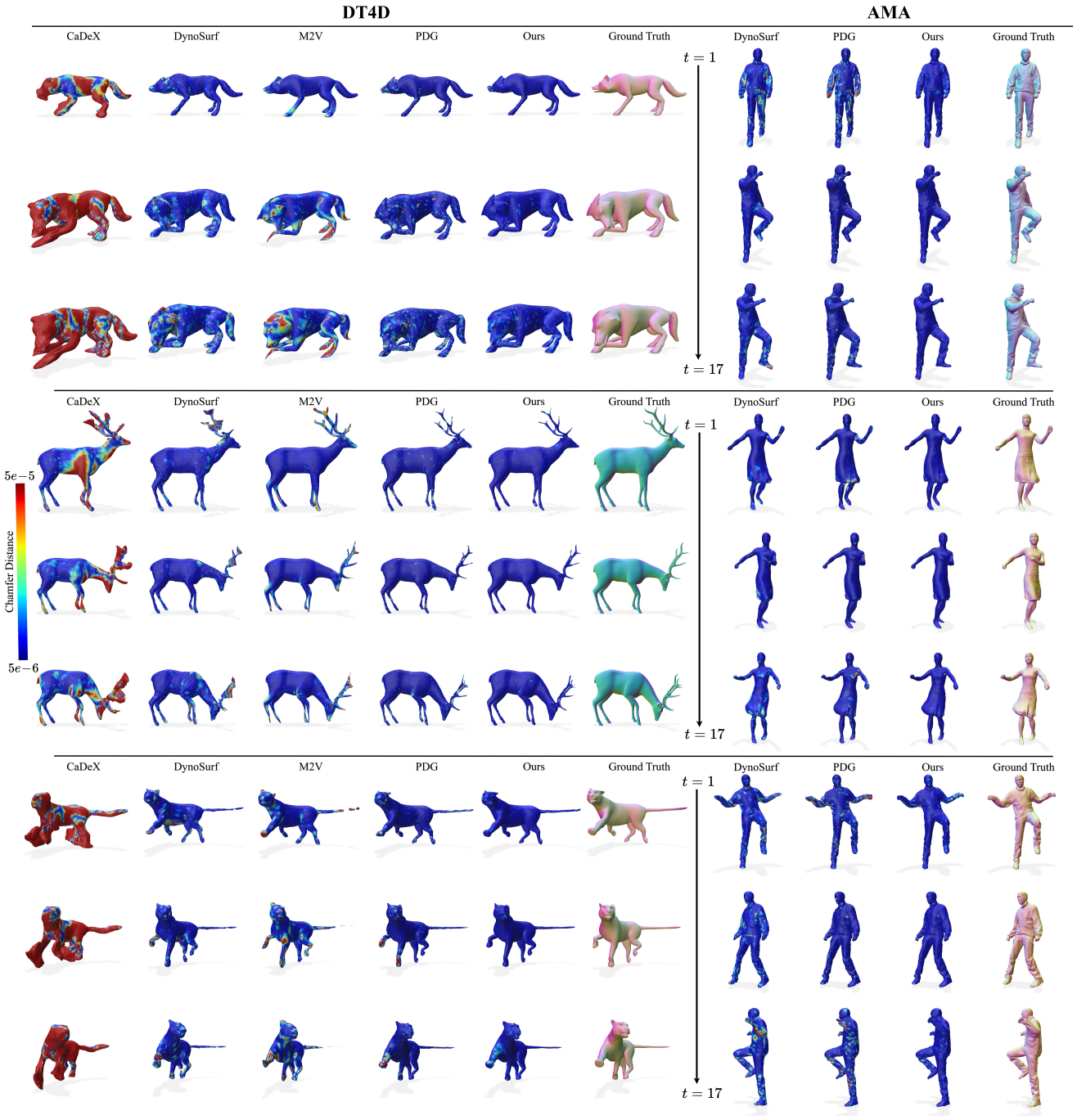

Figure 9. Extended qualitative results across diverse human and animal sequences. Neu-PiG consistently reconstructs temporally stable and detailed surfaces, even under large non-rigid deformations and challenging motion dynamics.

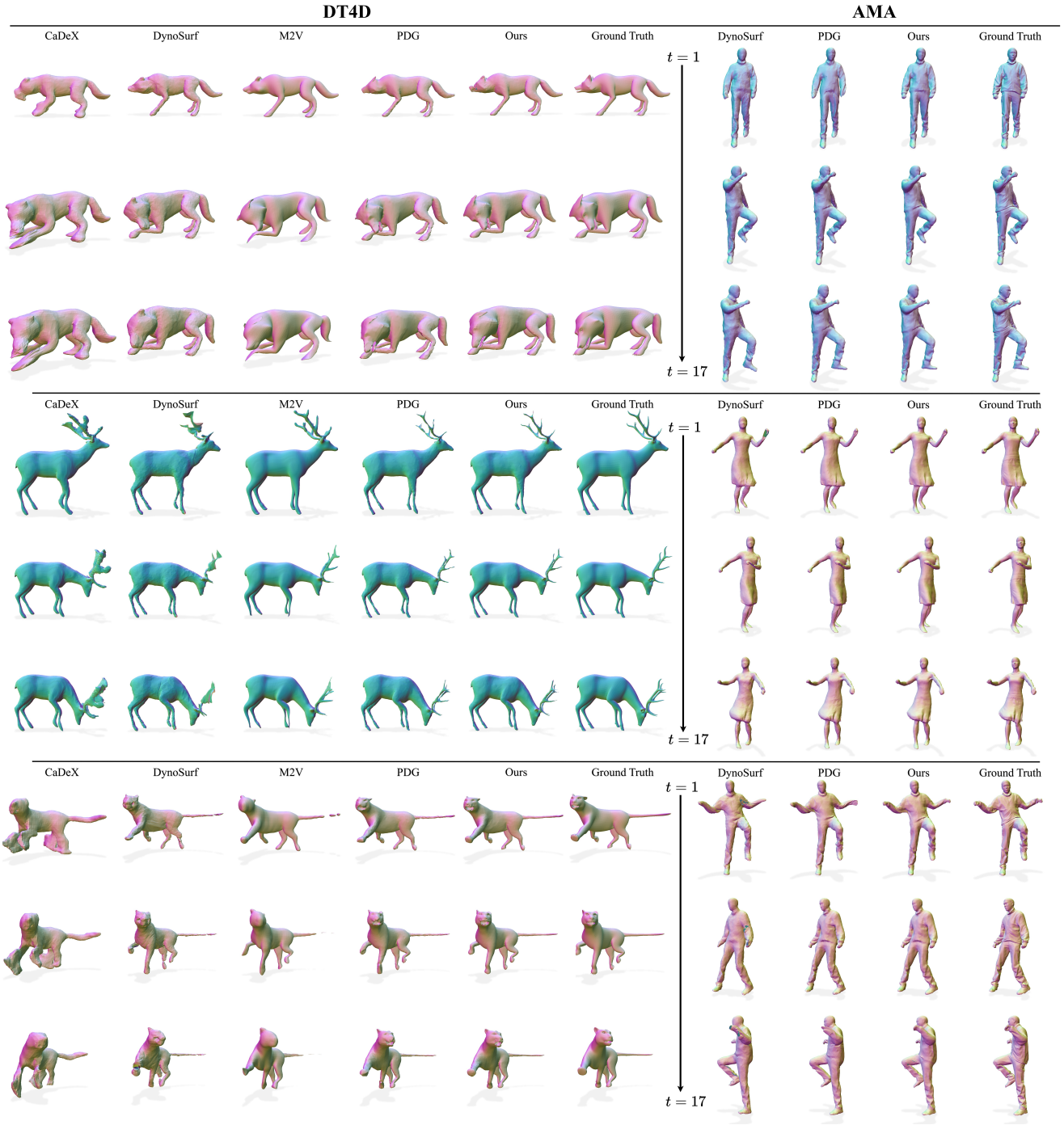

Figure 10. Extended visual results across diverse human and animal sequences. The surface normals of the deformed meshes are shown in comparison to ground truth surface normals.

Table 9. Reconstruction performance at varying input point cloud resolutions. Neu-PiG consistently improves with higher input densities, closely matching the behavior of PDG while maintaining superior quality across all settings. In contrast, DynoSurf shows limited scalability and degrading performance at higher resolutions.

$\lvert\mathcal{P}_t\rvert$		CD $[\times 10^{-5}]\downarrow$	NC $\uparrow$	F-0.5% $\uparrow$	Corr. $\downarrow$
2500	DynoSurf	1.56	0.896	0.862	0.041
	PDG	0.79	0.922	0.948	0.037
	Ours	**0.75**	**0.930**	**0.953**	**0.022**
5000	DynoSurf	1.01	0.918	0.921	0.044
	PDG	0.47	0.939	0.985	0.030
	Ours	**0.44**	**0.950**	**0.988**	**0.018**
10000	DynoSurf	1.28	0.906	0.897	0.037
	PDG	0.40	0.950	0.993	0.027
	Ours	**0.37**	**0.960**	**0.995**	**0.015**
20000	DynoSurf	1.45	0.902	0.887	0.040
	PDG	0.37	0.959	0.996	0.023
	Ours	**0.34**	**0.968**	**0.997**	**0.014**

Table 10. Ablation study on the temporal deformation network architecture, evaluated on the AMA dataset with 40 time steps per sequence. We vary latent dimensionality, number of time-encoding frequencies, and network capacity to assess their effect on reconstruction quality. Neu-PiG remains stable under most configurations, though extreme reductions in capacity or excessive latent dimensionality lead to degraded performance or overfitting. Our default configuration is shown in the last row.

$\lvert\psi\rvert$	$\lvert\boldsymbol{z}_\mathrm{p}\rvert$	$\lvert\boldsymbol{z}_\mathrm{n}\rvert$	$\lvert\boldsymbol{\gamma}\rvert$	CD $[\times 10^{-5}]\downarrow$	NC $\uparrow$	F-0.5% $\uparrow$	Corr. $\downarrow$
512	30	2	2	0.63	0.945	0.970	0.027
512	30	2	32	0.64	0.943	0.973	0.040
512	30	1	8	0.55	0.946	0.979	0.026
512	30	8	8	0.57	**0.947**	0.978	0.026
512	8	2	8	0.61	0.945	0.975	0.023
512	120	2	8	0.59	**0.947**	0.978	0.032
128	30	2	8	0.76	0.938	0.959	0.045
2048	30	2	8	0.54	0.948	0.980	0.021
512	30	2	8	**0.53**	**0.947**	**0.981**	**0.019**

Table 11. Comparison of different rotation parameterizations predicted by the deformation model. We evaluate quaternions, the Cayley transform, and the exponential map under identical optimization conditions. All representations achieve similar accuracy, while using quaternions leads to slightly faster convergence and improves performance with fewer epochs.

Epochs		CD $[\times 10^{-5}]\downarrow$	NC $\uparrow$	F-0.5% $\uparrow$	Corr. $\downarrow$
75	Cayley	1.96	0.921	0.866	0.047
	Exponential	1.62	**0.924**	**0.886**	**0.043**
	Quaternions	**1.59**	0.923	0.884	0.044
125	Cayley	0.72	0.940	0.955	0.029
	Exponential	0.68	0.941	0.961	**0.027**
	Quaternions	**0.65**	**0.942**	**0.965**	0.028
250	Cayley	0.52	0.947	0.978	0.021
	Exponential	0.54	**0.948**	0.976	0.021
	Quaternions	**0.50**	**0.948**	**0.982**	**0.020**
500	Cayley	0.47	**0.950**	0.985	0.019
	Exponential	0.47	**0.950**	0.985	**0.018**
	Quaternions	**0.46**	**0.950**	**0.986**	**0.018**
1000	Cayley	0.46	0.950	0.987	0.019
	Exponential	0.45	**0.951**	**0.988**	**0.017**
	Quaternions	**0.44**	**0.951**	**0.988**	**0.017**