

DynaTok: Token-Based 4D Reconstruction from Partial Point Clouds

Weirong Chen^{1 2 3} Keisuke Tateno² Hidenobu Matsuki² Michael Niemeyer² Daniel Cremers^{1 3}
Federico Tombari^{1 2 3}

Abstract

We address 4D reconstruction from partial point cloud sequences, where depth-sensor observations are incomplete, unordered, and lack explicit temporal correspondences. This geometry-only setting is challenging due to missing observations and ambiguous dynamics. While recent progress has largely relied on image-based methods, existing point-based approaches typically focus on single objects, assume relatively complete inputs, or require explicit correspondences. To address these limitations, we propose DynaTok, a point-based framework for correspondence-free 4D reconstruction from partial point cloud sequences without images. DynaTok encodes frames into compact latent tokens, aggregates incomplete observations over time with a Transformer-based spatiotemporal encoder, and decouples geometry and motion through residual tokens in a unified model. A flow-matching decoder then reconstructs complete, temporally consistent 4D point-cloud sequences conditioned on the latent tokens. Experiments on object- and scene-level benchmarks demonstrate improved reconstruction quality and temporal coherence from partial point cloud observations. Project page: <https://wrchen530.github.io/dynatok/>.

1. Introduction

Reconstructing dynamic 3D scenes over time, which is commonly referred to as 4D reconstruction, is a fundamental problem in computer vision with broad applications in robotics (Rajič et al., 2025; Huang et al., 2026), augmented reality (Wang et al., 2025c), and scene understanding (Zhang et al., 2025b; Zhou et al., 2024). Recent advances have been largely driven by methods that operate on dense visual observations, such as images or videos (Zhang et al., 2025c; Feng et al., 2025; Li et al., 2025b; Ren et al., 2024a; Wu

¹TU Munich ²Google ³Munich Center for Machine Learning. Correspondence to: Weirong Chen <weirong.chen@tum.de>.

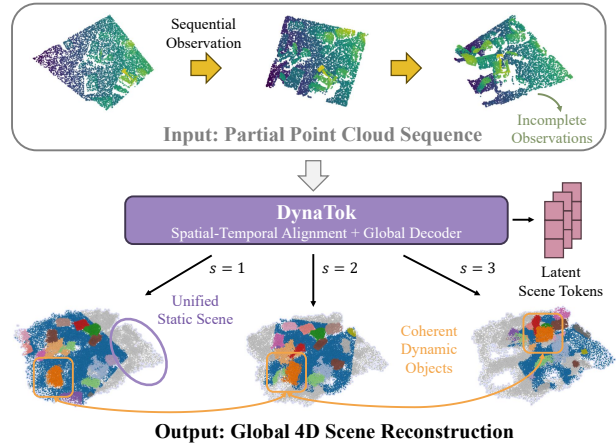


Figure 1. **DynaTok** reconstructs coherent 4D scenes from partial, unordered, and correspondence-free point cloud sequences. It temporally aggregates incomplete observations via spatiotemporal alignment and a global decoder, enabling consistent recovery of static background structure and dynamic objects even when large regions are unobserved. For visualization, we show the global 4D scene reconstruction at each time step: the colored region corresponds to the area observed in the partial input at time s , while the white regions are recovered through the temporal fusion. **Colors and segmentation labels are only used for visualization and are not provided to the model.**

et al., 2025), enabling high-fidelity reconstruction of dynamic geometry under controlled settings. However, these approaches typically assume visual input with rich texture and appearance cues. In contrast, point clouds are the native output of depth sensors and provide a scalable, geometry-centric representation without relying on appearance cues or surface connectivity. Despite their suitability for real-world sensing, existing point-based learning methods are largely restricted to static scenes or object-level dynamics and struggle to scale to complex, open-world dynamic environments.

In many practical scenarios, especially those involving depth sensors or multi-view geometry pipelines, the available observations at test time take the form of *partial point clouds*. These point clouds are incomplete due to limited sensor coverage, occlusions, and viewpoint changes, are unordered, and lack explicit point-to-point correspondence across time. Under this setting, 4D reconstruction faces several fundamental challenges. First, individual frames

may entirely miss large regions of the scene, making per-frame reconstruction fundamentally insufficient, particularly for dynamic objects. Second, without image appearance or tracking cues, it is inherently ambiguous to distinguish static scene structure from dynamic objects, as missing points may result from occlusion, motion, or viewpoint changes. Third, since the identity of points is not preserved across frames, temporal correspondence cannot be assumed, and naive temporal fusion often leads to inconsistent geometry and motion. Together, these challenges make coherent 4D reconstruction from partial point cloud sequences fundamentally difficult.

In this work, we study 4D reconstruction from partial, correspondence-free point cloud inputs, with a particular focus on how temporal aggregation across frames can compensate for missing geometry. Rather than assuming that each time step provides a complete or dense observation, our goal is to recover a temporally consistent 4D representation even when objects or regions are completely unobserved in individual frames (see Figure 1). This setting poses unique challenges that are not adequately addressed by existing methods, which often rely on complete inputs or focus on single-object dynamics (Cao et al., 2024; Niemeyer et al., 2019; Tang et al., 2021; Lei & Daniilidis, 2022). Moreover, many of these methods require watertight mesh supervision, which limits their applicability to real-world scanned data.

To address this challenge, we propose DynaTok, a token-based framework for temporally aggregating partial point cloud observations. The key idea is to encode sparse and irregular point cloud inputs into a compact set of temporally aligned latent tokens shared across frames. By aggregating complementary information over time, these tokens enable robust 4D reconstruction from incomplete observations. We further decouple the latent space into geometry and motion tokens, enabling disentangled modeling of shape and dynamics. Unlike prior approaches that rely on two separate networks for shape reconstruction and deformation modeling (Cao et al., 2024; Jiang et al., 2026; Zhang et al., 2025a), our residual token formulation achieves explicit geometry–motion decoupling within a single unified model. Conditioned on the aggregated latent tokens, DynaTok reconstructs coherent 4D scene representations using a point flow-matching decoder (Lipman et al., 2023), which produces temporally consistent point clouds and can be trained using only point cloud supervision. This design avoids reliance on watertight meshes or explicit correspondences and provides a unified framework for both object-level and scene-level 4D reconstruction.

In summary, our contributions are threefold. (i) We formulate the practical yet underexplored problem of global 4D reconstruction from partial, unordered, and correspondence-free point cloud sequences. (ii) We propose a unified token-based 4D representation and pipeline that enables

correspondence-free temporal aggregation and geometry–motion decomposition under point-cloud-only supervision, without requiring watertight meshes or canonical templates. (iii) We validate the framework on both object- and scene-level dynamic benchmarks through qualitative and quantitative evaluations.

2. Related Work

2.1. 3D Generation and Latent Representations

3D generative models aim to learn compact latent representations of shapes and scenes. Early work employs VAEs and GANs to model distributions over 3D geometry (Wu et al., 2016; Zhu et al., 2018; Pavllo et al., 2020; Kosiorek et al., 2021; Li et al., 2021a; Chan et al., 2022; Gao et al., 2022), while recent advances leverage diffusion models to generate high-fidelity 3D content. One line of work combines diffusion with neural rendering, generating multi-view images that are reconstructed into 3D via differentiable rendering (Liu et al., 2023; Gao et al., 2024; Szymanowicz et al., 2025; Tang et al., 2024; Hong et al., 2024; Voleti et al., 2024; Chen et al., 2024).

Another direction directly generates 3D representations such as point clouds or meshes using diffusion (Zhang et al., 2023; Li et al., 2025a; Jun & Nichol, 2023; Ren et al., 2024b; Vahdat et al., 2022; Wu et al., 2024b; Zhang et al., 2024b; Chang et al., 2024; Zheng et al., 2023b; Gupta et al., 2023; Xiang et al., 2025a). Closely related, NOVA3R (Chen et al., 2026) learns a global latent representation for static 3D scenes in a two-stage framework and supports image-conditioned reconstruction. Despite these advances, most 3D generative models are designed for static shapes or bounded scenes, with limited support for temporally evolving 4D geometry from partial, correspondence-free inputs. In contrast, we study this latent-representation perspective in the dynamic 4D setting, where the input is a partial, correspondence-free point-cloud sequence.

2.2. 4D Reconstruction from Video

Reconstructing 4D scenes from video aims to recover temporally coherent 3D geometry and motion from visual observations, providing a dynamic representation of objects or scenes over time (Wang et al., 2025c; Li et al., 2025b; Zhang et al., 2025c; Feng et al., 2025; Chen et al., 2025; Zhang et al., 2026; Qian et al., 2026; Karhade et al., 2025). Recent advances in image-based 4D reconstruction are driven by generative methods that model dynamic geometry from multi-view video, typically focusing on single objects, explicit surface representations (e.g., meshes or canonical templates), and dense visual supervision to decouple shape and motion. Early works incorporate generative or physical priors to regularize motion during optimization (Jiang

et al., 2024; Ren et al., 2023; Sang et al., 2025), while more recent approaches leverage diffusion models to directly recover time-varying geometry (Ren et al., 2024a; Liang et al., 2024; Pan et al., 2024; Xie et al., 2025; Zhang et al., 2024a; Jiang et al., 2026; Zhang et al., 2025a). However, these methods largely assume image inputs, single-object scenes, or explicit correspondences at inference, limiting their applicability to our setting of partial, correspondence-free point clouds.

Beyond 4D object generation, scene-level 4D reconstruction from images is commonly formulated as recovering dense point maps (Kopf et al., 2021; Zhang et al., 2022; 2025c; Li et al., 2025b; Wimbauer et al., 2025) or Gaussian Splats (Wu et al., 2024a; Wang et al., 2025c; Yang et al., 2024b; Matsuki et al., 2025) per time step. These methods exploit multi-view consistency, optical flow, and appearance cues to recover dynamic scenes, with recent work further incorporating dense correspondence and scene flow for more complete 4D reconstruction (Sucar et al., 2025; Wang et al., 2025e; Karhade et al., 2025; Feng et al., 2025; Zhang et al., 2026). By relying on dense image observations and rich appearance cues, these methods address a different input regime from the geometry-only point-cloud setting considered here.

2.3. 4D Reconstruction from Point Clouds

In contrast to image-based methods, recent work has explored 4D reconstruction from point cloud observations to address occlusions, limited coverage, and viewpoint changes in real-world systems (Zheng et al., 2023a). Such point clouds may be obtained from depth sensors (Ranftl et al., 2020; Yang et al., 2024a; Yin et al., 2023), LiDAR (Huang et al., 2023a;b), or sparse geometric cues such as multi-view anchor points (Dong et al., 2021) and motion capture systems (Alexiadis et al., 2016). The core challenge is to aggregate incomplete geometry over time to recover coherent 4D structure. Classical fusion and SLAM-based methods show that temporal aggregation substantially improves static reconstruction under sparse observations (Izadi et al., 2011; Dai et al., 2017), and recent learning-based approaches extend these principles to dynamic point cloud sequences.

A common approach to 4D reconstruction disentangles static structure from dynamic deformation, often via canonical-to-deformed formulations such as Occupancy Flow (Niemeyer et al., 2019) and related correspondence- or deformation-based object-level methods (Tang et al., 2021; Lei & Daniilidis, 2022; Cao et al., 2024; Wang et al., 2025b). Among these, Motion2VecSets (Cao et al., 2024) is most closely related to our work, but it infers a canonical shape from only the first frame and generates later frames as deformations of this fixed reference, whereas our one-stage alignment module jointly aggregates geometry across frames to better handle incomplete first-frame observations and

correspondence-free partial point-cloud sequences. Scene-level approaches adopt explicit grid or implicit representations for scalability (Huang et al., 2025; Zhong et al., 2024), but usually rely on relatively complete inputs such as LiDAR sweeps. In contrast, we address 4D reconstruction from partial, correspondence-free point clouds derived from depth maps with severe occlusions and viewpoint changes.

3. Method

Our goal is to reconstruct a coherent 4D scene representation from a sequence of partial, unordered, and correspondence-free point clouds. The main challenge lies in temporal aggregation: integrating incomplete and inconsistent observations across time to recover missing geometry and coherent motion in dynamic scenes. We address this challenge by proposing a 4D point tokenizer that embeds input point clouds into a compact latent space, enabling the recovery of global scene geometry at each time step. In the following, we first formalize the problem setting (Section 3.1), then describe the latent temporal aggregation mechanism (Section 3.2) and the conditional reconstruction process (Section 3.3). An overview of the pipeline is shown in Figure 2.

3.1. Problem Formulation

We consider the task of reconstructing a complete and temporally consistent 4D scene from a sequence of partial point cloud observations. The input is a sequence of dynamic point clouds $\mathcal{X} = \{\mathbf{X}_s\}_{s=1}^S$, where $\mathbf{X}_s \in \mathbb{R}^{N_{in} \times 3}$ denotes the 3D points observed at time step s . Each \mathbf{X}_s can be partial, unordered, and correspondence-free, reflecting realistic observations from depth sensors or multi-view reconstruction pipelines. The target is to recover a temporally consistent and complete 4D scene representation $\mathcal{Y} = \{\mathbf{Y}_s\}_{s=1}^S$, $\mathbf{Y}_s \in \mathbb{R}^{N_{out} \times 3}$, where each \mathbf{Y}_s represents the complete scene geometry at time step s . Importantly, the reconstruction at each time step must be consistent with the underlying scene motion, even when the corresponding geometry is unobserved in the input.

This problem setting is substantially more challenging than per-frame completion (Yu et al., 2021; Huang et al., 2020) or 4D autoencoders with complete geometry (Zhang et al., 2025a; Wang et al., 2024). Successful reconstruction requires the model to (i) aggregate incomplete and inconsistent observations across time, (ii) implicitly distinguish static scene structure from dynamic objects without correspondence or tracking cues, and (iii) maintain temporal coherence of the reconstructed geometry across frames. These challenges cannot be addressed by naive temporal fusion or frame-wise reconstruction, and instead demand structured mechanisms for effective temporal aggregation under partial observations (see Figure 3).

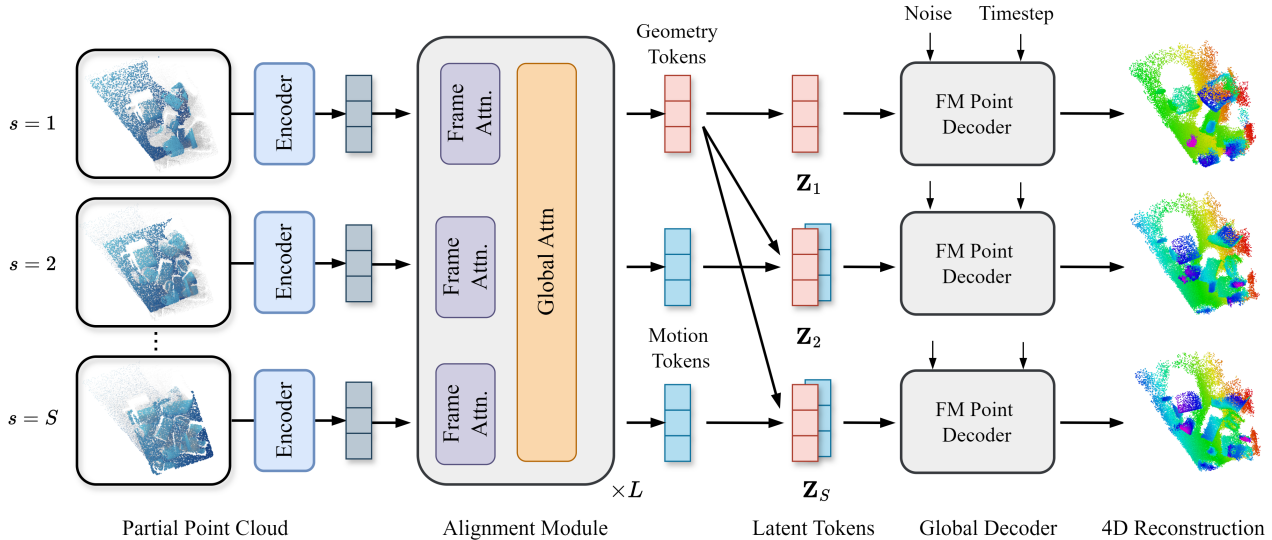


Figure 2. Overview of the DynaTok Pipeline. Given a sequence of incomplete point clouds, we extract per-frame point tokens and process them with a spatiotemporal alignment module to integrate temporal information. Using a residual token design, we obtain geometry tokens defining the canonical space at the reference frame ($s = 1$) and motion tokens for subsequent frames ($s > 1$), enabling joint shape and motion modeling without introducing an additional network. The resulting latent token \mathbf{Z}_s encodes 4D information and is fed into the flow-matching decoder to recover the global geometry at time step s . All components are trained from scratch.

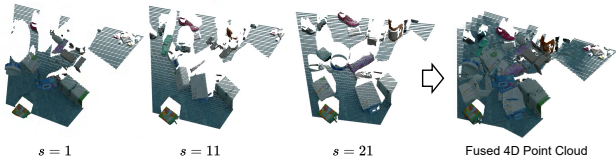


Figure 3. Illustration of 4D Fusion and Completion Task. Given the partial observation at each time step, the goal is to model the global dynamic scene.

3.2. Latent Temporal Aggregation

Per-Frame Token Extraction. To enable temporal aggregation over partial and correspondence-free point clouds, we first convert each input point cloud into a compact and structured latent representation. Given an input point cloud $\mathbf{X}_s \in \mathbb{R}^{N_{in} \times 3}$ at time step s , we extract a fixed number of latent tokens that represent the geometric content of the frame in a permutation-invariant manner.

Specifically, we select M point queries as anchors using Farthest Point Sampling (FPS). Inspired by recent geometry autoencoders (Li et al., 2025a; Chen et al., 2026), these queries attend to the full input point cloud through a cross-attention layer, where the raw point positions serve as keys and values. Prior to attention, 3D point coordinates are embedded into features using Fourier features to enhance spatial expressiveness (Li et al., 2025a). The resulting position features are further refined using self-attention layers to capture local geometric context. The output is a set of per-frame latent tokens $\mathbf{F}_s \in \mathbb{R}^{M \times D_z}$, providing a compact and regularized representation of each partial point cloud.

This tokenization step reduces the variability of raw point cloud inputs while preserving essential geometric structure, enabling efficient and consistent processing across time.

Spatial-Temporal Alignment. Given the per-frame tokens $\{\mathbf{F}_s\}_{s=1}^S$, we perform temporal aggregation using a transformer-based architecture (Wang et al., 2025a) that jointly processes tokens across time. The goal is to integrate partial observations from different frames and aggregate geometric information into a coherent latent representation.

Each transformer layer alternates between per-frame self-attention and cross-frame global attention. Per-frame attention preserves spatial coherence within a frame, and the global attention allows tokens to propagate information across time in the latent space. Through this interleaved attention mechanism, geometric evidence observed in one frame can influence the latent representation of other frames, enabling recovery of missing regions under partial observations.

Importantly, this aggregation operates entirely at the token level and does not assume explicit point-to-point correspondence across frames. Temporal consistency emerges implicitly through shared attention and repeated exposure to complementary observations. To encode spatial relationships, we extend rotary positional embeddings (Su et al., 2024) to 3D using the token query positions, allowing the model to reason about relative geometry during aggregation.

Latent Residual Decomposition. To model temporal variation in the aggregated latent representations, we adopt a simple residual formulation across time. We treat the latent tokens corresponding to the first frame ($s = 1$) as a reference representation, which serves as a canonical anchor for the sequence. This reference does not introduce a separate latent space or specialized tokens; all frames are represented in the same latent space.

Formally, we represent the latent tokens at time s as:

$$\mathbf{Z}_s = \mathbf{G} + \mathbf{H}_s, \quad (1)$$

where \mathbf{G} denotes the reference latent representation and \mathbf{H}_s captures the residual changes at time s . For the reference frame, the residual is set to zero, i.e., $\mathbf{H}_1 = 0$. This formulation encourages persistent structure to be shared across time while allowing flexible temporal variation through residual offsets. The resulting latent representations $\{\mathbf{Z}_s\}$ compactly encode both persistent scene structure and time-varying geometry, and are used by the decoder for conditional 4D reconstruction.

Importantly, the proposed residual formulation introduces an explicit yet simple separation between time-invariant structure and time-varying components within a shared latent space. This separation is realized through the reference-residual parameterization, without introducing additional tokens or auxiliary objectives. The model is trained end-to-end using only supervision from the global point cloud at each time step, under which it automatically learns to allocate persistent geometry to the reference component and temporal variation to the residuals, without requiring motion labels.

3.3. Conditional Global Decoder

Given the temporally aggregated latent representation \mathbf{Z}_s at time step s , we introduce a global 3D decoder to reconstruct the complete scene geometry at time step s as a point cloud $\mathbf{Y}_s \in \mathbb{R}^{N_{\text{out}} \times 3}$. We formulate this global reconstruction as learning a conditional distribution over point sets $p(\mathbf{Y}_s | \mathbf{Z}_s)$, where the latent tokens \mathbf{Z}_s explicitly encode the canonical scene geometry via \mathbf{G} and the time-varying deformation via the residual motion tokens \mathbf{H}_s . This formulation supports globally consistent reconstruction over unordered point sets with variable cardinality, enabling the decoder to output point clouds at any density during inference.

To parameterize this conditional distribution, we adopt a conditional flow-matching decoder (Chang et al., 2024), which models point cloud generation as a continuous transformation from a simple prior to the target scene geometry, conditioned on latent tokens \mathbf{Z}_s . This design is well aligned with our setting: flow matching naturally supports variable point-cloud cardinalities and avoids explicit point correspondences, making it well suited to correspondence-free

reconstruction.

Conditional Flow Matching. We denote s as the discrete frame index corresponding to the input sequence, and $t \in [0, 1]$ as the continuous time variable used in the flow-matching process (Lipman et al., 2023). Given a target point cloud \mathbf{x}_1 and a noise sample ϵ , we define a linear interpolation path from noise to data as

$$\mathbf{x}_t = (1 - t)\epsilon + t\mathbf{x}_1. \quad (2)$$

Thus, $t = 0$ corresponds to the prior distribution and $t = 1$ corresponds to the target point-cloud distribution. The decoder learns a conditional velocity field $\Phi_{\text{dec}}(\mathbf{x}_t, t, \mathbf{Z}_s)$, which defines the ordinary differential equation (ODE)

$$\frac{d\mathbf{x}_t}{dt} = \Phi_{\text{dec}}(\mathbf{x}_t, t, \mathbf{Z}_s). \quad (3)$$

For this linear path, the target velocity is given by $\mathbf{v}_{\text{target}} = \mathbf{x}_1 - \epsilon$. The decoder is trained to predict this velocity field conditioned on the scene token \mathbf{Z}_s , thereby transporting samples from the prior distribution to the target point-cloud distribution.

Training Objective. We train our decoder using the flow-matching loss as the sole loss term, without requiring any additional supervision. For each time step s , we randomly sample a target point cloud $\mathbf{x}_1 \sim \mathbf{Y}_s$, a noise sample $\epsilon \sim \mathcal{U}([-1, 1]^3)$ from a uniform cube prior, and a flow time $t \sim \mathcal{U}(0, 1)$. Given the interpolation path in Equation (2), the training objective minimizes the distance between the predicted and target velocity fields:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t, \mathbf{x}_1, \epsilon, \mathbf{Z}_s} \left[\|\Phi_{\text{dec}}(\mathbf{x}_t, t, \mathbf{Z}_s) - (\mathbf{x}_1 - \epsilon)\|_2^2 \right]. \quad (4)$$

This objective encourages the decoder to reconstruct complete and temporally consistent global geometry conditioned on the aggregated scene representation, without additional supervision. During inference, we sample $\epsilon \sim \mathcal{U}([-1, 1]^3)$ and integrate the learned ODE forward from $t = 0$ to $t = 1$, starting from $\mathbf{x}_{t=0} = \epsilon$. This yields the reconstructed point cloud \mathbf{Y}_s .

Implementation Details. All models are implemented in PyTorch and trained on 8 NVIDIA H100 GPUs. We use the AdamW optimizer with a learning rate of 10^{-3} and a linear warm-up schedule. Models are trained for 250k iterations with a batch size of 8 sequences per GPU. Input and target point clouds are normalized using median-based scaling. During training, we randomly sample $S = 8$ frames per sequence, while evaluation is performed on sequences of 16 frames. Unless otherwise specified, the input partial point cloud contains 8,192 points per frame. We use $M = 512$ tokens per frame for latent representation. Additional implementation details are provided in the supplementary material. We plan to release the code upon acceptance.

4. Experiments

4.1. Experimental Setup

We evaluate DynaTok on both scene-level and object-level dynamic point cloud benchmarks, focusing on the setting of partial, correspondence-free point cloud inputs. All methods are evaluated on their ability to reconstruct temporally coherent 4D geometry from incomplete observations.

Datasets. For object-level reconstruction, we adopt **DeformingThings4D–Animals (DT4D-A)** (Li et al., 2021b), a standard benchmark used by prior 4D point-based methods (Cao et al., 2024). The dataset contains 1,972 sequences of deformable humanoids and animals exhibiting articulated and non-rigid motion. Each sequence consists of a single object without background, allowing us to isolate deformation modeling without scene-level clutter. We follow prior work and use point clouds back-projected from synthetic RGB-D scans as input.

For scene-level reconstruction, we use the **Kubric** dataset (Greff et al., 2022), which contains complex interactions of static and dynamic objects from Google Scanned Objects (Downs et al., 2022). It provides ground-truth complete point clouds for both static and moving components, enabling evaluation of 4D fusion and completion under partial observations. Compared to dynamic video datasets such as DAVIS (Perazzi et al., 2016), Kubric presents more challenging scenarios with fast motion, multiple independently moving objects, and camera motion. We follow the MOVi-F setup, using 10K training sequences and 200 test sequences, each with 30 frames at 10 FPS captured by a moving camera.

Baselines. To our knowledge, no prior method directly addresses 4D reconstruction from partial, correspondence-free point clouds at both object and scene levels as in our setting. We therefore compare DynaTok with the closest baseline, Motion2VecSets (Cao et al., 2024), which models temporal dynamics but is limited to single objects and fixed-size inputs (512 points per frame). Image-based 4D generative methods such as GVFD (Zhang et al., 2025a) require explicit correspondences and complete shapes and are thus inapplicable. We also include representative 3D point VAEs—3DShape2VecSet (Zhang et al., 2023), TripoSG (Li et al., 2025a), and TRELIS (Xiang et al., 2025b)—applied per frame as reference baselines to quantify the effect of missing temporal aggregation. We use pretrained models and recommended settings when available, as baseline methods typically require complete mesh supervision for training.

Evaluation Metrics. We evaluate reconstruction quality using standard geometric metrics, consistent with prior work

Table 1. **Object-Level Reconstruction Results on DT4D.** Our method improves over object-level 4D baselines.

| Method | Unseen Motion | | | Unseen Individual | | |
|-----------------------------------|---------------|--------------|--------------|-------------------|--------------|--------------|
| | Acc. ↓ | Comp. ↓ | NC ↑ | Acc. ↓ | Comp. ↓ | NC ↑ |
| <i>3D methods</i> | | | | | | |
| Shape2VecSet (Zhang et al., 2023) | 0.022 | 0.174 | 0.703 | 0.030 | 0.167 | 0.696 |
| TripoSG (Li et al., 2025a) | 0.051 | 0.165 | 0.740 | 0.048 | 0.143 | 0.732 |
| TRELIS (Xiang et al., 2025b) | 0.026 | 0.190 | 0.813 | 0.049 | 0.194 | 0.787 |
| <i>4D methods</i> | | | | | | |
| Motion2VecSets (Cao et al., 2024) | 0.055 | 0.060 | 0.856 | 0.061 | 0.065 | 0.824 |
| Ours | 0.023 | 0.021 | 0.914 | 0.027 | 0.026 | 0.877 |

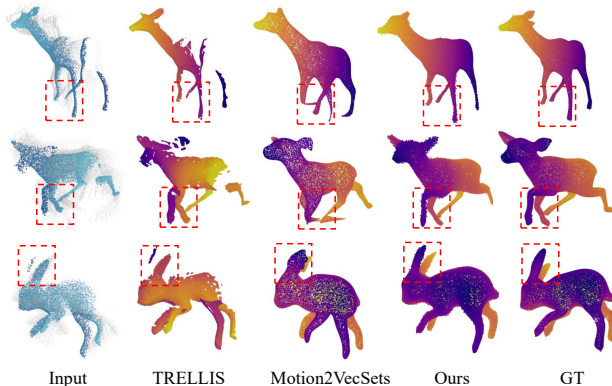


Figure 4. **Qualitative Reconstruction Results on the DT4D Dataset (Li et al., 2021b).** We compare the partial point cloud input setting from depth maps. Our method achieves clearer geometry that matches the ground truth point locations. We include the partial inputs from other time steps in white for reference.

such as CUT3R (Wang et al., 2025d). Specifically, we report Accuracy and Completeness, defined as the one-sided Chamfer Distance from prediction to ground truth and vice versa, respectively. We also report Normal Consistency, which measures the alignment of surface normals and reflects local geometric fidelity. For point cloud outputs, we estimate normals using Open3D (Zhou et al., 2018).

4.2. Object-level 4D Reconstruction

We evaluate object-level 4D reconstruction on DT4D-A under the standard Unseen Motion and Unseen Individual settings (Cao et al., 2024; Tang et al., 2021). All methods are assessed using point-based metrics, with mesh outputs uniformly downsampled to 50k points for consistency. As shown in Table 1, DynaTok consistently outperforms prior 4D baselines, achieving much lower completeness error and higher normal consistency, highlighting its ability to recover missing geometry via temporal aggregation.

Figure 4 further shows qualitative results under depth-based partial input. Motion2VecSets (Cao et al., 2024) captures coarse motion but exhibits missing regions and inaccurate articulation, while DynaTok produces cleaner geometry and more coherent motion, closely matching ground truth despite severe occlusions. These results show that latent-space

Table 2. Scene-Level Reconstruction on Kubric. Our method produces better reconstruction results than competing baselines. Motion2VecSets is restricted to 512 input points and is therefore excluded from FG + BG.

| Method | Foreground | | | Foreground + Background | | |
|-----------------------------------|--------------|--------------|--------------|-------------------------|--------------|--------------|
| | Acc. ↓ | Comp. ↓ | NC ↑ | Acc. ↓ | Comp. ↓ | NC ↑ |
| <i>3D methods</i> | | | | | | |
| Shape2VecSet (Zhang et al., 2023) | 0.008 | 0.027 | 0.680 | 0.009 | 0.026 | 0.724 |
| TripoSG (Li et al., 2025a) | 0.012 | 0.023 | 0.711 | 0.042 | 0.020 | 0.783 |
| TRELLIS (Xiang et al., 2025b) | 0.009 | 0.029 | 0.829 | 0.011 | 0.025 | 0.874 |
| <i>4D methods</i> | | | | | | |
| Motion2VecSets (Cao et al., 2024) | 0.061 | 0.063 | 0.545 | – | – | – |
| Ours | 0.008 | 0.010 | 0.835 | 0.010 | 0.014 | 0.888 |

temporal aggregation enables robust 4D reconstruction from correspondence-free partial point clouds.

4.3. Scene-level 4D Reconstruction

We evaluate scene-level 4D reconstruction on the Kubric dataset (Greff et al., 2022) under two settings: (i) multi-object foreground with sparse partial observations, and (ii) full scenes including static background geometry with denser inputs. Input point clouds are downsampled to $N_{in} = 8192$ to match baseline VAE training settings. As shown in Table 2, DynaTok achieves the best or tied-best accuracy and substantially improves completeness and normal consistency. Object-centric 4D methods such as Motion2VecSets (Cao et al., 2024) are not applicable to full scenes due to fixed input size and single-object assumptions. Compared to frame-wise 3D latent baselines, DynaTok achieves substantially better reconstruction quality, highlighting the importance of explicit temporal aggregation.

Qualitative results in Figure 5 further confirm these findings. Frame-wise 3D generative baselines (e.g., TripoSG (Li et al., 2025a), 3DShape2VecSet (Zhang et al., 2023)) produce fragmented and noisy geometry due to the lack of temporal integration, whereas DynaTok aggregates partial observations into a coherent canonical space, yielding dense and accurate reconstructions of both dynamic objects and background structures.

4.4. Ablation Study

We conduct comprehensive ablation studies on the DT4D-A dataset to critically assess the contribution of the architectural innovations within the DynaTok framework, focusing on decoupling effectiveness, tokenization efficiency, and robustness to partial input.

Per-frame (3D) vs Joint (4D) Encoder. The central finding confirms the importance of global attention. When the sequence is processed by the “Per-frame Only” baseline, performance severely degrades, demonstrating that simple token-wise extraction fails to establish the canonical

Table 3. Ablation Study of DynaTok Components on DT4D-A Dataset.

| Component | Configuration | Acc ↓ | Comp ↓ | NC ↑ |
|----------------------|-----------------|--------------|--------------|--------------|
| Encoder | Per-frame Only | 0.276 | 0.301 | 0.531 |
| | Geometry-Motion | 0.027 | 0.026 | 0.877 |
| # Tokens | $M = 128$ | 0.030 | 0.032 | 0.869 |
| | $M = 256$ | 0.027 | 0.026 | 0.877 |
| | $M = 512$ | 0.026 | 0.030 | 0.881 |
| # Train Views | 2 Views | 0.031 | 0.035 | 0.864 |
| | 4 Views | 0.028 | 0.032 | 0.872 |
| | 8 Views | 0.027 | 0.026 | 0.877 |

space and temporal consistency required for decoding. The “Per-frame Only” setting trains the model to perform single-frame completion. The substantial gain achieved by the full DynaTok model in Table 3 ($0.276 \rightarrow 0.027$ in Acc) verifies that joint spatiotemporal alignment in the token space is essential for stabilizing geometry and constructing a robust canonical representation for 4D reconstruction.

Number of Latent Tokens (M). This experiment investigates the critical trade-off between the token count per frame (M) and the resulting geometric fidelity and computational efficiency. We varied M across several settings, observing that scaling the token count from a coarse $M = 128$ yields substantial gains in precision and coverage.

Impact of Number of Training Views. We further test the generalization capacity of DynaTok by evaluating how input sparsity during training affects the quality of synthesized output. We vary the number of available training views among 2, 4, and 8 views, while testing against a consistent dense 16-view sequence. The quantitative results in Table 3 demonstrate a clear dependency: metrics show consistent improvement as the training view count increases. This confirms that exposing the network to richer input views is essential for distilling a highly accurate and robust global geometry token.

Canonical Space Evolution Across Time Steps. To analyze temporal aggregation, we study how the canonical representation (first frame) evolves as additional frames are incorporated ($S = 1, 4, 8, 12, 16$) in Figure 6. With only a single frame, the canonical space captures coarse geometry with limited coverage due to sparse observations. As more frames are added, the model progressively aggregates newly observed structures, refining both global shape and local details. The reconstruction becomes increasingly dense, smooth, and complete, indicating effective accumulation of information across time. This behavior confirms that our method fuses multi-frame observations into a unified canonical representation, rather than reconstructing each frame independently.

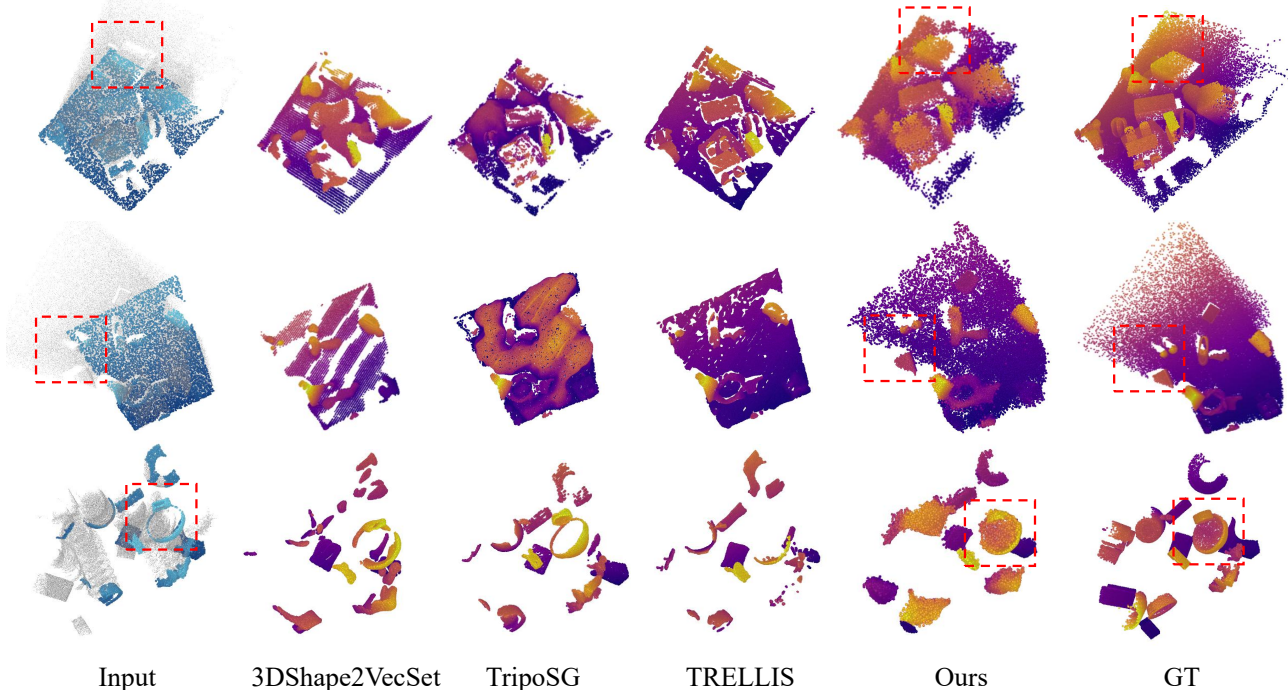


Figure 5. **Qualitative Reconstruction Results on Kubric** (Greff et al., 2022). The red boxes highlight regions where our method better preserves geometry than the baselines and effectively aggregates temporal information into a geometrically consistent canonical space. We include the partial inputs from other time steps in white for reference.

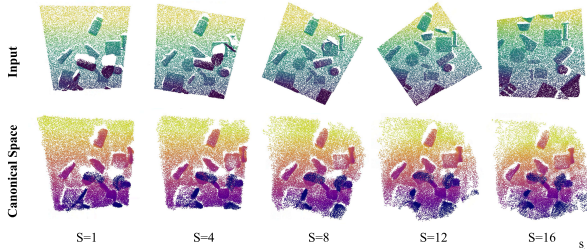


Figure 6. **Qualitative results for canonical space evolution across different time steps.** As additional frames are progressively observed, the canonical space anchored at the reference frame ($s = 1$) is expanded consistently to incorporate new geometric information.

Model Complexity. We further compare model complexity in terms of parameter count, FLOPs, and inference speed. As shown in Table 4, DynaTok uses only 19M parameters and requires approximately 6.6k GFLOPs per 16-frame scene. Inference speed is measured on an L40S GPU using 16-frame sequences. Compared with Motion2VecSets and TRELIS, DynaTok is substantially more compact and computationally efficient, while maintaining strong reconstruction quality.

Table 4. **Model complexity comparison.** FLOPs and inference time are measured for 16-frame scenes, with runtime evaluated on an L40S GPU. DynaTok is more compact and efficient than prior methods.

| Method | Params | GFLOPs | Time / scene |
|-----------------------------------|--------|--------|--------------|
| TRELIS (Xiang et al., 2025b) | 1.6B | ~3.7M | ~2 min |
| Motion2VecSets (Cao et al., 2024) | 423M | ~81.7k | ~45s |
| DynaTok (Ours) | 19M | ~6.6k | ~22s |

4.5. Zero-Shot In-the-Wild Results

To evaluate real-world generalization under more challenging input conditions, we test our model without fine-tuning on external datasets, including Bonn RGB-D (Palazzolo et al., 2019) and DAVIS (Perazzi et al., 2016). Instead of using depth sensor measurements, we directly use point maps predicted by the off-the-shelf CUT3R model (Wang et al., 2025d), which are noisy, incomplete, and often affected by scale ambiguity. Despite this substantial domain gap from the clean synthetic training data, our method preserves reasonable reconstruction quality and temporal coherence (Figure 7), demonstrating robustness to noisy and partial real-world observations. This suggests a downstream application of DynaTok: 4D fusion with off-the-shelf monocular reconstruction models (Zhang et al., 2026; Karhade et al., 2025).

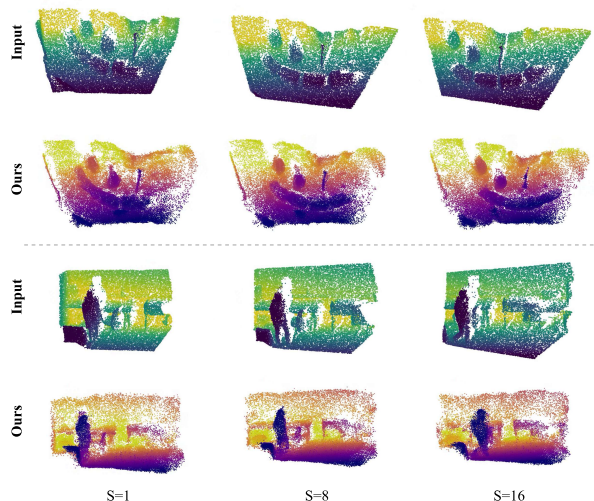


Figure 7. **Qualitative results on in-the-wild data from Bonn RGB-D (Palazzolo et al., 2019) and DAVIS (Perazzi et al., 2016).** We evaluate the generalization ability of our method on real-world scenes using CUT3R’s predictions as input.

5. Conclusion

We introduced DynaTok, a method for 4D reconstruction from partial, correspondence-free point cloud sequences. By emphasizing temporal aggregation in a latent representation, our approach integrates incomplete observations across time to recover coherent 4D geometry without relying on images or explicit tracking. Experiments on object- and scene-level benchmarks demonstrate improved reconstruction quality and temporal consistency under sparse inputs. We believe this work highlights the importance of structured temporal aggregation for point-based 4D reconstruction and provides a foundation for extending dynamic scene modeling to richer sensing modalities such as video and longer temporal horizons.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Alexiadis, D. S., Chatzitofis, A., Zioulis, N., Zoidi, O., Louizis, G., Zarpalas, D., and Daras, P. An integrated platform for live 3d human reconstruction and motion capturing. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(4):798–813, 2016.

Cao, W., Luo, C., Zhang, B., Nießner, M., and Tang, J. Motion2vecsets: 4d latent vector set diffusion for non-rigid shape reconstruction and tracking. In *CVPR*, pp.

20496–20506, 2024.

Chan, E. R., Lin, C. Z., Chan, M. A., Nagano, K., Pan, B., De Mello, S., Gallo, O., Guibas, L. J., Tremblay, J., Khamis, S., et al. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, pp. 16123–16133, 2022.

Chang, J.-H. R., Wang, Y., Martin, M. A. B., Gu, J., Zhao, X., Susskind, J., and Tuzel, O. 3d shape tokenization via latent flow matching. *arXiv preprint arXiv:2412.15618*, 2024.

Chen, W., Zhang, G., Wimbauer, F., Wang, R., Araslanov, N., Vedaldi, A., and Cremers, D. Back on track: Bundle adjustment for dynamic scene reconstruction. In *ICCV*, pp. 4951–4960, 2025.

Chen, W., Zheng, C., Zhang, G., Vedaldi, A., and Cremers, D. Nova3r: Non-pixel-aligned visual transformer for amodal 3d reconstruction. In *ICLR*, 2026.

Chen, Y., Zheng, C., Xu, H., Zhuang, B., Vedaldi, A., Cham, T.-J., and Cai, J. Mvsplat360: Feed-forward 360 scene synthesis from sparse views. *NeurIPS*, 37:107064–107086, 2024.

Dai, A., Nießner, M., Zollhöfer, M., Izadi, S., and Theobalt, C. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, 36(4):1, 2017.

Dong, Z., Song, J., Chen, X., Guo, C., and Hilliges, O. Shape-aware multi-person pose estimation from multi-view images. In *ICCV*, pp. 11158–11168, 2021.

Downs, L., Francis, A., Koenig, N., Kinman, B., Hickman, R., Reymann, K., McHugh, T. B., and Vanhoucke, V. Google scanned objects: A high-quality dataset of 3d scanned household items. In *ICRA*, pp. 2553–2560. IEEE, 2022.

Feng, H., Zhang, J., Wang, Q., Ye, Y., Yu, P., Black, M. J., Darrell, T., and Kanazawa, A. St4rtrack: Simultaneous 4d reconstruction and tracking in the world. In *ICCV*, pp. 8503–8513, 2025.

Gao, J., Shen, T., Wang, Z., Chen, W., Yin, K., Li, D., Litany, O., Gojcic, Z., and Fidler, S. Get3d: A generative model of high quality 3d textured shapes learned from images. *NeurIPS*, 35:31841–31854, 2022.

Gao, R., Holynski, A., Henzler, P., Brussee, A., Martin Brullalla, R., Srinivasan, P., Barron, J., and Poole, B. Cat3d: Create anything in 3d with multi-view diffusion models. *NeurIPS*, 37:75468–75494, 2024.

Greff, K., Belletti, F., Beyer, L., Doersch, C., Du, Y., Duckworth, D., Fleet, D. J., Gnanaprasam, D., Golemo, F.,

- Herrmann, C., Kipf, T., Kundu, A., Lagun, D., Laradji, I., Liu, H.-T. D., Meyer, H., Miao, Y., Nowrouzezahrai, D., Oztireli, C., Pot, E., Radwan, N., Rebain, D., Sabour, S., Sajjadi, M. S. M., Sela, M., Sitzmann, V., Stone, A., Sun, D., Vora, S., Wang, Z., Wu, T., Yi, K. M., Zhong, F., and Tagliasacchi, A. Kubric: a scalable dataset generator. 2022.
- Gupta, A., Xiong, W., Nie, Y., Jones, I., and Oğuz, B. 3dgen: Triplane latent diffusion for textured mesh generation. *arXiv preprint arXiv:2303.05371*, 2023.
- Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., and Tan, H. Lrm: Large reconstruction model for single image to 3d. In *ICLR*, 2024.
- Huang, J., Gojcic, Z., Atzmon, M., Litany, O., Fidler, S., and Williams, F. Neural kernel surface reconstruction. In *CVPR*, pp. 4369–4379, 2023a.
- Huang, S., Gojcic, Z., Wang, Z., Williams, F., Kasten, Y., Fidler, S., Schindler, K., and Litany, O. Neural lidar fields for novel view synthesis. In *ICCV*, pp. 18236–18246, 2023b.
- Huang, W., Chao, Y.-W., Mousavian, A., Liu, M.-Y., Fox, D., Mo, K., and Fei-Fei, L. Pointworld: Scaling 3d world models for in-the-wild robotic manipulation. *arXiv preprint arXiv:2601.03782*, 2026.
- Huang, X., Wang, J., Xia, Q., Chen, S., Yang, B., Li, X., Wang, C., and Wen, C. V2x-r: Cooperative lidar-4d radar fusion with denoising diffusion for 3d object detection. In *CVPR*, pp. 27390–27400, June 2025.
- Huang, Z., Yu, Y., Xu, J., Ni, F., and Le, X. Pf-net: Point fractal network for 3d point cloud completion. In *CVPR*, pp. 7662–7670, 2020.
- Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 559–568, 2011.
- Jiang, Y., Zhang, L., Gao, J., Hu, W., and Yao, Y. Consistent4d: Consistent 360° dynamic object generation from monocular video. In *ICLR*, 2024.
- Jiang, Z., Zheng, C., Laina, I., Larlus, D., and Vedaldi, A. Mesh4d: 4d mesh reconstruction and tracking from monocular video. *CVPR*, 2026.
- Jun, H. and Nichol, A. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023.
- Karhade, J., Keetha, N., Zhang, Y., Gupta, T., Sharma, A., Scherer, S., and Ramanan, D. Any4d: Unified feed-forward metric 4d reconstruction. *arXiv preprint arXiv:2512.10935*, 2025.
- Kopf, J., Rong, X., and Huang, J.-B. Robust consistent video depth estimation. In *CVPR*, pp. 1611–1621, June 2021.
- Kosiorrek, A. R., Strathmann, H., Zoran, D., Moreno, P., Schneider, R., Mokrá, S., and Rezende, D. J. Nerf-vae: A geometry aware 3d scene generative model. In *ICML*, pp. 5742–5752. PMLR, 2021.
- Lei, J. and Daniilidis, K. Cadex: Learning canonical deformation coordinate space for dynamic surface representation via neural homeomorphism. In *CVPR*, pp. 6624–6634, 2022.
- Li, R., Li, X., Hui, K.-H., and Fu, C.-W. Sp-gan: Sphere-guided 3d shape generation and manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–12, 2021a.
- Li, Y., Takehara, H., Taketomi, T., Zheng, B., and Nießner, M. 4dcomplete: Non-rigid motion estimation beyond the observable surface. In *ICCV*, pp. 12706–12716, 2021b.
- Li, Y., Zou, Z.-X., Liu, Z., Wang, D., Liang, Y., Yu, Z., Liu, X., Guo, Y.-C., Liang, D., Ouyang, W., et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025a.
- Li, Z., Tucker, R., Cole, F., Wang, Q., Jin, L., Ye, V., Kanazawa, A., Holynski, A., and Snavely, N. Megasam: Accurate, fast and robust structure and motion from causal dynamic videos. In *CVPR*, pp. 10486–10496, 2025b.
- Liang, H., Yin, Y., Xu, D., Wang, Z., Plataniotis, K. N., Zhao, Y., Wei, Y., et al. Diffusion4d: Fast spatial-temporal consistent 4d generation via video diffusion models. *NeurIPS*, 37:110854–110875, 2024.
- Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *ICLR*, 2023.
- Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., and Vondrick, C. Zero-1-to-3: Zero-shot one image to 3d object. In *ICCV*, pp. 9298–9309, 2023.
- Matsuki, H., Bae, G., and Davison, A. J. 4dtam: Non-rigid tracking and mapping via dynamic surface gaussians. In *CVPR*, pp. 26921–26932, 2025.
- Niemeyer, M., Mescheder, L., Oechsle, M., and Geiger, A. Occupancy flow: 4d reconstruction by learning particle dynamics. In *ICCV*, 2019.

- Palazzolo, E., Behley, J., Lottes, P., Giguere, P., and Stachniss, C. Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. In *IROS*, pp. 7855–7862. IEEE, 2019.
- Pan, Z., Yang, Z., Zhu, X., and Zhang, L. Efficient4d: Fast dynamic 3d object generation from a single-view video. *arXiv preprint arXiv:2401.08742*, 2024.
- Pavlo, D., Spinks, G., Hofmann, T., Moens, M.-F., and Lucchi, A. Convolutional generation of textured 3d meshes. *NeurIPS*, 33:870–882, 2020.
- Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., and Sorkine-Hornung, A. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, pp. 724–732, 2016.
- Qian, S., Zhang, G., Wu, S., and Cremers, D. Flow4r: Unifying 4d reconstruction and tracking with scene flow. *arXiv preprint arXiv:2602.14021*, 2026.
- Rajič, F., Xu, H., Mihajlovic, M., Li, S., Demir, I., Gündoğdu, E., Ke, L., Prokudin, S., Pollefeys, M., and Tang, S. Multi-view 3d point tracking. In *ICCV*, pp. 59–68, 2025.
- Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., and Koltun, V. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1623–1637, 2020.
- Ren, J., Pan, L., Tang, J., Zhang, C., Cao, A., Zeng, G., and Liu, Z. Dreamgaussian4d: Generative 4d gaussian splatting. *arXiv preprint arXiv:2312.17142*, 2023.
- Ren, J., Xie, C., Mirzaei, A., Kreis, K., Liu, Z., Torralba, A., Fidler, S., Kim, S. W., Ling, H., et al. L4gm: Large 4d gaussian reconstruction model. *NeurIPS*, 37:56828–56858, 2024a.
- Ren, X., Huang, J., Zeng, X., Museth, K., Fidler, S., and Williams, F. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *CVPR*, pp. 4209–4219, 2024b.
- Sang, L., Canfes, Z., Cao, D., Marin, R., Bernard, F., and Cremers, D. Twosquared: 4d generation from 2d image pairs. *3DV*, 2025.
- Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- Sucar, E., Lai, Z., Insafutdinov, E., and Vedaldi, A. Dynamic point maps: A versatile representation for dynamic 3d reconstruction. In *ICCV*, pp. 7295–7305, 2025.
- Szymanowicz, S., Zhang, J. Y., Srinivasan, P., Gao, R., Brussee, A., Holynski, A., Martin-Brualla, R., Barron, J. T., and Henzler, P. Bolt3d: Generating 3d scenes in seconds. In *ICCV*, pp. 24846–24857, 2025.
- Tang, J., Xu, D., Jia, K., and Zhang, L. Learning parallel dense correspondence from spatio-temporal descriptors for efficient and robust 4d reconstruction. In *CVPR*, pp. 6022–6031, 2021.
- Tang, J., Chen, Z., Chen, X., Wang, T., Zeng, G., and Liu, Z. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *ECCV*, pp. 1–18. Springer, 2024.
- Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., Kreis, K., et al. Lion: Latent point diffusion models for 3d shape generation. *NeurIPS*, 35:10021–10039, 2022.
- Voleti, V., Yao, C.-H., Boss, M., Letts, A., Pankratz, D., Tochilkin, D., Laforte, C., Rombach, R., and Jampani, V. Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. In *ECCV*, pp. 439–457. Springer, 2024.
- Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., and Novotny, D. Vggt: Visual geometry grounded transformer. In *CVPR*, pp. 5294–5306, 2025a.
- Wang, L., Zheng, W., Ren, Y., Jiang, H., Cui, Z., Yu, H., and Lu, J. Occsora: 4d occupancy generation models as world simulators for autonomous driving. *arXiv preprint arXiv:2405.20337*, 2024.
- Wang, M., Li, C., and Vaxman, A. Canfields: Consolidating diffeomorphic flows for non-rigid 4d interpolation from arbitrary-length sequences. In *ICCV*, pp. 28587–28598, 2025b.
- Wang, Q., Ye, V., Gao, H., Zeng, W., Austin, J., Li, Z., and Kanazawa, A. Shape of motion: 4d reconstruction from a single video. In *ICCV*, pp. 9660–9672, 2025c.
- Wang, Q., Zhang, Y., Holynski, A., Efros, A. A., and Kanazawa, A. Continuous 3d perception model with persistent state. In *CVPR*, pp. 10510–10522, 2025d.
- Wang, S., Jiang, Z., Yang, X., and Wang, X. C4d: 4d made from 3d through dual correspondences. In *ICCV*, pp. 7570–7580, 2025e.
- Wimbauer, F., Chen, W., Muhle, D., Rupprecht, C., and Cremers, D. Anycam: Learning to recover camera poses and intrinsics from casual videos. In *CVPR*, pp. 16717–16727, 2025.
- Wu, G., Yi, T., Fang, J., Xie, L., Zhang, X., Wei, W., Liu, W., Tian, Q., and Wang, X. 4d gaussian splatting for real-time

- dynamic scene rendering. In *CVPR*, pp. 20310–20320, 2024a.
- Wu, J., Zhang, C., Xue, T., Freeman, B., and Tenenbaum, J. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *NeurIPS*, 29, 2016.
- Wu, R., Gao, R., Poole, B., Trevithick, A., Zheng, C., Barron, J. T., and Holynski, A. Cat4d: Create anything in 4d with multi-view video diffusion models. In *CVPR*, pp. 26057–26068, 2025.
- Wu, S., Lin, Y., Zhang, F., Zeng, Y., Xu, J., Torr, P., Cao, X., and Yao, Y. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. *NeurIPS*, 37:121859–121881, 2024b.
- Xiang, J., Lv, Z., Xu, S., Deng, Y., Wang, R., Zhang, B., Chen, D., Tong, X., and Yang, J. Structured 3d latents for scalable and versatile 3d generation. In *CVPR*, pp. 21469–21480, 2025a.
- Xiang, J., Lv, Z., Xu, S., Deng, Y., Wang, R., Zhang, B., Chen, D., Tong, X., and Yang, J. Structured 3d latents for scalable and versatile 3d generation. In *CVPR*, 2025b.
- Xie, Y., Yao, C.-H., Voleti, V., Jiang, H., and Jampani, V. Sv4d: Dynamic 3d content generation with multi-frame and multi-view consistency. In *ICLR*, 2025.
- Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., and Zhao, H. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, pp. 10371–10381, 2024a.
- Yang, Z., Yang, H., Pan, Z., and Zhang, L. Real-time photo-realistic dynamic scene representation and rendering with 4d gaussian splatting. In *ICLR*, 2024b.
- Yin, W., Zhang, C., Chen, H., Cai, Z., Yu, G., Wang, K., Chen, X., and Shen, C. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *ICCV*, pp. 9043–9053, 2023.
- Yu, X., Rao, Y., Wang, Z., Liu, Z., Lu, J., and Zhou, J. Pointr: Diverse point cloud completion with geometry-aware transformers. In *ICCV*, pp. 12498–12507, 2021.
- Zhang, B., Tang, J., Niessner, M., and Wonka, P. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics (TOG)*, 42(4):1–16, 2023.
- Zhang, B., Xu, S., Wang, C., Yang, J., Zhao, F., Chen, D., and Guo, B. Gaussian variation field diffusion for high-fidelity video-to-4d synthesis. In *ICCV*, pp. 12502–12513, 2025a.
- Zhang, C., Moing, G. L., Koppula, S., Rocco, I., Momeni, L., Xie, J., Sun, S., Sukthankar, R., Barral, J. K., Hadsell, R., et al. Efficiently reconstructing dynamic scenes one d4rt at a time. *CVPR*, 2026.
- Zhang, D., Li, G., Li, J., Bressieux, M., Hilliges, O., Pollefeys, M., Van Gool, L., and Wang, X. Egogaussian: Dynamic scene understanding from egocentric video with 3d gaussian splatting. In *3DV*, pp. 1091–1102. IEEE, 2025b.
- Zhang, H., Chen, X., Wang, Y., Liu, X., Wang, Y., and Qiao, Y. 4diffusion: Multi-view video diffusion model for 4d generation. *NeurIPS*, 37:15272–15295, 2024a.
- Zhang, J., Herrmann, C., Hur, J., Jampani, V., Darrell, T., Cole, F., Sun, D., and Yang, M.-H. Monst3r: A simple approach for estimating geometry in the presence of motion. *ICLR*, 2025c.
- Zhang, L., Wang, Z., Zhang, Q., Qiu, Q., Pang, A., Jiang, H., Yang, W., Xu, L., and Yu, J. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)*, 43(4):1–20, 2024b.
- Zhang, Z., Cole, F., Li, Z., Rubinstein, M., Snavely, N., and Freeman, W. T. Structure and motion from casual videos. In *ECCV*, pp. 20–37. Springer, 2022.
- Zheng, L., Li, S., Tan, B., Yang, L., Chen, S., Huang, L., Bai, J., Zhu, X., and Ma, Z. Rcfusion: Fusing 4-d radar and camera with bird’s-eye view features for 3-d object detection. *IEEE Transactions on Instrumentation and Measurement*, 72:1–14, 2023a.
- Zheng, X.-Y., Pan, H., Wang, P.-S., Tong, X., Liu, Y., and Shum, H.-Y. Locally attentional sdf diffusion for controllable 3d shape generation. *ACM Transactions on Graphics (ToG)*, 42(4):1–13, 2023b.
- Zhong, X., Pan, Y., Stachniss, C., and Behley, J. 3d lidar mapping in dynamic environments using a 4d implicit neural representation. In *CVPR*, pp. 15417–15427, 2024.
- Zhou, Q.-Y., Park, J., and Koltun, V. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018.
- Zhou, X., Lin, Z., Shan, X., Wang, Y., Sun, D., and Yang, M.-H. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *CVPR*, pp. 21634–21643, 2024.
- Zhu, J.-Y., Zhang, Z., Zhang, C., Wu, J., Torralba, A., Tenenbaum, J., and Freeman, B. Visual object networks: Image generation with disentangled 3d representations. *NeurIPS*, 31, 2018.

A. Appendix

A.1. Implementation Details

Point Token Extractor. We adopt the same encoder backbone as the TripoSG VAE encoder (Li et al., 2025a) to extract latent point tokens. The encoder takes as input a point cloud $\mathbf{X}_s \in \mathbb{R}^{N \times 3}$ and outputs 512 latent tokens of dimension 64. We use one cross-attention layer to fuse spatial features, followed by six self-attention layers to enhance token-wise interactions. During evaluation, we use 8,192 input points per frame, subsampled from valid depth pixels at 224×224 resolution.

Alignment Module. To aggregate information across frames, we use an alignment module composed of eight alternating frame-level and global-level attention blocks. Frame-level attention updates tokens using intra-frame relationships, while global-level attention enables cross-frame aggregation to progressively form geometry and motion tokens. The module outputs 512 geometry tokens and $512 \times S$ motion tokens, each with dimension 64.

Flow-Matching Decoder. The decoder is a lightweight 3-layer transformer with self- and cross-attention, similar to (Chang et al., 2024), and is conditioned on the latent tokens \mathbf{Z}_s . It takes an interpolated point cloud \mathbf{x}_t and the flow time t as input, and predicts the conditional velocity field. The decoder uses a hidden dimension of 64. During inference, the final reconstructed 3D point positions are obtained by integrating the learned ODE.

Training and Inference. During training, we use 8-frame sequences with 30,000 input points per frame, a batch size of 4 per GPU, a learning rate of 10^{-3} , and train for 250k iterations. For flow matching, we sample noise from a uniform cube distribution and uniformly sample flow times from $[0, 1]$. Following the main text, we define a linear interpolation path from noise to data and train the model to predict the corresponding target velocity field. The interpolated points are Fourier-embedded and decoded by the transformer conditioned on the latent tokens \mathbf{Z}_s .

During inference, we solve the learned ODE using 25 Euler steps to generate 50,000 output points. For Kubric evaluation, we use 16-frame sequences and compare the reconstructed point clouds against 100,000 ground-truth target points.

A.2. Additional Analysis on RGB Input with DINOv2 Tokens

Our work focuses on partial point cloud input, a practical setting motivated by depth cameras, LiDAR, and 3D scanners. Nevertheless, our framework is modular and can naturally incorporate RGB cues. To analyze this extension, we use a frozen DINOv2 encoder to extract image tokens, project them to the point-token feature dimension, and concatenate them with point-cloud tokens before the alignment module. The remaining architecture is unchanged. For efficiency, we train this variant for 10 epochs on Kubric Scene while freezing the point cloud encoder, DINOv2 encoder, and decoder, and fine-tuning only the projection layer and alignment module.

Table S1 summarizes the results. RGB-only conditioning with frozen DINOv2 features already produces reasonable 4D reconstructions, although it underperforms partial point-cloud input, as DINOv2 features are semantic and not optimized for 3D geometry. When combined with point clouds, DINOv2 tokens are most useful in the sparse-input regime: with 1024 input points, they improve the accuracy from 0.014 to 0.013, Completeness from 0.020 to 0.017, and Normal consistency from 0.855 to 0.866. With semi-dense input, i.e., 8192 points, the gains are limited and Normal consistency slightly decreases. These results suggest that RGB cues provide complementary information mainly when geometric observations are sparse, and further demonstrate that our framework can be extended to multi-modal inputs in a shared latent token space with minimal architectural changes.

A.3. Further analysis of decoupling and temporal consistency

We provide two additional quantitative analyses to directly validate (i) geometry–motion decomposition and (ii) temporal consistency.

Geometry–motion decomposition. We directly validate geometry–motion decoupling by comparing decoding with full tokens, i.e., geometry and motion tokens, against decoding with geometry-only tokens, where frame-0 tokens are replicated across time. As shown in Table S2, the full model remains stable over time, while geometry-only decoding degrades monotonically. Moreover, the motion contribution, measured as the Chamfer distance between geometry-only and full reconstructions, increases steadily over time. The small nonzero gap at $t = 0$ is due to the stochasticity of the flow-matching

Table S1. Additional analysis on RGB input with DINOv2 tokens. We evaluate both RGB-only conditioning and RGB-augmented point-cloud input. Frozen DINOv2 features can support reasonable 4D reconstruction from RGB alone and provide complementary cues under sparse point-cloud input.

| Setting | # Points | Method | Accuracy ↓ | Completeness ↓ | Normal ↑ |
|------------------|----------|-----------------------------|------------|----------------|----------|
| RGB-only | – | DINOv2 tokens | 0.024 | 0.021 | 0.800 |
| Geometry-only | 8192 | Partial Point Cloud | 0.010 | 0.014 | 0.880 |
| Sparse input | 1024 | Point-based | 0.014 | 0.020 | 0.855 |
| Sparse input | 1024 | Point-based + DINOv2 tokens | 0.013 | 0.017 | 0.866 |
| Semi-dense input | 8192 | Point-based | 0.010 | 0.014 | 0.888 |
| Semi-dense input | 8192 | Point-based + DINOv2 tokens | 0.010 | 0.013 | 0.881 |

Table S2. Geometry–motion decomposition analysis. We compare full-token decoding with geometry-only decoding, where frame-0 tokens are replicated across time. Lower accuracy error is better, while a higher motion contribution indicates a larger deformation component captured by motion tokens.

| Frame | Geo-Only Acc. ↓ | Motion Contribution ↑ | Full Model Acc. ↓ |
|-------|-----------------|-----------------------|-------------------|
| 0 | 0.137 | 0.021 | 0.139 |
| 4 | 0.151 | 0.047 | 0.138 |
| 8 | 0.162 | 0.071 | 0.137 |
| 12 | 0.168 | 0.081 | 0.140 |

decoder. Overall, these results support that geometry tokens capture the canonical structure, while residual tokens encode the time-varying deformation required to maintain reconstruction quality.

Temporal consistency. We measure frame-to-frame consistency using $CD(\text{pred}(t), \text{pred}(t + 1))$. As shown in Table S3, compared to TRELLIS (Xiang et al., 2025b), DynaTok achieves a $2\times$ lower mean Chamfer distance and a $2.5\times$ lower variance, indicating substantially more stable temporal reconstruction.

A.4. Robustness to Sparse Input Points

To evaluate the robustness of our method under varying input sparsity levels, we conduct experiments using different numbers of input points ($N = 512, 2048, 8192, 32768$). As shown in Figure 8, even when provided with extremely sparse input (e.g., $N = 512$), our model is still able to recover the overall scene structure and generate a coherent and geometrically plausible reconstruction in the canonical space. As the number of input points increases, the reconstruction gradually becomes denser and more detailed, while maintaining consistent geometry across all settings. This demonstrates that our method does not rely on dense or high-fidelity point inputs and can effectively infer missing structures, highlighting its strong robustness and generalization capability under sparse observations.

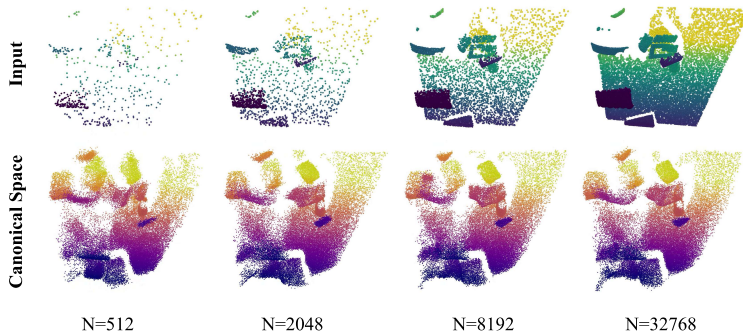


Figure 8. **Qualitative comparison on different numbers of input points.** Our method shows strong robustness towards sparse input (e.g., $N=512$).

Table S3. Temporal consistency analysis. We report frame-to-frame Chamfer distance between consecutive predictions. Lower values indicate more stable temporal reconstruction.

| Metric | DynaTok | TRELLIS (Xiang et al., 2025b) |
|-----------|---------|-------------------------------|
| Mean CD ↓ | 0.0044 | 0.0089 |
| Std. ↓ | 0.0007 | 0.0018 |
| Max CD ↓ | 0.0056 | 0.0124 |

A.5. Comparison with Image-based 4D Methods

Our work focuses on geometry-only input, i.e., partial point clouds, motivated by depth sensors, LiDAR, and 3D scanners. We therefore follow prior point-based 4D reconstruction settings such as Motion2VecSets (Cao et al., 2024). Since datasets such as DT4D do not provide RGB observations, direct comparison with image-based methods such as Diffusion4D (Liang et al., 2024) and Consistent4D (Jiang et al., 2024) is infeasible.

As an additional reference, we evaluate the publicly available pretrained Mesh4D (Jiang et al., 2026) model on Kubric, which provides both RGB and geometry, and reports results after ICP alignment. This comparison is not fully matched: Mesh4D uses RGB input, and its pretrained weights are not specifically trained for our scene-level reconstruction setting. Therefore, these results should be interpreted as a cross-model reference rather than a direct method ranking.

Table S4. Exploratory comparison with pretrained Mesh4D (Jiang et al., 2026) on Kubric (Greff et al., 2022) after ICP alignment. Since Mesh4D uses RGB input and is not specifically trained for our scene-level setting, this serves as a cross-modal reference rather than a fully matched comparison.

| Setting | Method | Comp. ↓ | Acc. ↓ | Norm. ↑ |
|-----------------|---------|---------|--------|---------|
| Foreground-Obj. | DynaTok | 0.0109 | 0.0088 | 0.8296 |
| Foreground-Obj. | Mesh4D | 0.0854 | 0.0780 | 0.5068 |
| Full | DynaTok | 0.0140 | 0.0112 | 0.8740 |
| Full | Mesh4D | 0.0576 | 0.0842 | 0.5359 |

A.6. Discussion of Failure Cases

Our model is generally robust to short-term occlusions and moderate motion, but it can degrade under prolonged occlusion, very fast motion, heavy object overlap, or motion patterns not represented in the training distribution. The scene model is also less reliable for highly deformable or strongly non-rigid objects, since our Kubric-based training data mainly covers rigid or near-rigid dynamics. Expanding training to larger and more diverse dynamic datasets is an important next step.