Marrying Optimal Transport and ODEs for Unified Continuous-Time 4D Reconstruction and Tracking

Liying Yang[1], **Hao Mo**[1], **Jialun Liu**[2*†], **Chen Liu**[2], **Xinxing Yu**[1], **Chenhao Guan**[1], **Hui Ma**[3,4], **Xiao Cao**[5], **Ajian Liu**[6], **Yanyan Liang**[1†]
School of Computer Science and Engineering, Macau University of Science and Technology[1]
The University of Queensland[2] Great Bay University[3] Tsinghua University[4]
National University of Singapore[5] Institute of Automation, Chinese Academy of Sciences[6]
Project Leader[*] Corresponding Authors[†]

Abstract

Existing unified 4D reconstruction and point tracking approaches typically rely on heuristic interpolations or just predict at integer timestamps, lacking kinematic coherence and failing to model dynamics at any arbitrary timestamp. In this paper, we propose Uni4R, a framework that unifies these tasks by learning continuous velocity fields through the synergy of Optimal Transport (OT) and Ordinary Differential Equation (ODE). Importantly, this continuous velocity field acts as a kinematic prior that mutually benefits both 4D reconstruction and point tracking. Specifically, we propose the Flow Matching Guided Decoder (FMGD). A global velocity branch first extracts anchor features that capture the global dynamic state of the sequence. Then, FMGD leverages Flow Matching (FM) theory to formulate a probability path defined by OT on the anchor feature manifold, instantiating it as FM-guided velocity features for velocity prediction. This establishes a robust kinematic inductive bias. Meanwhile, a point reconstruction branch provides geometric features. The local velocity prediction module then joint above features and time embeddings, to decode velocities at arbitrary timestamps. To overcome the absence of high-quality ground-truth velocities in fractional frames, we propose an integral-consistency training strategy. This strategy uses an ODE solver to integrate velocities to recover target pointmaps, enabling the model to be supervised end-to-end directly from integer timestamps. Experimental results demonstrate that Uni4R achieves SOTA performance in both 4D reconstruction and point tracking, and achieves SOTA in our new kinematics-aware benchmark at continuous time. Project page: `https://LiyingCV.github.io/Uni4R/`

1 Introduction

Existing unified 4D reconstruction and point tracking methods [7, 39] mainly focus on integer time steps, which restricts their temporal resolution. To establish point correspondences at arbitrary timestamps, some recent methods [24] resort to heuristic interpolations (e.g., B-spline). Unfortunately, these approximations lack underlying kinematic coherence, which requires points to propagate along continuous and energy-minimizing trajectories. By treating motion modeling as mere curve fitting rather than a continuous geometric flow, they frequently fail to preserve local geometric structures, often leading to distortions. More importantly, without a continuous kinematic prior, these methods fail to fully exploit the inherent mutual benefits between reconstruction and tracking.

Formulating 4D dynamics as an Ordinary Differential Equation (ODE) [3, 5, 29] local initial value problem offers a principled paradigm to achieve true spatiotemporal continuity. Given that ODE naturally yield smooth trajectories, we leverage ODE to learn the continuous velocity field. Furthermore,

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

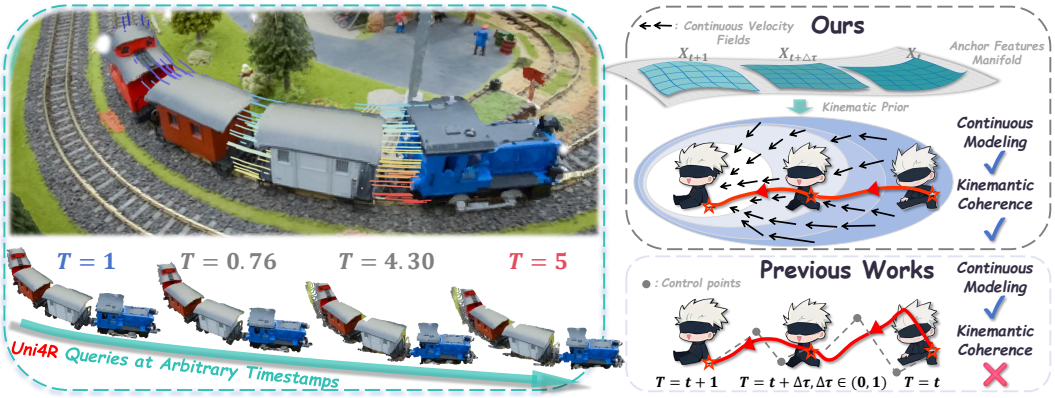

Figure 1: **Left:** Uni4R is a feed-forward transformer that queries point maps at arbitrary timestamp. **Right:** While existing methods like TraceAnything [24] rely on B-spline fitting and inherently lack kinematic coherence, Uni4R learns continuous velocity fields via the synergy of OT and ODE. Specifically, we formulate a probability path defined by OT on the anchor feature manifold to provide a kinematic inductive bias. Then, an ODE solver integrates velocities queried from the continuous velocity field to achieve continuous modeling with kinematic coherence.

this continuous velocity field acts as a kinematic priors that seamlessly bridges spatial topology and temporal dynamics. *Under this prior constraint, accurate point tracking provides kinematic guidance for geometric deformation, while high-quality 4D reconstruction regularizes the trajectory space, allowing the two tasks to mutually enhance each other's quality.* However, realizing this continuous formulation poses an optimization dilemma that when supervising a continuous velocity field using only point maps at integer timestamps, the inverse problem becomes highly ill-posed. Infinitely many arbitrary, curved velocity fields transport a point from state $\mathbf{P}_t$ to $\mathbf{P}_{t+1}$, rendering the model prone to overfitting and resulting in unstable trajectories.

To overcome these limitations, we propose Uni4R, a unified framework establishing a deep synergy between Optimal Transport (OT) and Ordinary Differential Equations (ODEs). Within this synergy, ODEs provide the mathematical foundation for continuous motion prediction at any arbitrary time, while OT provides a kinematic inductive bias with a global velocity direction prior to constrain the ill-posed problem. At its core, our proposed Flow Matching Guided Decoder (FMGD) leverages Flow Matching (FM) theory [21] to formulate the probability path defined by OT on anchor feature manifold. Specifically, a global velocity branch first extracts anchor features that capture the global dynamic state of the input sequence, implicitly embedding motion information for any non-integer moment. By formalizing this OT-defined probability path within the feature manifold, we obtain FM-guided velocity features that encode a global velocity direction prior as a kinematic inductive bias. Furthermore, a point reconstruction branch extracts geometric features. These geometric features and the FM-guided velocity features, conditioned on continuous time embeddings, are then sent to our local velocity prediction module to decode velocities at arbitrary timestamps. Through this unified framework, FMGD effectively constrains the ill-posed continuous inverse problem, enabling smooth and kinematically consistent 4D trajectories.

While FMGD can decode velocities, training this velocity field is hindered by the inherent absence of high-quality ground-truth velocities in fractional frames. To bridge the gap between predicted dynamics and integer-time ground-truth, we propose an integral-consistency training strategy. By employing an ODE solver to integrate velocities to recover target pointmaps, we enable robust end-to-end supervision from integer pointmaps. Ultimately, within this unified framework, the continuous velocity field acts as a kinematic prior that mutually benefits the accuracy of two tasks by implicitly modeling continuous motion. Remarkably, despite employing ODE, given 80-frame inputs, Uni4R requires only 1.57 seconds to infer results (66.52% faster than Trace Anything [24]).

Current evaluation protocols for point tracking mainly focus on integer timestamp, lacking standard evaluation in continuous time. To bridge this gap, we introduce a kinematics-aware benchmark designed to systematically evaluate kinematic coherence at continuous time.

In summary, the main contributions of our work are as follows:

1. We propose Uni4R, a unified framework that shifts 4D reconstruction and point tracking from heuristic interpolations to continuous kinematic modeling, unlocking the capability to query dynamics at any arbitrary timestamp.
2. We propose Flow Matching Guided Decoder (FMGD), which leverages OT to provide a global velocity direction prior as a kinematic inductive bias for constraining the ill-posed inverse problem. Moreover, we design an integral-consistency training strategy, enabling end-to-end supervision of continuous velocity field from integer time steps through ODE.
3. We present a kinematics-aware benchmark that establishes a unified evaluation standard for kinematic coherence in continuous time.
4. Experimental results demonstrate that Uni4R achieves SOTA performance in both 4D reconstruction and point tracking, and achieves SOTA in kinematics-aware benchmark.

2 Related Works

Reconstruction and Generation. Creating high-quality 3D and 4D contents, such as voxels [51, 73, 53, 72],meshes [25, 69, 12], point clouds [28, 59, 57, 58, 60], and 3D Gaussian splats [18, 66, 40] has become a rapidly evolving research area. Recent 4D generation methods [71, 68, 63, 48, 54] employ dynamic or multi-view prior from video diffusion [36, 43] or multi-view diffusion [22, 35] to optimize dynamic 4D representations via Score Distillation Sampling (SDS) [32, 27]. Recent methods [34, 62, 56, 55] generate 4D gaussian splatting in feed-forward manner based on diffusion or autoregressive model. Different from them, recent 3D and 4D reconstruction [45, 52, 44, 46, 38, 24, 39] output disjoint per-frame pointmaps instead of gaussian splatting [18, 47]. However, most of them focus on integer time steps. In contrast, Uni4R commits to modeling dynamics both at integer and fractional time steps.

Point Tracking. The paradigm of motion estimation has rapidly evolved from 2D optical flow to 3D point tracking. To enable tracking through occlusions, CoTracker [16] employs a transformer-based architecture, while CoTracker3 [17] leveraged unlabeled data to boost performance. SpatialTracker [49] proposes the first feed-forward 3D tracker though combining 2D tracking with monocular depth priors. SpatialTrackerV2 [50] scaled training across real and synthetic data. Recently, 3D tracking-based methods [7, 67, 24, 39, 33, 64] has shown impressive results. St4RTrack [7] and POMATO [67] demonstrate the benefits of jointly predict 4D reconstruction and tracking. Track4World [26] applies a 3D correlation scheme to estimate dense 2D and 3D tracking. TraceAnything [24] predicts a set of control points that parameterizes a trajectory via B-spline. However, they frequently fail to preserve local geometric structures due to lack underlying kinematic coherence. In contrast, Uni4R models dynamics via kinematic inductive both at integer and fractional time steps.

Flow Matching. Continuous Normalizing Flows (CNFs) [2, 10] have emerged as powerful tools for modeling distributions via Ordinary Differential Equations (ODEs). The pioneering work [21] established a simulation-free training paradigm for continuous vector fields and was improved by [41]. [41] integrates Optimal Transport (OT) to constrain ODE trajectories into deterministic, minimum-energy linear paths. However, its potential as a kinematic inductive bias for solving point tracking and 4D reconstruction remains unexplored. Our Uni4R formulates probability path defined by OT not for generation, but to query continuous and kinematics-aware velocities across feature manifold.

3 Methodology

Given a monocular frame sequence of a dynamic scene, Uni4R first extracts visual features using a shared ViT encoder. These features are then processed by FMGD for unified 4D reconstruction and point tracking. Fig. 2 shows the overview of Uni4R.

3.1 Continuous 4D Dynamics Modeling via Ordinary Differential Equation

Formally, to achieve continuous dynamic modeling, we model the 4D dynamics as a sequence of local initial value problems. Instead of integrating from a single global origin (i.e., global-based ODE), we define $P(k)$ as the anchor position at integer timestamp k. The position of a point at any arbitrary timestamp $k + \tau$ (where $\tau \in [0, 1)$) is then determined by integrating the velocities queried

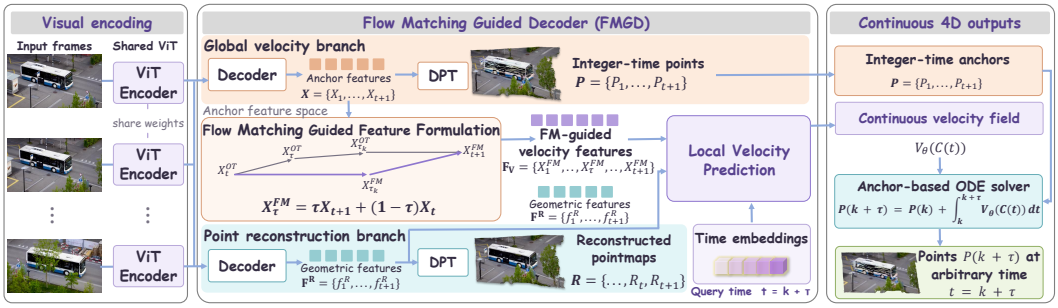

Figure 2: **Pipeline of Uni4R.** A ViT encoder extracts visual features from input frames, which are processed into geometric features F^R and anchor features X. Then, we formulate FM-guided velocity features F_V via an OT probability path on the anchor feature manifold. This provides a kinematic inductive bias characterizing feature-space transport. Finally, a Local Velocity Prediction (LVP) module fuses X, F^R, F_V, and time embeddings to predict a continuous velocity field, which an ODE solver integrates to produce points at any arbitrary time t.

from continuous velocity field $\mathbf{V}_\theta(C(t))$ starting from its nearest anchor $P(k)$, named anchor-based ODE, where $C(t)$ denotes the time-dependent conditioning information of the velocity predictor:

$$P(k+\tau) = P(k) + \int_k^{k+\tau} \mathbf{V}_\theta(C(t))dt \tag{1}$$

By grounding the continuous velocity field on these periodic anchors, our framework effectively mitigates the accumulation of integration errors. However, recovering the continuous velocity field solely from observations at discrete integer timestamps remains a highly ill-posed inverse problem. While the ODE formulation ensures that a given velocity field yields a deterministic trajectory, there exists an infinite number of continuous paths that can align with the ground truth at integer timestamps. Therefore, our framework requires a robust kinematic inductive bias to constrain the solution space and predict the most kinematically consistent trajectory among all possible continuous paths.

3.2 Flow Matching Guided Decoder

To mitigate above ill-posed problem, we propose the Flow Matching Guided Decoder (FMGD). FMGD leverages a kinematic inductive bias based on Optimal Transport (OT). This design establishes the synergy of OT and ODE. **The ODE enables continuous motion prediction at arbitrary times, while OT provides a global velocity direction prior to constrain the ill-posed problem.**

Flow Matching Guided Feature Formulation: According to the Flow Matching theorem for OT, the efficient probability path between a source state x_0 and a target state x_1 is a straight line [21]. For any arbitrary time t, we define the target conditional latent velocity field u_t to construct this optimal path as $u_t(x|x_0, x_1) = x_1 - x_0$. This formulation minimizes kinetic energy [1] and significantly reduces numerical truncation errors during continuous integration [23]. The continuous latent velocity field v_θ is thus optimized via the conditional flow matching loss:

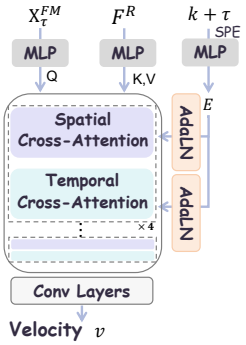

Figure 3: The details of LVP.

$$\mathcal{L}_{CFM}(\theta) = \mathbb{E}_{t,q(x_0,x_1),p_t(x|x_0,x_1)} \left[\|v_\theta(x,t) - u_t(x|x_0,x_1)\|_2^2 \right]. \tag{2}$$

To learn continuous latent velocity field v_θ within FMGD, we propose a local velocity prediction (LVP) module (Fig. 3). Inspired by [7], we decouple coarse motion and geometry using two independent decoder-DPT pairs [45], including a global velocity branch and a point reconstruction branch.

From global velocity branch, we extract anchor features $\mathbf{X} \in \mathbb{R}^{B\times T\times N\times D}$ at integer time 1 to $T+1$. These anchor features capture the global dynamic state of the sequence, implicitly embedding

motion information for any non-integer moment. Guided by the optimal transport (OT) prior for minimum-energy paths [41, 1], we construct FM-guided velocity features at any fractional timestamp $\tau \in (0, 1)$ via convex combination: $\mathbf{X}_\tau^{\text{FM}} = \tau \mathbf{X}_1 + (1 - \tau)\mathbf{X}_2$. We randomly sample a set $\{\tau\}$ in training, while $\{\tau\}$ set is determined based on the specified integration step size in inference. Specifically, $\mathbf{X}_2 = \{X_1, \ldots, X_T\}$ and $\mathbf{X}_1 = \{X_2, \ldots, X_{T+1}\}$ act as source state x_0 and target state x_1 defined in (2). This defines a probability path that minimizes the OT cost [21] to yield kinematically consistent features. Note that this OT-defined path lies in the anchor feature space and does not directly represent the 3D point trajectory. The point reconstruction branch extracts geometric features $\mathbf{F}^{\mathbf{R}} \in \mathbb{R}^{B\times T\times N\times D}$ (decoded into pointmaps via DPT) to provide geometric context for velocity prediction.

To decode velocities for an arbitrary fractional sequence of length L (equal to the size of the set $\{\tau\}$), the arbitrary time values $k + \tau$ (k is any integer time) are embedded into $\mathbf{E} \in \mathbb{R}^{B\times L\times d}$ via sinusoidal positional encodings (SPE) and an MLP. We then utilize $\mathbf{X}_\tau^{\text{FM}}$ as queries to attend to the geometric context $\mathbf{F}^{\mathbf{R}}$ via a Spatial-Temporal Cross-Attention module, accelerated by FlashAttention [4]. We perform spatial cross-attention along spatial axes while temporal cross-attention along time axis. Prior to each attention layer, queries are conditioned on $\mathbf{E}$ via Adaptive Layer Normalization (AdaLN) to inject continuous temporal awareness. Here, spatial attention corrects local trajectories against the global geometry, while temporal attention enforces smoothness across all timestamps.

Finally, the LVP outputs velocity field features $\mathbf{F}_{\text{vel}} \in \mathbb{R}^{B\times L\times T\times N\times D}$. We perform the CFM loss between $\mathbf{X}_1 - \mathbf{X}_2$ (as the value of $u_t(\cdot)$) and $\mathbf{F}_{\text{vel}}$ (as the value of $v_\theta(\cdot)$) via (2). Moreover, convolution layers decode $\mathbf{F}_{\text{vel}}$ into continuous 3D velocity field $\mathbf{V}_\theta(C(t))$, where $C(t)$ consists of the corresponding FM-guided velocity features, geometric features, and time embeddings. This allows us to query 3D velocities at integer and fractional timesteps. By conditioning FM-guided velocity features on geometric features and $\mathcal{L}_{CFM}$, the learned velocity field adheres to the underlying geometric structure. Integrating this structurally constrained field yields smooth, kinematically coherent trajectories. The reconstruction regularizes this field, while the resulting point tracking provides reliable motion guidance for reconstruction. This unified framework ultimately drives two tasks to mutually enhance each other's quality.

3.3 Integral-Consistency Training Strategy

Obtaining high-quality ground truth velocity for fractional timestamps is inherently difficult. While discrete anchors mitigate long-range drift, integrating velocities without direct supervision leaves the trajectory vulnerable to positional discontinuity (i.e., lacking C^0 continuity [6]) at integer boundaries. We address this by proposing an integral-consistency training strategy. Given integer timestep T, we take the pointmaps predicted by DPT at T, and integrate them forward to $T + 1$ using velocity field $\mathbf{V}_\theta(C(t))$. Specifically, $P(t)$ denotes the point trajectory state obtained during ODE integration. We solve the ODE using the efficient first order Euler method (details in A.6), which is highly effective because our kinematic inductive bias based on OT naturally reduces discretization errors [23, 21]. Next, the integrated pointmaps is then supervised by the GT pointmaps at $T + 1$. The process is formulated as follows:

$$\mathcal{L}_{ICT} = \mathbb{E}_{T,P}\left[\left\| P_{T+1} - \left(P_T + \int_T^{T+1} \mathbf{V}_\theta(C(t))dt \right) \right\|_2^2\right] \tag{3}$$

In the absence of intermediate velocity supervision, this training strategy bridges the gap by translating unobservable velocity fields into explicitly supervised pointmaps. Importantly, the loss also backpropagates through integration path directly into the upstream DPT and Decoder. This gradient flow explicitly regularizes the accuracy of the predicted pointmaps and velocities at integer timestamp.

Total Loss Function: Following DUSt3R [45], we also employ DUSt3R's loss functions. The total loss is $\mathcal{L} = \lambda_1\mathcal{L}_{ICT} + \lambda_2\mathcal{L}_{CFM} + \lambda_3\mathcal{L}_{DUSt3R}$, where $\lambda_1 : \lambda_2 : \lambda_3 = 1 : 1 : 1$.

Kinematics-Aware Benchmark. Existing point tracking protocols mainly evaluate at integer timestamps, lacking continuous-time metrics. To bridge this gap, we introduce a kinematics-aware benchmark to assess whether continuous trajectories maintain kinematic coherence. Constructed from Kubric dynamic scenes [11], our benchmark provides dense GT at both integer and fractional

Table 1: **World Coordinate 3D Point Tracking.** We report the performance on datasets PO, DR, ADT, and PStudio using the APD (APD_{3D} ↑) after global median alignment. We evaluate the accuracy of both all points and dynamic points. All the experiments of methods are carried out using the code and pre-trained model from their official GitHub. The best results are **bold**.

		All Points				Dynamic Points			
Category	**Methods**	PO	DR	ADT	PStudio	PO	DR	ADT	PStudio
Combinational	SpaTracker+RANSAC-Procrustes	44.03	55.01	50.87	52.05	53.77	58.58	66.49	52.05
	SpaTracker+MonST3R	47.65	55.49	51.95	50.16	58.61	59.21	69.94	50.16
Feed-forward	MonST3R	33.47	58.06	74.35	51.32	39.36	51.86	67.92	51.32
	SpaTracker	38.54	54.85	45.65	62.59	51.20	58.65	67.65	62.59
	POMATO	49.73	68.36	57.10	64.91	58.14	62.84	78.12	64.87
	St4RTrack	67.95	73.74	76.01	69.67	68.72	68.13	75.34	69.67
	TraceAnything	67.58	73.09	74.72	70.34	71.86	75.96	76.14	70.34
	V-DPM	81.27	75.32	85.72	78.76	82.57	78.28	73.45	78.76
	Uni4R (Ours)	**81.59**	**76.05**	**86.55**	**79.43**	**83.16**	**78.91**	**77.74**	**79.43**

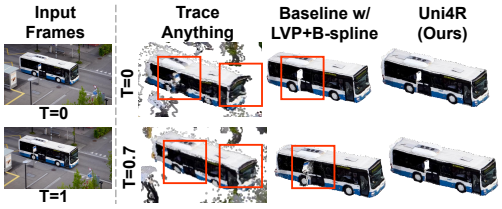

Figure 4: **Comparison with Continuous Modeling Variant and Trace Anything** [24] **at fractional timestamp**. We visualize the dynamic part. With integer time frames as input, baselines produce unstable geometry structures with distorted windows and wheels at fractional timestep, our method maintains the geometric coherence and local geometric structures of the dynamic components.

Table 2: **World Coordinate 4D Reconstruction.** We report performance on PO and TUM-Dynamics after global median scaling. All the experiments of methods are carried out using the code and pre-trained model from their official GitHub. The best results are marked in bold.

		Point Odyssey		TUM-Dynamics	
Category	Method	APD↑	EPE↓	APD↑	EPE↓
w/ Global Align.	DUSt3R+GA	43.90	0.609	70.49	0.315
	MASt3R+GA	60.44	0.403	68.38	0.519
	MonST3R+GA	72.31	0.263	63.87	0.343
Feedforward	DUSt3R	45.79	0.639	72.26	0.289
	MASt3R	56.90	0.464	66.22	0.551
	MonST3R	68.25	0.304	61.38	0.365
	POMATO	66.50	0.385	49.80	0.509
	St4RTrack	78.73	0.241	83.42	0.185
	V-DPM	69.05	0.342	75.92	0.273
	Uni4R (Ours)	**80.71**	**0.222**	**84.36**	**0.181**

timestamps. We evaluate kinematic coherence via two aspects, temporal smoothness Mean Acceleration Magnitude [9] (MAM) and spatial structural stability based on integer and fractional timestamps, including Edge Length Variance (ELV), Variance of Local Pairwise Distances [37] (VLPD) and APD. Moreover, we also compute GT's MAM, ELV, and VLPD to evaluate absolute deviation Δ between each method and GT. More details are provided in A.4.

4 Experiments

Datasets and Training. We train our model on 8 H800 GPUs using three datasets containing dynamic scene and camera motions: Point Odyssey [70], Dynamic Replica [15], and Kubric [11], which provide ground-truth at integer timestamps. During evaluation, we utilize an ODE's integration step size of 0.1 between adjacent integer frames for our models. Detailed studies regarding the impact of step size on accuracy and inference time, training details, are available in A.3 and A.7.3.

4.1 Point Tracking in World Coordinates

Benchmark. Following St4RTrack [7], we employ their proposed benchmark for point tracking in world coordinates. Specifically, the benchmark leverages two real-world datasets, including Aerial Digital Twin (ADT) [30] and Panoptic Studio (PStudio) [13], and two additional synthetic test sets from Point Odyssey (PO) and Dynamic Replica (DR).

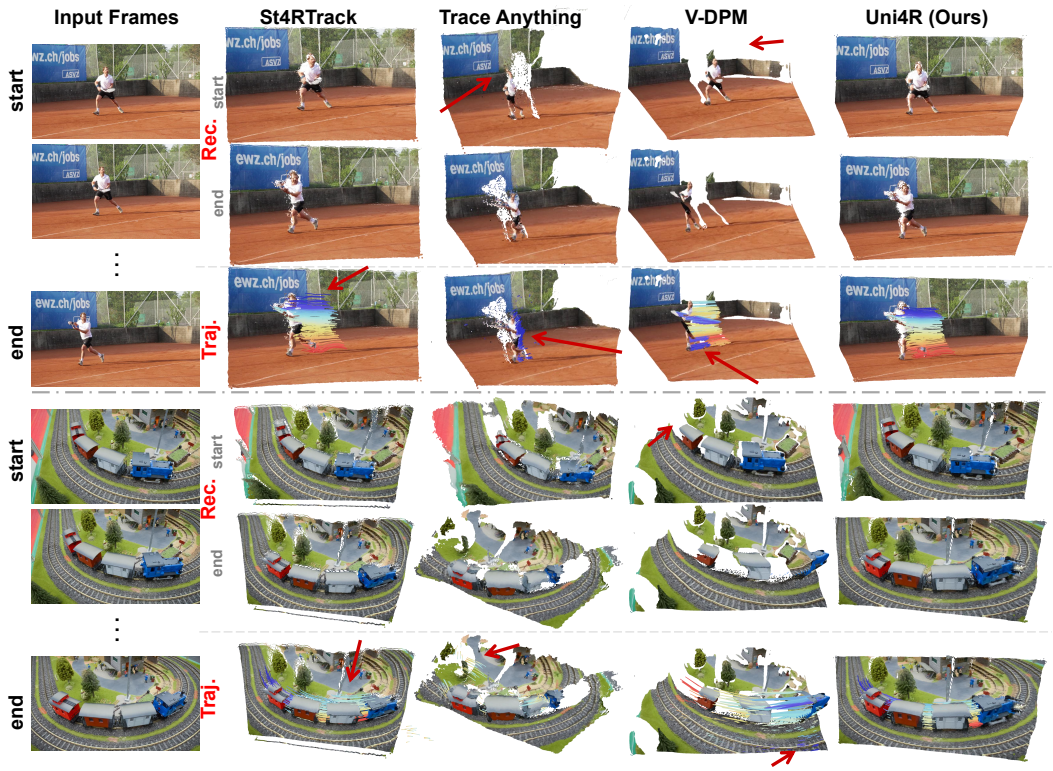

Figure 5: **Qualitative Results of 4D Reconstruction and Point Tracking on real-world dataset DAVIS** [31]. We visualize our Uni4R and baselines, including St4RTrack [7] (we apply their test-time adaptation), Trace Anything [24], V-DPM [39]. Our Uni4R effectively captures the geometric structure and motion trajectories in complex motion scenarios, whereas baselines produce incomplete geometry reconstruction (e.g., background and the moving train), and trajectories drift.

Metrics. Following St4RTrack [7] and TAPVid-3D protocol [19], we use Average percent of Points within Delta (APD) for evaluation. Following St4RTrack [7], we first align the predicted point trajectories with the ground truth through normalizing them with their global median. Then we compute the prediction error and we measure the percentage of points whose error falls below thresholds $\delta_{3D} \in \{0.1m, 0.3m, 0.5m, 1.0m\}$ over the first 64 frames. Let $\hat{\mathbf{P}}_t^i$ denote the i-th point at time t, and $\mathbf{P}_t^i$ is its corresponding ground-truth location. Formally, $\text{APD}_{3D} \equiv \sum_{i,t} \mathbb{1}\left(\left\|\hat{\mathbf{P}}_t^i - \mathbf{P}_t^i\right\| < \delta_{3D}\right)$, where $\mathbb{1}$ denotes indicator function and $\|\cdot\|$ is Euclidean norm.

Baselines. Since we perform 3D tracking in world coordinates following St4RTrack, we first compare against the camera coordinate method SpatialTracker [49] and a dynamic 3D reconstruction method MonST3R [65]. In addition, we compare two combinational baselines in world coordinate 3D tracking, which applies Procrustes alignment and RANSAC to the camera coordinate 3D tracks predicted by SpatialTracker to offset the camera motion, and the second is the model which leverages the camera poses predicted by the MonST3R [8] to compensate for camera motion [7]. Moreover, we also compare a dynamic 3D tracking method POMATO [67], and the recent dynamic 3D tracking method TraceAnything [24], St4RTrack [7], and V-DPM [39].

Results. As shown in Tab. 1, we achieve SOTA performance in all test datasets and metrics. The performance gap highlights the lack of explicit kinematic priors in baselines. Thus their predictions at integer timestamps remain inherently inaccurate under complex motion scenarios. As shown in Fig. 5, baselines suffer from trajectory drift. In contrast, our framework resolves this by synergizing OT and ODE to learn a continuous velocity field. The continuous velocity field acts as a robust kinematic prior. By leveraging this prior, Uni4R yields significantly more accurate spatial-temporal correspondences to predict point tracking. Besides, in Tab. 3. our Uni4R achieves a highly efficient inference speed (1.57s), trailing only St4RTrack. Although Uni4R has slightly higher VRAM footprint compared to Trace Anything is heavily outweighed by a massive performance margin.

Table 3: **Efficiency comparison on 80-frame input**. We test on the same H800 GPU. Our Uni4R employs 10 integration steps between adjacent integer frames.

Methods	Time (s)	VRAM (GB)
St4RTrack	1.34	24.06
Trace Anything	4.69	35.71
V-DPM	194.14	75.27
Uni4R (Ours)	1.57	43.67

Table 4: **Kinematic Coherence in Continuous Time.** We report the performance on kinematics-aware Benchmark after global median alignment.The best results are marked in bold.

Methods	Temporal Kinematic	Spatial Topology		
	MAM ↓ / Δ ↓	APD_{3D} ↑	ELV ↓ / Δ ↓	VLPD ↓ / Δ ↓
Ground Truth (GT)	6.99 / -	-	10.2316 / -	0.0003 / -
Baseline w/ B-Spline	15.93 / 8.94	75.04	10.3107 / 0.0791	0.0274 / 0.0271
Baseline w/ LVP + B-Spline	13.75 / 6.76	76.37	10.2989 / 0.0673	0.0146 / 0.0143
Baseline w/ LVP + ODE	13.64 / 6.65	76.89	10.2970 / 0.0654	0.0115 / 0.0112
TraceAnything	7.89 / 0.90	77.90	10.2553 / 0.0237	0.0015 / 0.0012
Uni4R (Ours)	**7.27 / 0.28**	**79.53**	**10.2437 / 0.0121**	**0.0004 / 0.0001**

4.2 4D Reconstruction

Benchmark. Following St4RTrack [7], we evaluate the quality of 4D reconstruction on synthetic dataset Point Odyssey, and real-world dataset TUM-Dynamics.

Metrics. Following St4RTrack, we compare the reconstructed 3D point clouds to the ground truth using the Average percent of Points within Distance (APD) and End-Point Error (EPE) metrics. Before evaluation, we filter out ambiguous floating points from the ground-truth data and align the point clouds for each sequence using the median scale.

Baselines. We compare our Uni4R with baselines, including MonST3R, MASt3R, DUSt3R, POMATO, recent methods St4RTrack, and V-DPM. Following St4RTrack, for the global align (GA), we apply global alignment on baselines; For the feedforward baselines, we construct image pairs of a video that align all frames to a common anchor frame.

Results. As shown in Tab. 2, we achieve SOTA performance in all test datasets and metrics. Because baselines ignore constructing or using kinematic priors, thus their predictions at integer timestamps remain fragmented geometry and topological distortions, as shown in Fig. 5. In contrast, our Uni4R constructs and leverages kinematic prior from continuous velocity field, Uni4R effectively captures the geometric structure in complex motion scenarios.

4.3 Continuous-Time Kinematic Coherence

Benchmark. We employ our kinematics-aware benchmark to evaluate our Uni4R and some baselines on the Kinematic Coherence in continuous time. The benchmark includes 100 diverse dynamic scene. Each dynamic scenes has 24 integer frames. And we render 24 integer and 230 fractional timestamps' depth, masks, camera poses and ground-truth point trajectories for each dynamic scene.

Metrics. As we introduce in Sec. 3.3, we employ MAM, ELV, VLPD and APD to evaluate the kinematics coherence in continuous time, and also report the MAM, ELV, VLPD absolute deviation Δ between each method and GT. We evaluate on both integer and fractional timestamps.

Baselines. Since most recent works focus on integer timestamps, only few works perform on fractional timestamps, e.g., TraceAnything [24]. Therefore, we compare our Uni4R with TraceAnything. Besides, we also construct a series baseline, which replaces FMGD with two decoder and two DPT. One decoder and DPT predicts the reconstructed point maps, while another predicts velocities on integer timestamps. Then the velocities will be interpolated via B-Spline (Baseline w/ B-Spline) on fractional timestamps; or employ LVP to predict control points to control the velocity (Baseline w/ LVP + B-Spline) on fractional timestamps; or employ MLPs and ODE to predict velocities (Baseline w/ LVP + ODE) on fractional timestamps.

Results. As shown in Tab. 4, we achieve SOTA performance in all metrics, proving ours superior kinematic coherence. Our Uni4R achieves the closest overall agreement with GT among the compared baselines. Besides, our low MAM verifies smooth continuous trajectories, while low ELV and VLPD, and high APD confirm the preservation of local geometric structures. As shown in Fig. 4, baselines lack kinematic priors to constrain continuous geometry deformations, causing severe distortions like warped wheels and stretched windows. In contrast, Uni4R learns a continuous velocity field that acts as a robust kinematic prior. This kinematic prior preserves local geometric structures, ensuring topologically consistent geometric structures at any fractional timestamp.

4.4 Ablation Study

Basic Ablations: To validate the effect of our proposed components, we progressively remove the core components within FMGD of Uni4R in Tab. 5. Starting from Uni4R, removing the continuous flow matching loss (w/o $\mathcal{L}_{CFM}$) leads to a notable performance drop in all tasks due to the lack of continuous velocity regularization. Further removing the FM-guided Velocity Features (w/o FVF) leads that causes the model to lose its kinematic inductive bias, causing a continued decline in all tasks accuracy, especially kinematic coherence. Next, we remove integral-consistency training strategy (w/o ICT) means remove the end-to-end velocity supervision, resulting in a performance degradation. It confirms that simply applying an ODE is insufficient because of ill-posed problem, leading to low performance compared to previous model. Finally, we omit the Local Velocity Prediction module (w/o LVP) and replace it using simple cross-attention, which cannot accurate captures geometry and motion information to predict precise velocity, yielding the lowest overall performance. Besides, we replace Uni4R's anchor-based ODE integration by global-based ODE (model G), which leads to a performance drop in all tasks due to the accumulation of integration errors. It validates that we discuss in Sec. 3.1, anchor-based ODE mitigates the accumulation of integration errors.

Comparison with Continuous Modeling Variant: In Tab. 5 and Fig. 4, we further evaluate our Uni4R against continuous modeling variants based on Baseline. The Baseline replaces FMGD with two decoder and two DPT. One decoder and DPT predict reconstructed point maps, while another predicts velocities on integer timestamps. Based on Baseline, model B which first employ LVP to predict control points then applying B-spline yields marginal improvements in both two tasks. But there exists a large performance gap compared to our Uni4R and model E (model E means to directly apply ODE without ICT). Because model B focuses on geometric curve fitting instead of leveraging kinematic priors for dynamic modeling.

Table 5: **Ablation Experiments** on 4D Reconstruction and Point Tracking (report APD_{3D} ↑), and continuous-time kinematic coherence (named KC, we report ELV ↓). **Top**: we present the baselines using continuous modeling variant. **Middle**: We compare models which cumulatively (from C to F) remove core components within FMGD of Uni4R. **Below**: We replace Uni4R's anchor-based ODE integration by global-based ODE, which directly integrate velocity from time 0 to last.

Methods	Point Tracking				Reconstruction		KC
	PO	DR	ADT	PStudio	PO	TUM	
A. Baseline	74.31	74.19	79.83	74.66	77.94	81.05	/
B. Baseline w/ LVP + B-Spline	74.99	74.41	81.26	76.21	78.38	82.14	10.2989
Uni4R (Ours)	**81.59**	**76.05**	**86.55**	**79.43**	**80.71**	**84.36**	**10.2437**
C. w/o $\mathcal{L}_{CFM}$	80.16	75.71	85.43	78.94	79.88	84.11	10.2569
D. w/o FVF	79.36	75.43	83.35	77.12	78.95	83.72	10.2728
E. w/o ICT	75.27	74.65	81.50	76.39	78.52	82.93	10.2951
F. w/o LVP	74.68	74.48	80.63	75.07	78.33	81.76	10.3045
G. Global-based ODE	79.10	75.57	85.38	79.02	78.73	83.76	10.2819

Mutual Enhancement of Reconstruction and Tracking: To investigate the mutual benefit between 4D reconstruction and point tracking within Uni4R, we evaluate models under different training strategies (with the same pre-trained models) in Tab. 6. Training only the velocity branch (model M1) or the reconstruction branch (model M2) leads to lower performance than our Uni4R both in two tasks. Because separate optimization breaks the inherent coupling between geometry and motion. In contrast, joint training (our Uni4R) unlocks the full potential of learned continuous velocity field. Specifically, the continuous velocity filed acts as a joint kinematic prior. The accurate point tracking provides kinematic guidance for geometric deformation, while high-quality 4D reconstruction regularizes the trajectory space. This kinematics-aware and unified framework allows the two tasks to mutually enhance each other, yielding superior performance in both two tasks.

Table 6: **Ablation Experiments** on the Mutual Enhancement of Reconstruction and Tracking. We compare the models whether train velocity branch (vel.) or train reconstruction branch (rec.). Each model loads the same pre-trained weights before training.

Methods	Train		Point Tracking				Reconstruction	
	Vel.	Rec.	PO	DR	ADT	PStudio	PO	TUM
M1	✓	✗	79.74	75.18	84.89	78.67	78.89	83.01
M2	✗	✓	78.72	74.05	82.94	78.01	79.78	83.98
Ours	✓	✓	**81.59**	**76.05**	**86.55**	**79.43**	**80.71**	**84.36**

Comparison of First Order Euler and RK4: We compare the performance of the first order Euler method against the 4th-order Runge-Kutta (RK4) method in Tab. 7. The results demonstrate that RK4 achieves almost identical performance to the first order Euler method in both reconstruction and tracking tasks. However, RK4 is substantially more computationally expensive, requiring four

Table 7: **Comparison of First Order Euler and RK4** on 4D Reconstruction and Point Tracking (report APD_{3D} ↑), and continuous-time kinematic coherence (named KC, we report ELV under fractional timestamps ↓). The inference time is tested based on 80-frame input.

Methods	Point Tracking				Reconstruction		KC	Inference Time (s)
	PO	DR	ADT	PStudio	PO	TUM		
RK4	81.62	76.01	86.67	79.45	80.69	84.42	10.2419	4.09
Ours (First Order Euler)	**81.59**	**76.05**	**86.55**	**79.43**	**80.71**	**84.36**	**10.2437**	**1.57**

function evaluations per step. Consequently, we employ the Euler method in our model, as it provides sufficient integration precision while being significantly more efficient.

5 Limitations

While our method demonstrates strong capabilities in continuous-time 4D reconstruction and point tracking, there are several limitations and future work for exploration:

Lacks real-world continuous-time datasets. Although we can extract the continuous-time data from synthetic data such as Kubric, obtaining real-world continuous-time data still important for better robustness and accurate kinematics evaluations of models. However, acquiring such dense data for complex real-world dynamic scenes remains an open challenge within the community. Therefore, we plan to introduce event-based vision sensors to process real-world data at arbitrary timestamps.

Limited scale of training datasets. Limited to our computational resources, the scale of the training dataset is relatively moderate. However, our extensive experiments have demonstrated that the proposed method already achieves remarkable robustness and strong generalization capabilities despite this constraint. This validates the effectiveness of proposed method. In the future, we will expand the scale of training datasets to include scenarios with extreme non-rigid deformations and fluid dynamics, thereby extending our method to handle more complex topological and kinematical properties changes in real-world scenarios.

Challenging severe ego-motion scenes required further investigation. Although Uni4R is trained with 16-frame sequences, our experiments in A.7.4 demonstrate its generalization to 64-frame sequences. However, sequences with severe ego-motion or complete loss of visual co-visibility remain challenging settings that require further investigation.

6 Conclusion

In this paper, we propose Uni4R, a novel framework that unifies 4D reconstruction and point tracking. By synergizing Optimal Transport (OT) and Ordinary Differential Equations (ODE), our method learns a continuous velocity field that serves as a robust kinematic prior, ensuring kinematic coherence at any arbitrary timestamp. A key component is the Flow Matching Guided Decoder (FMGD), which effectively leverages FM theory to decode these continuous velocities. Furthermore, our integral-consistency training strategy addresses the absence of fractional ground truth velocities, enabling end-to-end supervision directly from integer frames. Our method achieves state-of-the-art performance across two tasks, and our new kinematics-aware benchmark.

Acknowledgments and Disclosure of Funding

This work is supported by the Science and Technology Development Fund of Macau Project (grant No. 0096/2023/RIA2, 0044/2024/AGJ, 0140/2024/AGJ, 0084/2024/RIB2), and is partially supported by the AI Supercomputing Center of Macau University of Science and Technology (MUST). We acknowledge the technical support provided by the Information Technology Development Office of MUST. We also thank the anonymous reviewers for their valuable comments and suggestions. The authors declare no competing interests.

References

[1] Benamou , J.-D. & Brenier , Y. (2000) A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik* **84**(3):375–393.

[2] Chen , C., Li , C., Chen , L., Wang , W., Pu , Y., & Duke , L. C. (2018.) Continuous-time flows for efficient inference and density estimation. In *International Conference on Machine Learning* pages 824–833. PMLR.

[3] Chen , R. T., Rubanova , Y., Bettencourt , J., & Duvenaud , D. K. (2018.) Neural ordinary differential equations. *Advances in neural information processing systems* **31**.

[4] Dao , T., Fu , D., Ermon , S., Rudra , A., & Ré , C. (2022) Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems* **35**:16344–16359.

[5] Dupont , E., Doucet , A., & Teh , Y. W. (2019) Augmented neural odes. *Advances in neural information processing systems* **32**.

[6] Farin , G. E. (2002) *Curves and surfaces for CAGD: a practical guide*. *Curves and surfaces for CAGD: a practical guide*: Morgan Kaufmann.

[7] Feng , H., Zhang , J., Wang , Q., Ye , Y., Yu , P., Black , M. J., Darrell , T., & Kanazawa , A. (2025) St4rtrack: Simultaneous 4d reconstruction and tracking in the world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* pages 8503–8513.

[8] Fischler , M. A. & Bolles , R. C. (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* **24**(6):381–395.

[9] Flash , T. & Hogan , N. (1985) The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of neuroscience* **5**(7):1688–1703.

[10] Grathwohl , W., Chen , R. T., Bettencourt , J., Sutskever , I., & Duvenaud , D. (2018) Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*

[11] Greff , K., Belletti , F., Beyer , L., Doersch , C., Du , Y., Duckworth , D., Fleet , D. J., Gnanapragasam , D., Golemo , F., Herrmann , C., & others (2022) Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* pages 3749–3761.

[12] Hunyuan3D , T., Zhang , B., Guo , C., Liu , H., Yan , H., Shi , H., Huang , J., Yu , J., Li , K., Wang , P., & others (2025) Hunyuan3d-omni: A unified framework for controllable generation of 3d assets. *arXiv preprint arXiv:2509.21245*

[13] Joo , H., Liu , H., Tan , L., Gui , L., Nabbe , B., Matthews , I., Kanade , T., Nobuhara , S., & Sheikh , Y. (2015) Panoptic studio: A massively multiview system for social motion capture. In *2015 IEEE International Conference on Computer Vision (ICCV)* pages 3334–3342.

[14] Kanazawa , A., Zhang , J. Y., Felsen , P., & Malik , J. (2019) Learning 3d human dynamics from video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* pages 5614–5623.

[15] Karaev , N., Rocco , I., Graham , B., Neverova , N., Vedaldi , A., & Rupprecht , C. (2023) Dynamicstereo: Consistent dynamic depth from stereo videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* pages 13229–13239.

[16] Karaev , N., Rocco , I., Graham , B., Neverova , N., Vedaldi , A., & Rupprecht , C. (2024) Cotracker: It is better to track together. In *European conference on computer vision* pages 18–35. Springer.

[17] Karaev , N., Makarov , Y., Wang , J., Neverova , N., Vedaldi , A., & Rupprecht , C. (2025) Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* pages 6013–6022.

[18] Kerbl , B., Kopanas , G., Leimkühler , T., Drettakis , G., & others (2023) 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.* **42**(4):139–1.

[19] Koppula , S., Rocco , I., Yang , Y., Heyward , J., Carreira , J., Zisserman , A., Brostow , G., & Doersch , C. (2024) Tapvid-3d: A benchmark for tracking any point in 3d. *Advances in Neural Information Processing Systems* **37**:82149–82165.

[20] Leroy , V., Cabon , Y., & Revaud , J. (2024) Grounding image matching in 3d with mast3r. In *European conference on computer vision* pages 71–91. Springer.

[21] Lipman , Y., Chen , R. T., Ben-Hamu , H., Nickel , M., & Le , M. (2022) Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*

[22] Liu , R., Wu , R., Van Hoorick , B., Tokmakov , P., Zakharov , S., & Vondrick , C. (2023) Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision* pages 9298–9309.

[23] Liu , X., Gong , C., & Liu , Q. (2022) Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*

[24] Liu , X., Xiao , Y., Chen , D. Y., Feng , J., Tai , Y.-W., Tang , C.-K., & Kang , B. (2025) Trace anything: Representing any video in 4d via trajectory fields. *arXiv preprint arXiv:2510.13802*

[25] Long , X., Guo , Y.-C., Lin , C., Liu , Y., Dou , Z., Liu , L., Ma , Y., Zhang , S.-H., Habermann , M., Theobalt , C., & others (2024) Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* pages 9970–9980.

[26] Lu , J., Xu , J., Hu , W., Zhu , R., Zhao , C., Yeung , S.-K., Shan , Y., & Liu , Y. (2026) Track4world: Feedforward world-centric dense 3d tracking of all pixels. In *European Conference on Computer Vision* pages 184–204. Springer.

[27] Lyu , Y., Geng , C., Dharmarajan , K., Zhang , Y., Alzayer , H., Wu , S., & Wu , J. (2026) Choreographing a world of dynamic objects. *arXiv preprint arXiv:2601.04194*

[28] Melas-Kyriazi , L., Rupprecht , C., & Vedaldi , A. (2023) Pc 2: Projection-conditioned point cloud diffusion for single-image 3d reconstruction. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* pages 12923–12932. IEEE.

[29] Norcliffe , A., Bodnar , C., Day , B., Simidjievski , N., & Liò , P. (2020) On second order behaviour in augmented neural odes. *Advances in neural information processing systems* **33**:5911–5921.

[30] Pan , X., Charron , N., Yang , Y., Peters , S., Whelan , T., Kong , C., Parkhi , O., Newcombe , R., & Ren , Y. C. (2023) Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* pages 20133–20143.

[31] Perazzi , F., Pont-Tuset , J., McWilliams , B., Van Gool , L., Gross , M., & Sorkine-Hornung , A. (2016) A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* pages 724–732.

[32] Poole , B., Jain , A., Barron , J. T., & Mildenhall , B. (2022) Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*

[33] Qian , S., Zhang , G., Wu , S., & Cremers , D. 2026.

[34] Ren , J., Xie , K., Mirzaei , A., Liang , H., Zeng , X., Kreis , K., Liu , Z., Torralba , A., Fidler , S., Kim , S. W., & others (2024) L4gm: Large 4d gaussian reconstruction model. *Advances in Neural Information Processing Systems* **37**:56828–56858.

[35] Shi , R., Chen , H., Zhang , Z., Liu , M., Xu , C., Wei , X., Chen , L., Zeng , C., & Su , H. (2023) Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*

[36] Singer , U., Polyak , A., Hayes , T., Yin , X., An , J., Zhang , S., Hu , Q., Yang , H., Ashual , O., Gafni , O., & others (2022) Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*

[37] Sorkine , O., Alexa , M., & others (2007) As-rigid-as-possible surface modeling. In *Symposium on Geometry processing 4*, pp. 109–116.

[38] Sucar , E., Lai , Z., Insafutdinov , E., & Vedaldi , A. (2025) Dynamic point maps: A versatile representation for dynamic 3d reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* pages 7295–7305.

[39] Sucar , E., Insafutdinov , E., Lai , Z., & Vedaldi , A. (2026) V-dpm: 4d video reconstruction with dynamic point maps. *arXiv preprint arXiv:2601.09499*

[40] Tang , J., Chen , Z., Chen , X., Wang , T., Zeng , G., & Liu , Z. (2024) Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision* pages 1–18. Springer.

[41] Tong , A., Fatras , K., Malkin , N., Huguet , G., Zhang , Y., Rector-Brooks , J., Wolf , G., & Bengio , Y. (2023) Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*

[42] Vedder , K., Peri , N., Khatri , I., Li , S., Eaton , E., Kocamaz , M., Wang , Y., Yu , Z., Ramanan , D., & Pehserl , J. (2025) Neural eulerian scene flow fields. In *International Conference on Learning Representations 2025*, pp. 99546–99564.

[43] Wan , T., Wang , A., Ai , B., Wen , B., Mao , C., Xie , C.-W., Chen , D., Yu , F., Zhao , H., Yang , J., & others (2025) Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*

[44] Wang , J., Chen , M., Karaev , N., Vedaldi , A., Rupprecht , C., & Novotny , D. (2025.) Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference* pages 5294–5306.

[45] Wang , S., Leroy , V., Cabon , Y., Chidlovskii , B., & Revaud , J. (2024) Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* pages 20697–20709.

[46] Wang , Y., Zhou , J., Zhu , H., Chang , W., Zhou , Y., Li , Z., Chen , J., Pang , J., Shen , C., & He , T. (2025.) π^3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*

[47] Wang , Y., Wang , W., Zhang , S., Yu , X., & Chen , Z. (2026) Mcgs: Markov chain gaussian splatting for dynamic scenes reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence 40*, pp. 10341–10348.

[48] Wu , Z., Yu , C., Jiang , Y., Cao , C., Wang , F., & Bai , X. (2024) Sc4d: Sparse-controlled video-to-4d generation and motion transfer. In *European Conference on Computer Vision* pages 361–379. Springer.

[49] Xiao , Y., Wang , Q., Zhang , S., Xue , N., Peng , S., Shen , Y., & Zhou , X. (2024) Spatialtracker: Tracking any 2d pixels in 3d space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* pages 20406–20417.

[50] Xiao , Y., Wang , J., Xue , N., Karaev , N., Makarov , Y., Kang , B., Zhu , X., Bao , H., Shen , Y., & Zhou , X. (2025) Spatialtrackerv2: Advancing 3d point tracking with explicit camera motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* pages 6726–6737.

[51] Xie , H., Yao , H., Sun , X., Zhou , S., & Zhang , S. (2019) Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* pages 2690–2698. IEEE.

[52] Yang , J., Sax , A., Liang , K. J., Henaff , M., Tang , H., Cao , A., Chai , J., Meier , F., & Feiszli , M. (2025.) Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *Proceedings of the Computer Vision and Pattern Recognition Conference* pages 21924–21935.

[53] Yang , L., Zhu , Z., Lin , X., Nong , J., & Liang , Y. (2023) Long-range grouping transformer for multi-view 3d reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* pages 18257–18267.

[54] Yang , L., Liu , C., Zhu , Z., Liu , A., Ma , H., Nong , J., & Liang , Y. (2025.) Not all frame features are equal: Video-to-4d generation via decoupling dynamic-static features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* pages 7494–7504.

[55] Yang , L., Liu , J., Hu , J., Guan , C., Huang , H., Yi , F., Zhang , C., & Liang , Y. (2026) Spatial-temporal state propagation autoregressive model for 4d object generation. *arXiv preprint arXiv:2602.18830*

[56] Yao , C.-H., Xie , Y., Voleti , V., Jiang , H., & Jampani , V. (2025) Sv4d 2.0: Enhancing spatio-temporal consistency in multi-view video diffusion for high-quality 4d generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* pages 13248–13258.

[57] Yu , X., Li , J., Wong , C.-C., Vong , C.-M., & Liang , Y. (2025) Facnet: Feature alignment fast point cloud completion network. *Computational Visual Media* **11**(1):141–157.

[58] Yu , X., Liu , A., Qiang , S., Ma , H., Yang , L., Wang , Y., Rao , Z., & Liang , Y. (2026.) Pointcsp: Cross-sample semantic propagation and stability preservation in self-supervised point cloud learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* pages 10016–10026.

[59] Yu , X., Liu , A., Qiang , S., Wang , Y., Ma , H., & Liang , Y. (2026.) Pointmc: Multi-view consistent encoding and center-global feature fusion for point clouds understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence 40*, pp. 12169–12177.

[60] Yu , X., Yang , L., Mo , H., Ma , H., Kai , F., Liu , A., & Liang , Y. (2026.) Pointchr: Point cloud analysis via curvature-aware hyperbolic rectification. In *Forty-third International Conference on Machine Learning*

[61] Yuan , Y. & Kitani , K. (2020) Dlow: Diversifying latent flows for diverse human motion prediction. In *European Conference on Computer Vision* pages 346–364. Springer.

[62] Yuan , Y., Wickremasinghe , T., Nadir , Z., Wang , X., Chi , Y., & Chan , S. H. (2025) Seeu: Seeing the unseen world via 4d dynamics-aware generation. *arXiv preprint arXiv:2512.03350*

[63] Zeng , Y., Jiang , Y., Zhu , S., Lu , Y., Lin , Y., Zhu , H., Hu , W., Cao , X., & Yao , Y. (2024) Stag4d: Spatial-temporal anchored generative 4d gaussians. In *European Conference on Computer Vision* pages 163–179. Springer.

[64] Zhang , C., Moing , G. L., Koppula , S., Rocco , I., Momeni , L., Xie , J., Sun , S., Sukthankar , R., Barral , J. K., Hadsell , R., & others (2025.) Efficiently reconstructing dynamic scenes one d4rt at a time. *arXiv preprint arXiv:2512.08924*

[65] Zhang , J., Herrmann , C., Hur , J., Jampani , V., Darrell , T., Cole , F., Sun , D., & Yang , M.-H. (2024.) Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*

[66] Zhang , K., Bi , S., Tan , H., Xiangli , Y., Zhao , N., Sunkavalli , K., & Xu , Z. (2024.) Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision* pages 1–19. Springer.

[67] Zhang , S., Ge , Y., Tian , J., Xu , G., Chen , H., Lv , C., & Shen , C. (2025.) Pomato: Marrying pointmap matching with temporal motions for dynamic 3d reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* pages 5680–5689.

[68] Zhao , Y., Yan , Z., Xie , E., Hong , L., Li , Z., & Lee , G. H. (2023) Animate124: Animating one image to 4d dynamic scene. *arXiv preprint arXiv:2311.14603*

[69] Zhao , Z., Lai , Z., Lin , Q., Zhao , Y., Liu , H., Yang , S., Feng , Y., Yang , M., Zhang , S., Yang , X., & others (2025) Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*

[70] Zheng , Y., Harley , A. W., Shen , B., Wetzstein , G., & Guibas , L. J. (2023) Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* pages 19855–19865.

[71] Zheng , Y., Li , X., Nagano , K., Liu , S., Hilliges , O., & De Mello , S. (2024) A unified approach for text-and image-guided 4d scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* pages 7300–7309.

[72] Zhu , Z., Yang , L., Li , N., Jiang , C., & Liang , Y. (2023.) Umiformer: Mining the correlations between similar tokens for multi-view 3d reconstruction. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* pages 18180–18189. IEEE.

[73] Zhu , Z., Yang , L., Lin , X., Yang , L., & Liang , Y. (2023.) Garnet: Global-aware multi-view 3d reconstruction network and the cost-performance tradeoff. *Pattern Recognition* **142**:109674.

Appendix Contents

A Technical Appendices and Supplementary Material **15**

A.1 Video Results 15

A.2 Details of Cross-Frame Feature Interaction within Visual Encoding and Decoders . 15

A.3 Training Details 15

A.4 Details of Kinematics-aware Benchmark 15

A.5 Network Architecture Details of LVP 17

A.6 Clarification of the Numerical Solver for ODE 17

A.7 More Experiments 18

B Kinematic Grounding of the OT Probability Path **21**

A Technical Appendices and Supplementary Material

A.1 Video Results

We recommend watching the video (.mp4) in the supplementary materials, which shows the results of our method.

A.2 Details of Cross-Frame Feature Interaction within Visual Encoding and Decoders

The input of Uni4R is the complete video frame sequence without camera poses. After the shared ViT encoder, the decoders in both branches receive two sequences, where sequence-1 corresponds to the reference frame (the first frame) and sequence-2 contains the video frames from the first to the last frame. The decoders in the point reconstruction branch and the global velocity branch perform bidirectional cross-attention between sequence-1 and sequence-2, introducing multiview information into geometric features and cross-frame temporal information into velocity features.

A.3 Training Details

Training. The ViT Encoder, Decoder and DPT will load the weight of MASt3R [20] when starting training. During training, we use the AdamW optimizer with a learning rate of 4×10^{-5} and batch size of 8. We train our model for 180 epochs on 8 H800 GPUs. At each iteration, we sample 16 frames of each video as input.

Training datasets. Following [7, 24], for each dataset, we calculate the sparse and masked supervision for point tracking by scene mesh vertices. We employ per-frame depth maps and camera ground-truth for reconstruction supervision.

A.4 Details of Kinematics-aware Benchmark

A.4.1 Rendering Kubric data

Relying on the Kubric simulation engine [11], we design a pipeline to extract trajectory at integer and fractional timestamps. This pipeline directly queries the underlying physics engine to obtain kinematic states at arbitrary timestamps, providing ground truth (both integer and fractional timestamps) for our benchmark. The data generation process is formulated as follows:

We define $t = 0$ as the anchor frame. For a given dynamic object in the scene, we first obtain its segmentation mask $\mathbf{M}_{dyn}$ and the raw depth map $\mathbf{D}_0$ at $t = 0$. Using the camera intrinsic matrix $\mathbf{K}$ and the camera-to-world extrinsic matrix $\mathbf{E}^0_{c2w}$, the 2D pixels (u, v) on the dynamic object are unprojected into 3D points in the world coordinate system, denoted as $\mathbf{P}^0_w$.

Subsequently, we extract the Object-to-World pose matrix $\mathbf{M}_{o2w}^{0}$ of the object at $t = 0$. To establish a consistent foundation for subsequent kinematic calculations, we inversely transform and bind these surface points to the object's canonical local space:

$$\mathbf{P}_{local} = (\mathbf{M}_{o2w}^{0})^{-1}\mathbf{P}_{w}^{0} \tag{4}$$

Then, we capture the complex dynamics at fractional timestamps. Specifically, for a given integer frame interval $t \to t+1$, we divide it into L sub-frame steps (e.g., $L = 10$). For each fractional timestamp $t + \frac{l}{L}$ (where $l \in [0, L-1]$), we instruct the Blender physics engine to explicitly update to this corresponding sub-frame state.

In this state, we query the physics engine to return the body pose $\mathbf{M}_{o2w}^{t,l}$ of the dynamic object at that sub-timestamp. Because this pose is computed by the underlying dynamics solver (e.g., PyBullet), it more faithfully reflects the object's acceleration, collision responses, and rotational trajectories, thereby effectively mitigating common non-physical interpolation artifacts.

While we obtain pose at fractional timestamps, the local points are transformed back into the global space for the corresponding timestamp, and subsequently projected into the coordinate system of the anchor camera ($t = 0$) to facilitate evaluation:

$$\mathbf{P}_{cam0}^{t,l} = (\mathbf{E}_{c2w}^{0})^{-1}\mathbf{M}_{o2w}^{t,l}\mathbf{P}_{local}$$

Ultimately, for each video sequence, we generate a dense point map tensor of shape $\mathbb{R}^{T\times L\times H\times W\times 3}$. To support structural and topological evaluations, we also export the corresponding dynamic anchor mask and instance segmentation map.

A.4.2 Evaluation Metrics

To evaluate the kinematic consistency, we introduce three metrics, including Mean Acceleration Magnitude (MAM), Edge Length Variance (ELV), and Variance of Local Pairwise Distances (VLPD). Let $\mathcal{X} = \{x_i(t) \in \mathbb{R}^3 \mid i = 1, \ldots, N, t \in \mathcal{T} \subset [0, T_{max}]\}$ denote the 3D trajectories of N points. Unlike traditional integer frame indices, the evaluation temporal set $\mathcal{T} \subset \mathbb{R}_{\geq 0}$ can consist of arbitrary fractional timestamps.

Mean Acceleration Magnitude (MAM). To evaluate the temporal smoothness of predicted trajectories, we utilize the Mean Acceleration Magnitude (MAM). The similar metrics has widely adopted in human dynamics and continuous motion forecasting [14, 61].We approximate the instantaneous acceleration via second-order finite differences over consecutive timestamps with a step size Δt. The MAM is defined as:

$$MAM = \frac{1}{N|\mathcal{T}^*|}\sum_{i=1}^{N}\sum_{t\in\mathcal{T}^*}\left\|\frac{x_i(t+\Delta t) - 2x_i(t) + x_i(t-\Delta t)}{\Delta t^2}\right\|_2 \tag{5}$$

where $\mathcal{T}^*$ denotes the set of valid timestamps where the second-order derivative can be computed. A lower MAM indicates higher temporal smoothness.

Edge Length Variance (ELV). To evaluate the spatial structural stability at arbitrary timestamps, we introduce Edge Length Variance (ELV). Given a pointmap, let $\mathcal{E}$ denote the set of pairs (i, j) representing spatially adjacent points within the 2D grid (i.e., immediate horizontal and vertical neighbors). To avoid wrongly penalizing points that naturally move apart at object boundaries (such a moving foreground object and a static background), we only include a pair in $\mathcal{E}$ if both points belong to the same instance mask. Because we have these masks for every integer and fractional timestep, we can apply such at any timestep. Let $l_{i,j}(t) = \|x_i(t) - x_j(t)\|_2$ be the 3D Euclidean distance between these adjacent points i and j at time t. The ELV is computed as the temporal variance of these local distances averaged over all valid adjacent pairs. Formally,

$$ELV = \frac{1}{|\mathcal{E}|}\sum_{(i,j)\in\mathcal{E}}\mathrm{Var}_{t\in\mathcal{T}}\left(l_{i,j}(t)\right). \tag{6}$$

Variance of Local Pairwise Distances (VLPD). For each point x_i, we define its local spatial neighborhood $\mathcal{N}_K(i)$ by identifying its K-nearest neighbors at the initial state $t = 0$. The distance between pair (i, j) at any time t is $d_{i,j}(t) = \|x_i(t) - x_j(t)\|_2$. VLPD measures the temporal variance of these local pairwise distances ($K = 15$):

$$VLPD = \frac{1}{NK} \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_K(i)} \mathrm{Var}_{t \in \mathcal{T}} \left(d_{i,j}(t) \right). \tag{7}$$

Average percent of Points within Delta (APD). Following St4RTrack and TAPVid-3D protocol, we also introduce APD between the predict point trajectories and the ground truth at both integer and fractional timestamps, which predict error and measure the percentage of points whose error falls below thresholds $\delta_{3D} \in \{0.1m, 0.3m, 0.5m, 1.0m\}$.

A.5 Network Architecture Details of LVP

The Latent Velocity Prediction (LVP) module is designed to predict continuous velocity field features. Initially, the FM-guided velocity features and geometric features, both with an input dimension of 768, are projected into a 256-dimensional latent space via respective two-layer MLPs equipped with GELU activations, transitioning from 768 to 512 and finally down to 256 channels. The core architecture comprises four Spatial-Temporal Cross-Attention blocks. To condition the network on continuous time, 128-dimensional time embeddings are injected into the features via AdaLN before each attention and MLP operation. Within each of these four blocks, both the spatial and temporal cross-attention layers are set to hidden dimension of 256 and utilize 8 heads. Following the Spatial-Temporal Cross-Attention, sequence is decoded through a two-layer MLP head to output the final velocity field features. The features are then sent to four convolutional layers to decode the final velocity values at integer and fractional timestamps.

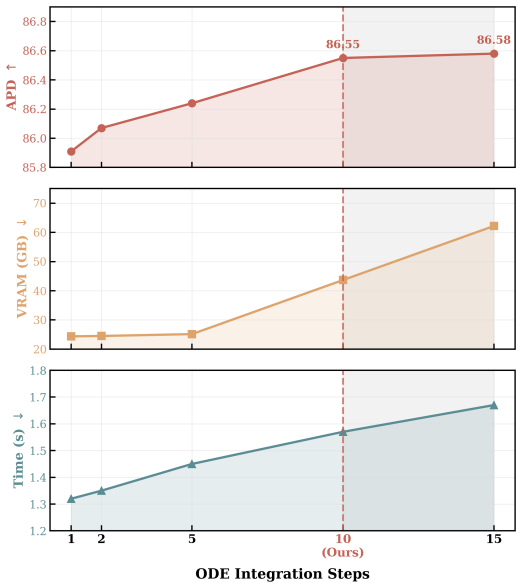

Figure 6: **Impact of ODE integration steps between adjacent integer frames on performance and computational efficiency**. The charts illustrate the **top:** Average percent of Points within Delta (APD). We report the APD on ADT dataset under the tracking task; **middle:** GPU VRAM consumption; **bottom :** inference time with 80-frame input. In our model, we employ 10 steps (ODE's integration step size of 0.1).

We emphasize that at $t = 0.5$, $X_{0.5}^{FM}$ is symmetric with respect to the two anchor features. However, the LVP predicts velocities by jointly conditioning on the FM-guided velocity features, geometric features from the point reconstruction branch, and queried-time embeddings. Reversing the frame order also changes the geometric features provided to the LVP and hence its complete conditioning inputs. Moreover, in Eq. (2), the conditional target vector field is $u_t(x|x_0, x_1) = x_1 - x_0$. Reversing the frame order changes the target vector field to $x_0 - x_1 = -(x_1 - x_0)$, thereby reversing the directional supervision for the continuous latent velocity field learned by the LVP. Thus, the same $X_{0.5}^{FM}$ corresponds to different LVP outputs and signs under different frame orders.

A.6 Clarification of the Numerical Solver for ODE

In our Uni4R, we apply the first-order Euler method as the numerical solver for ODE. The first-order Euler method is a standard, computationally efficient, and widely used numerical solver for ODE

initial-value problems, and has also been used in Rectified Flow to integrate predicted vector fields. The ODE specifies the continuous relationship between the point state and its time derivative, while the discrete sequence produced by the Euler method numerically approximates its solution.

More specifically, let $C(t)$ denote the conditioning information provided to the LVP, including the FM-guided velocity features, geometric features, and queried-time embedding. The LVP predicts the time-dependent 3D velocity $V_\theta(t \mid C(t))$, and the ODE governing the point trajectory is

$$\frac{dP(t)}{dt} = V_\theta(t \mid C(t)), \qquad P(t_0) = P_0. \tag{8}$$

This is a time-dependent but state-independent ODE: its right-hand side $V_\theta(t \mid C(t))$ depends on t through the time-dependent conditioning $C(t)$, but does not depend on the current point state $P(t)$. Within each adjacent integer-time interval, the anchor timestamp t_0 and the temporally ordered queried timestamps $t_1 < \cdots < t_Q$ define a sequence of time steps $\Delta t_i = t_{i+1} - t_i$. After the LVP predicts the time-dependent 3D velocities at the corresponding timestamps, Uni4R starts from the anchor point state P_0 and applies the first-order Euler method to numerically solve this ODE by sequentially accumulating the displacement contributed by each predicted velocity over its corresponding time step:

$$P_{i+1} = P_i + \Delta t_i V_\theta(t_i \mid C(t_i)). \tag{9}$$

At each step, the displacement $\Delta t_i V_\theta(t_i \mid C(t_i))$ is added to the point state obtained from the preceding step. For example,

$$P_1 = P_0 + \Delta t_0 V_\theta(t_0 \mid C(t_0)), \tag{10}$$

and

$$P_2 = P_1 + \Delta t_1 V_\theta(t_1 \mid C(t_1)) = P_0 + \Delta t_0 V_\theta(t_0 \mid C(t_0)) + \Delta t_1 V_\theta(t_1 \mid C(t_1)). \tag{11}$$

By sequentially applying this update over all time steps, Uni4R obtains the point state P_Q after Q Euler updates, which numerically approximates the continuous point position $P(t_Q)$:

$$P(t_Q) \approx P_Q = P_0 + \sum_{i=0}^{Q-1} \Delta t_i V_\theta(t_i \mid C(t_i)) \tag{12}$$

The velocities at different queried timestamps are predicted from their corresponding conditioning information, whereas each point state P_i is obtained by cumulatively applying the preceding Euler updates. Thus, although P_i is not an explicit input to the LVP, it is determined by the velocities accumulated over the preceding time steps. During training, gradients from the integral-consistency loss are backpropagated through these Euler updates to the predicted velocities, thereby guiding the LVP to learn a continuous 3D velocity field whose integrated endpoints agree with the integer-time ground-truth point maps.

Table 8: **Ablation Experiments** about LVP and ODE Contributions on 4D Reconstruction and Point Tracking (report APD_{3D} ↑), and continuous-time kinematic coherence (named KC, we report ELV ↓).

Methods	Point Tracking				Reconstruction		KC
	PO	DR	ADT	PStudio	PO	TUM	
Uni4R (Ours)	**81.59**	**76.05**	**86.55**	**79.43**	**80.71**	**84.36**	**10.2437**
Direct $X_1 - X_0 \to V$	77.91	74.71	82.48	76.65	78.64	82.99	10.2853
Direct velocity prediction w/o ODE	79.15	75.28	83.64	76.83	78.59	83.81	10.2802

A.7 More Experiments

A.7.1 More ablations about LVP and ODE Contributions

To further verify the LVP and ODE contributions, we conduct several variant experiments as shown in Tab. 8. First, we feed $X_1 - X_0 \to V$ directly to the velocity decoder. And the second experiment is that we direct apply velocity prediction without ODE. These variants are trained under the same setting as Uni4R.

Both variants consistently underperform the full Uni4R across all metrics. In particular, directly feeding $X_1 - X_0 \to V$ as the input to the velocity decoder does not improve performance over the original formulation, showing that $X_1 - X_0$ cannot serve as a direct substitute for the velocity field features F_{vel} produced within the LVP.

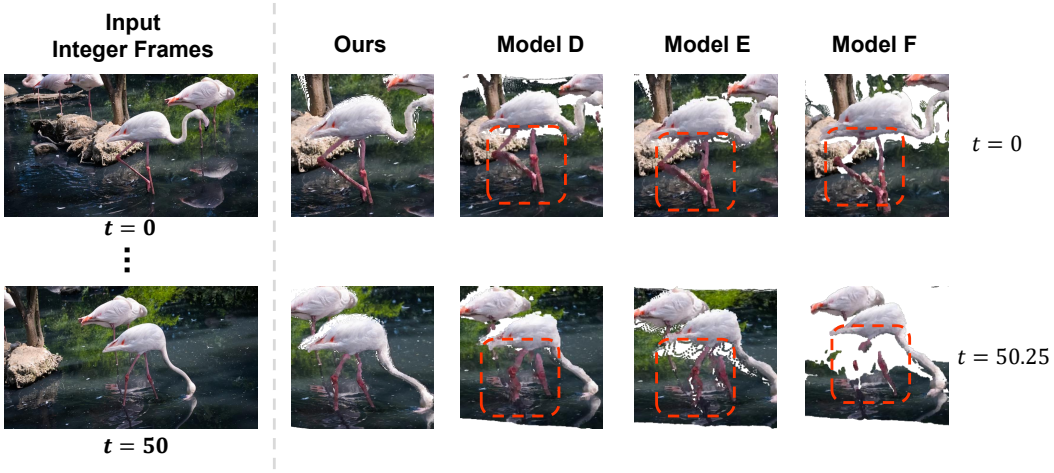

Figure 7: **Qualitative Ablation of Core Components.** We compare qualitative results by the model D, E, F in Tab. 5. As core components are progressively removed from model D to model F, the quality of results at fractional timestamps significantly degrades, leading to obvious structural tearing (such as the leg and background). These verify the effectiveness of our core components.

A.7.2 Qualitative Ablation of Core Components

Please see Fig. 7.

A.7.3 Impact of Integration Step Size

We conduct an ablation study on the number of ODE integration steps between two adjacent integer frames. As illustrated in Fig. 6, increasing the number of integration steps from 1 to 10 yields a steady improvement in tracking performance, with the APD rising from 85.90 to 86.55. This indicates that finer integration steps allow the model to capture continuous motion dynamics more accurately.

However, as the step count exceeds 10, the performance gain rapidly saturates, reaching 86.58 for 15 steps. In contrast, both inference time and VRAM consumption surge, with memory usage exceeding 60 GB at 15 steps. Therefore, to balance performance and computational efficiency, we set the number of ODE integration steps to 10 as the default configuration for our model.

A.7.4 Visualization of Long-sequence Reconstruction and Point Tracking

Although our model is trained on 16 frames, it demonstrates generalization capabilities when processing longer sequences during inference. As illustrated in Fig. 8, our method maintains high-quality outputs over longer sequences. For 4D reconstruction (top of Fig. 8), the model successfully preserves local geometric structures even at temporally distant frames (e.g., t = 35 and t = 45). For point tracking, trajectories remain smooth over a long sequence. Furthermore, the experiments in the main manuscript (such as Tab.1 and Tab.2) are mainly based on 64 frames. These results demonstrate the generalization capabilities of our model on long sequence reconstruction and point tracking.

A.7.5 Visualization of Collision

To demonstrate the robustness of our method against abrupt motion like collision, we visualize our point tracking and reconstruction results on challenging scenarios involving collisions and high-speed movements (see Fig. 9). In the scenario about multi-Object collision (Top Row), objects undergo sudden directional changes due to physical impact. Our method accurately captures the sharp inflection points of the point trajectories (visualized as color-coded streaks) during the collision. The local geometry structure remains preserved without topology collapse. For the fast-moving yellow object (Bottom Row) experiencing severe motion blur and a sudden stop against the white box, our method successfully predicts its abrupt trajectory transitions.

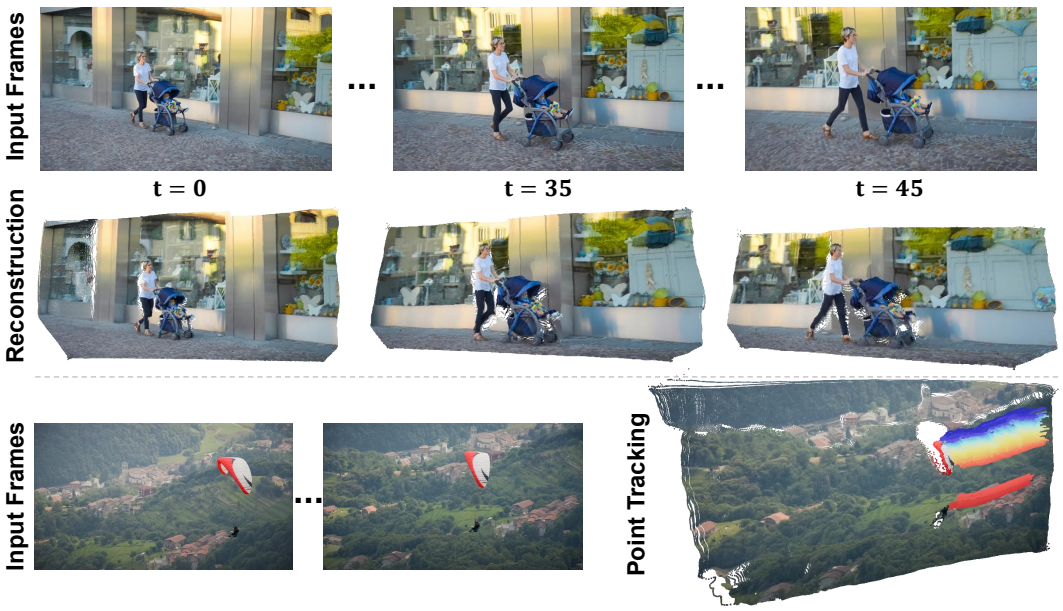

Figure 8: **Our Results when input long-sequence frames.** when we input long-sequence frames, our method still successfully predicts reconstruction pointmaps (top) and tracking (below, we present the tracking from time 5 to time 40) without topology collapse.

Table 9: Fundamental Differences between Uni4R and EulerFlow.

Aspect	**Uni4R (Ours)**	**EulerFlow** [42]
Task definition	Unified 4D reconstruction and point tracking at arbitrary timestamps directly from video.	Continuous-time scene flow estimation from point-cloud sequences, with emergent 3D point tracking through ODE integration.
Input	Video frames.	Point-cloud sequences.
Learning setting	Feed-forward model.	Per-sequence test-time optimization.
Velocity field modeling	Uses an OT-defined probability path on the anchor feature manifold and the LVP to predict continuous 3D velocities.	Directly optimizes a scene-specific Eulerian flow field.
Training or optimization objectives	Constrains the endpoints obtained through anchor-based ODE integration using integral-consistency training.	Optimizes the scene-specific flow field using reconstruction and cycle-consistency objectives.

A.7.6 Comparison with Optimization-based Continuous Motion Modeling Approach

Recent optimization-based approaches formulate continuous motion modeling by fitting scene-specific velocity fields for individual sequences, such as EulerFlow [42]. Uni4R instead learns a feed-forward continuous velocity field from videos, enabling unified 4D reconstruction and point tracking. Specifically, they differ in several aspects, as summarized in Tab. 9 .

We further compare our Uni4R with EulerFlow under the point tracking (PT) and kinematics-aware (KA) evaluation protocols, as shown in Tab. 10. Since EulerFlow requires point-cloud sequences as input, we first apply MegaSaM to convert each input video into a point-cloud sequence. Because EulerFlow does not directly reconstruct point maps from video, reconstruction metrics are not applicable; we therefore report its PT and KA results only. PT includes PO, DR, ADT, and PS on all and dynamic points, while KA includes MAM, APD, ELV, and VLPD. Uni4R consistently outperforms EulerFlow across all reported evaluations.

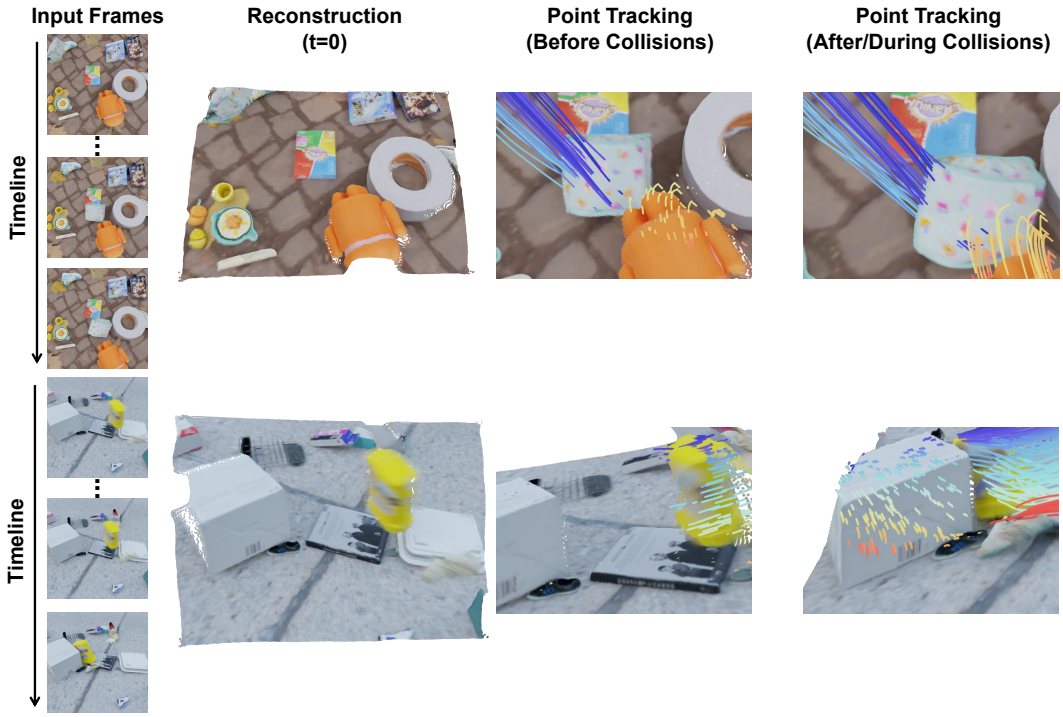

Figure 9: **Our Results about Collision Scenarios.** We visualize our point tracking and reconstruction results on challenging scenarios involving collisions and high-speed movements. Our method successfully predicts their abrupt trajectory transitions while preserves the local geometry structure.

Table 10: **Comparison of Uni4R and EulerFlow** on Point Tracking (report APD_{3D} ↑), and continuous-time kinematic coherence (named KC, we report ELV under fractional timestamps ↓).

Methods	Point Tracking (All Points)				Point Tracking (Dynamic Points)				KC
	PO	DR	ADT	PStudio	PO	DR	ADT	PStudio	
EulerFlow	31.66	57.46	67.67	68.05	34.23	57.99	74.74	68.04	10.3086
Uni4R (Ours)	**81.59**	**76.05**	**86.55**	**79.43**	**83.16**	**78.91**	**77.74**	**79.43**	**10.2437**

B Kinematic Grounding of the OT Probability Path

Conceptually, our framework constrains the learned features to reside on the *anchor feature manifold*. Mathematically, this manifold is exactly the *trajectory manifold* $\mathcal{M}$ formed by the support of all valid optimal transport (OT) paths. While traversing this manifold ensures optimal feature-space transport, we clarify that an arbitrary Euclidean straight line in a learned feature space is not automatically a kinematically meaningful trajectory in the 3D world coordinates. The desired connection holds only when the feature geometry is kinematically grounded. In other word, when the decoder maps local feature displacements within $\mathcal{M}$ to world-space physical displacements with controlled metric distortion. In this section, we formalize this condition and show how our unified training objective encourages such grounding.

Let $\mathcal{Z} \subset \mathbb{R}^d$ denote the anchor feature space and let $\mathcal{X} \subset \mathbb{R}^{3N}$ denote the world-coordinate pointmap space. Let

$$D_\phi : \mathcal{Z} \to \mathcal{X} \tag{13}$$

be the decoder that maps an anchor feature to a 3D/4D pointmap. For two matched anchor features $z_0, z_1 \in \mathcal{Z}$, the OT-CFM probability path is given by the displacement interpolation [21]:

$$z_t = (1-t)z_0 + tz_1, \qquad t \in [0,1]. \tag{14}$$

The corresponding conditional vector field is

$$u_t(z_t \mid z_0, z_1) = \frac{dz_t}{dt} = z_1 - z_0. \tag{15}$$

Minimum-energy property in feature space. For any absolutely continuous feature path $\gamma : [0,1] \to \mathcal{Z}$ satisfying $\gamma(0) = z_0$ and $\gamma(1) = z_1$, define its feature-space kinetic energy as

$$\mathcal{E}_{\mathcal{Z}}[\gamma] = \int_0^1 \|\dot{\gamma}(t)\|_2^2 \, dt. \tag{16}$$

By the Cauchy–Schwarz inequality,

$$\mathcal{E}_{\mathcal{Z}}[\gamma] = \int_0^1 \|\dot{\gamma}(t)\|_2^2 \, dt \geq \left\| \int_0^1 \dot{\gamma}(t) \, dt \right\|_2^2 = \|z_1 - z_0\|_2^2. \tag{17}$$

Equality is achieved if and only if $\dot{\gamma}(t)$ is constant almost everywhere, i.e., when

$$\gamma(t) = (1-t)z_0 + tz_1. \tag{18}$$

Therefore, the OT displacement interpolation is the unique constant-speed minimum-energy path in the Euclidean feature chart.

Under the Conditional Flow Matching objective,

$$\mathcal{L}_{\mathrm{CFM}} = \mathbb{E}_{t,z_0,z_1}\left[\|v_\theta(z_t, t) - (z_1 - z_0)\|_2^2 \right], \tag{19}$$

the population optimum satisfies

$$v_\theta^\star(z_t, t) = z_1 - z_0. \tag{20}$$

Thus, the learned latent velocity field recovers the minimum-energy feature-space transport direction induced by the OT coupling.

Why feature-space optimality does not automatically imply world-space optimality. The decoded world trajectory is

$$x_t = D_\phi(z_t). \tag{21}$$

Its world-space velocity is given by the chain rule:

$$\dot{x}_t = J_\phi(z_t)\dot{z}_t, \tag{22}$$

where

$$J_\phi(z) = \nabla_z D_\phi(z) \tag{23}$$

is the decoder Jacobian. Hence the world-space kinetic energy of a decoded feature path γ is

$$\mathcal{E}_{\mathcal{X}}[D_\phi \circ \gamma] = \int_0^1 \dot{\gamma}(t)^\top G_\phi(\gamma(t))\dot{\gamma}(t) \, dt, \tag{24}$$

where

$$G_\phi(z) = J_\phi(z)^\top J_\phi(z) \tag{25}$$

is the pullback metric induced by the decoder.

This expression shows that world-space energy is not generally minimized by a Euclidean straight line in $\mathcal{Z}$. It is minimized by geodesics under the pullback metric G_ϕ. Therefore, to make the OT feature path kinematically meaningful, the feature metric must be aligned with the world-space kinetic metric.

Kinematic metric alignment. We theoretically expect the decoder to be locally near-isometric on the support of anchor trajectories to ensure kinematic coherence. As the ambient feature space dimension is much larger than the 3D world-coordinate space, the decoder's Jacobian is inherently rank-deficient. Therefore, rather than imposing a global full-rank constraint, the near-isometry is defined strictly for valid local feature displacements δz lying on the tangent space of the trajectory manifold. For features z in the OT trajectory tube and their valid displacements δz, the spatial distance generated by the decoder is bounded by:

$$(1-\epsilon)\lambda\|\delta z\|_2^2 \leq \|D_\phi(z + \delta z) - D_\phi(z)\|_2^2 \leq (1+\epsilon)\lambda\|\delta z\|_2^2 \tag{26}$$

up to higher-order terms, where $\lambda > 0$ is a global scale factor.

In our framework, this property is implicitly encouraged through the robust pointmap regression objective adopted from DUSt3R [45]. By directly penalizing the 3D coordinate regression errors at anchor frames, the decoder D_ϕ is heavily constrained against degenerate feature reparameterizations. Furthermore, driven by the continuous inductive bias of the Flow Matching guided features, this unified optimization naturally regularizes the feature-space geometry. It encourages the decoder to map smooth latent interpolants into kinematically consistent 3D trajectories without requiring computationally expensive Jacobian supervision.

Approximate minimum-energy transfer to world coordinates. Let

$$\ell(t) = (1-t)z_0 + tz_1 \tag{27}$$

be the OT feature path lying on the trajectory manifold $\mathcal{M}$. Because its tangent vector $\dot{\ell}(t) = z_1 - z_0$ represents a valid displacement in $T_{\ell(t)}\mathcal{M}$, under the restricted metric alignment condition, its decoded world energy satisfies:

$$\mathcal{E}_{\mathcal{X}}[D_\phi \circ \ell] = \int_0^1 \dot{\ell}(t)^\top G_\phi(\ell(t))\dot{\ell}(t)\, dt \le (1+\epsilon)\lambda \|z_1 - z_0\|_2^2. \tag{28}$$

For any other absolutely continuous feature path γ connecting z_0 and z_1 within the same trajectory manifold $\mathcal{M}$ (ensuring $\dot{\gamma}(t) \in T_{\gamma(t)}\mathcal{M}$), we have:

$$\mathcal{E}_{\mathcal{X}}[D_\phi \circ \gamma] \ge (1-\epsilon)\lambda \int_0^1 \|\dot{\gamma}(t)\|_2^2\, dt. \tag{29}$$

Using the feature-space minimum-energy result for straight lines,

$$\int_0^1 \|\dot{\gamma}(t)\|_2^2\, dt \ge \|z_1 - z_0\|_2^2. \tag{30}$$

Therefore,

$$\inf_{\gamma:\gamma(0)=z_0,\gamma(1)=z_1,\gamma\subset\mathcal{M}} \mathcal{E}_{\mathcal{X}}[D_\phi \circ \gamma] \ge (1-\epsilon)\lambda \|z_1 - z_0\|_2^2. \tag{31}$$

Combining the upper bound for the OT path and the lower bound for arbitrary decoded paths on $\mathcal{M}$ gives:

$$\mathcal{E}_{\mathcal{X}}[D_\phi \circ \ell] \le \frac{1+\epsilon}{1-\epsilon} \inf_{\gamma:\gamma(0)=z_0,\gamma(1)=z_1,\gamma\subset\mathcal{M}} \mathcal{E}_{\mathcal{X}}[D_\phi \circ \gamma]. \tag{32}$$

Thus, the decoded OT path is a bounded-distortion approximation to the minimum-energy world-coordinate path. In the special case $\epsilon = 0$, i.e., when the decoder is locally isometric up to scale, the decoded OT path is the minimum-energy trajectory in world coordinates within the learned motion manifold $\mathcal{M}$.

Kinematic coherence through integral consistency. The above metric argument establishes energy consistency in the latent space. We further enforce temporal kinematic coherence by requiring the predicted continuous 3D velocity field $\mathbf{V}_\theta(C(t))$ to integrate consistently to the decoded pointmaps. Theoretically, the ideal world-space velocity should be the exact push-forward of the latent velocity v_θ through the pointmap decoder's Jacobian J_ϕ:

$$\mathbf{V}^*(t) = J_\phi(z_t) v_\theta(z_t, t), \tag{33}$$

where $P(t) = D_\phi(z_t)$ denotes the continuous spatial trajectory state of the tracked point. However, computing the exact Jacobian-vector product for a highly non-linear decoder (e.g., DPT) is computationally prohibitive. Therefore, as described in Sec. 3.2, we employ independent convolution layers to decode the latent representations into the continuous 3D velocity field $\mathbf{V}_\theta(C(t))$, where $C(t)$ denotes the conditioning information of the velocity predictor. These convolution layers act as an efficient neural approximation of the exact theoretical push-forward operator. Note that $P(t)$ is the state propagated by ODE integration and is not an input to the velocity predictor.

To guarantee that this empirical approximation faithfully satisfies the required mathematical equivalence, for any interval $[t_a, t_b] \subset [0,1]$, a kinematically coherent trajectory must satisfy:

$$D_\phi(z_{t_b}) = D_\phi(z_{t_a}) + \int_{t_a}^{t_b} \mathbf{V}_\theta(C(\tau))\, d\tau. \tag{34}$$

We therefore explicitly penalize deviations from this equivalence using the integral consistency loss:

$$\mathcal{L}_{\mathrm{ICT}} = \mathbb{E}_{t_a,t_b}\left[\left\| D_\phi(z_{t_b}) - \left(D_\phi(z_{t_a}) + \int_{t_a}^{t_b} \mathbf{V}_\theta(C(\tau))\, d\tau \right) \right\|_2^2\right]. \tag{35}$$

This loss does not require ground-truth velocities at fractional timestamps. Instead, it uses an ODE solver to integrate the predicted continuous velocity field and supervises the resulting trajectory through available pointmap observations at integer timestamps.

Together, the DUSt3R loss constrains the spatial endpoints, the CFM loss constrains the latent-space transport direction, and $\mathcal{L}_{\mathrm{ICT}}$ enforces that the neurally approximated velocity field integrates consistently through time.

Summary. The OT probability path provides an exact minimum-energy trajectory in the Euclidean anchor feature space. This property transfers to world coordinates only under a strict kinematic grounding condition on the decoder. Under this grounding condition, by approximately aligning the decoder's pullback metric with the Euclidean feature metric along the trajectory manifold $\mathcal{M}$ through robust pointmap regression and enforcing integral consistency of the decoded velocity field, feature-space straight-line OT paths decode into bounded-distortion, kinematically coherent world-coordinate trajectories. When the learned decoder is locally isometric up to scale ($\epsilon = 0$), the decoded OT trajectory is exactly minimum-energy within the learned motion manifold $\mathcal{M}$; otherwise, its energy distortion is bounded by the factor $\frac{1+\epsilon}{1-\epsilon}$.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: This paper claims the contributions that we propose Uni4R, a unified framework that shifts 4D reconstruction and point tracking from heuristic interpolations to continuous kinematic modeling, unlocking the capability to query dynamics at any arbitrary timestamp. Uni4R learns continuous velocity fields through the synergy of Optimal Transport (OT) and Ordinary Differential Equation (ODE).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss our limitations in Section. 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Our proof has been validated and verified through Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We will open-source our code.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Currently, the code needs to be cleaned. We will open-source the code after the manuscript published. The data we used is public dataset.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We have outlined the details of our training.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in Appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to the high cost of model training, it is difficult for us to run repeated experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We use internal clusters to train our models. In terms of estimation, the training process consumes 120 GPU hours for one model.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have strictly followed the NeurIPS Code of Ethics throughout this paper.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: 4D reconstruction and point tracking has a broad impact on society. On one hand, these tasks significantly improve the production efficiency of virtual reality and autonomous driving. On the other hand, we closely monitor the potential societal risks related to the misuse of results reconstructed by our models. We have ensured that our training and evaluation processes utilize publicly available datasets free of sensitive personal information.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Not applicable.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite related papers in Section. 4.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: In the A.4, we provide more details about the new benchmark.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We do not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: We do not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We do not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.