MoSE3: Learning World-Space SE(3) at Every Pixel

Jiahuan Cheng[1,3*] **Zhiyi Li**[4*] **Tian Xia**[1*] **Ruojin Cai**[1,2] **Yilun Du**[1,2] **Qianqian Wang**[1,2]
[1]Harvard University [2]Kempner Institute [3]Johns Hopkins University [4]MIT

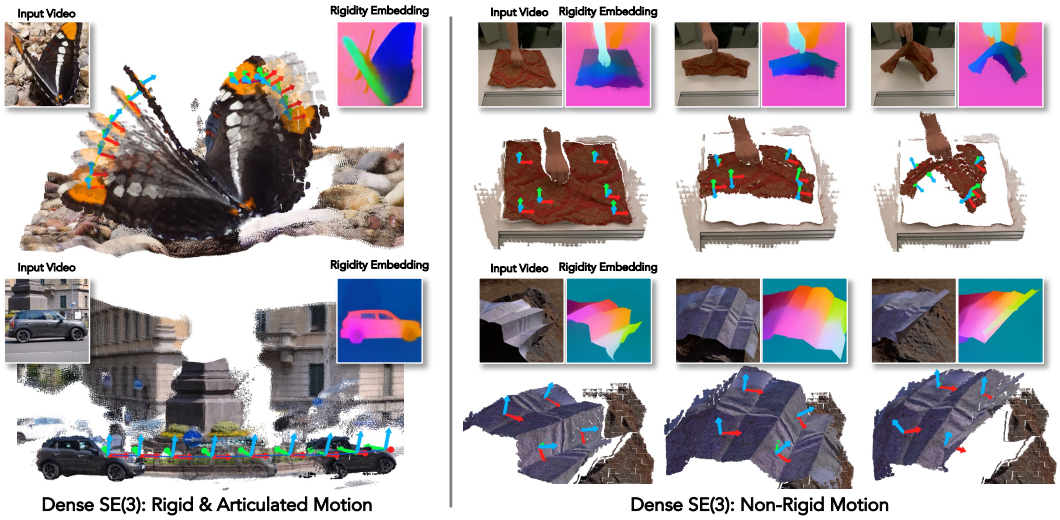

Figure 1: **MoSE3** estimates dense $\mathrm{SE}(3)$ motion from monocular video in a single forward pass. It predicts a 3D rotation and translation at every pixel in a shared world coordinate frame, capturing rigid and articulated motion (left) as well as deformable motion (right). Colored coordinate axes visualize the predicted local transformations at sampled points; insets show the input frames and learned rigidity embeddings.

Abstract

Dense 3D point tracking has been a prominent paradigm for modeling motion in dynamic scenes, but a point track is just a 3-DoF translation curve per pixel: it captures where pixels go, not the rotation of the underlying part, nor which pixels move together as one body. We propose MoSE3, the first feed-forward model that predicts dense $\mathrm{SE}(3)$ motion from monocular RGB video, producing full 6-DoF rigid transforms at every pixel in world space. Per-pixel $\mathrm{SE}(3)$ motion offers a richer view of how a scene moves: rotation, translation, and grouping all at once. Directly predicting $\mathrm{SE}(3)$ is challenging: rotations lie on a curved manifold that is ill-suited to Euclidean regression, and annotations for $\mathrm{SE}(3)$ are particularly difficult to acquire. To address these challenges, MoSE3 predicts per-pixel $\mathrm{SE}(3)$ through two jointly learned intermediates, 3D point tracks and rigidity embeddings, and recovers $\mathrm{SE}(3)$ by differentiably fitting transforms within each soft rigid cluster, enabling end-to-end prediction and supervision. To close the data gap, we introduce Art-Kubric, a large-scale synthetic dataset with dense $\mathrm{SE}(3)$ and rigidity labels for articulated objects with rich physical interactions. MoSE3 achieves state-of-the-art $\mathrm{SE}(3)$ estimation at pixel, part, and object levels on both rigid and articulated benchmarks, and state-of-the-art average 3D point tracking accuracy across three datasets, while showing strong generalization to real-world videos despite being trained solely on synthetic motion data. **Project page:** mose3-tracker.github.io

*Equal contribution.

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

1 Introduction

Understanding how a scene moves over time from video is a foundational problem in computer vision, underpinning applications from physical reasoning to robotic manipulation. The accuracy and richness of the motion understanding bound what downstream systems can do.

Motion estimation from monocular video has advanced rapidly, from short-range optical flow [28, 62, 81], to long-range 2D point tracking [12, 13, 24, 40], to 3D point tracking [67, 100, 110] powered by geometric foundation models [87, 92, 93], each step extending motion estimation across longer horizons and into stronger spatial grounding. Despite this progress, these representations remain low-level: each is a translation curve per pixel, with no rotation and no rigid grouping. However, many downstream tasks need a more structured and compact representation to act on. For example, a robot opening a hinged door wants its rotation axis and joint angle, not the trajectories of a million surface points. The signal exists in current trackers but is buried at the level of individual points.

In this paper, we propose to learn a more structured motion representation from monocular RGB videos: *SE(3) trajectories*, where each pixel carries a full 6-DoF rigid transform in a shared world coordinate frame. A single transform explains the motion of an entire moving part, and the pixels that share it define that part. Deformable objects in turn become smooth fields of locally rigid pieces. This is the representation that physical reasoning, manipulation, and articulated-object understanding need, but directly predicting world-space $\mathrm{SE}(3)$ for in-the-wild RGB video has largely remained out of reach, for several reasons. First, recovering geometry and camera motion for dynamic scenes has itself been an open problem until recently. Second, direct $\mathrm{SE}(3)$ regression generalizes poorly due to the difficulty of its prediction space [82, 100]. Third, long-range scene $\mathrm{SE}(3)$ annotation is exceptionally scarce: even 3D track ground truth is largely confined to synthetic data, and most such datasets lack $\mathrm{SE}(3)$ labels altogether. The prior approaches that recover $\mathrm{SE}(3)$ either operate at the granularity of single rigid objects with known CAD models or depth sensors [96, 97, 99], require parametric body models [60, 118], or rely on slow per-video optimization [49, 89].

To address these challenges, we propose MoSE3, a feed-forward model that predicts per-pixel $\mathrm{SE}(3)$ motion from monocular RGB video. The key insight is to decompose $\mathrm{SE}(3)$ prediction into two intermediates that are individually easier to learn and generalize: dense 3D point tracks, and *rigidity embeddings* that group co-moving pixels into rigid bodies. From these, $\mathrm{SE}(3)$ is recovered analytically by a differentiable weighted Horn [27] fit within each soft rigid cluster. The decomposition lets us leverage the more available 3D tracking supervision and train the embeddings via the differentiable $\mathrm{SE}(3)$ recovery, making the full prediction end-to-end trainable. To close the data gap, we introduce Art-Kubric, a synthetic dataset and generation pipeline for generating articulated scenes with dense $\mathrm{SE}(3)$ and rigidity labels across a wide range of categories with multi-body physical interactions.

At inference, by clustering the learned rigidity embeddings, MoSE3 recovers object- and part-level 6-DoF poses directly from monocular video. Experiments show MoSE3 achieves state-of-the-art $\mathrm{SE}(3)$ estimation at pixel, part, and object levels on both rigid and articulated benchmarks, and state-of-the-art average 3D point tracking accuracy across three datasets, with strong generalization to real-world videos.

In summary, our contributions are: **1)** MoSE3, the first feed-forward model for dense $\mathrm{SE}(3)$ motion from monocular RGB video, without category priors, depth sensors, or per-video optimization. **2)** A decomposition of $\mathrm{SE}(3)$ into 3D point tracks and rigidity embeddings, connected by a differentiable closed-form fit, enabling end-to-end $\mathrm{SE}(3)$ supervision from scarce annotations and strong generalization. **3)** Art-Kubric, a large-scale dataset and generation pipeline with dense $\mathrm{SE}(3)$ and rigidity labels for articulated scenes. **4)** State-of-the-art performance for feed-forward $\mathrm{SE}(3)$ estimation at pixel, part, and object levels, and for 3D point tracking, across multiple benchmarks.

2 Related Work

Optical Flow and 2D Point Tracking. Optical flow estimates a dense 2D displacement field between consecutive frames [28, 62]. Deep learning [15, 31, 32, 79, 81, 94, 102] has driven dramatic accuracy gains, but their predictions are inherently short-horizon. Long-range point tracking traces back to classical particle-video formulations [73, 74]. Modern methods [9, 12–14, 24, 25, 39, 40, 48, 52, 76, 88, 116] have substantially improved long-range 2D tracking accuracy through supervised

learning on synthetic data. However, these methods operate in the image plane: their output is a 2D translation trajectory per pixel, with no 3D structure or rigid grouping.

Scene Flow and 3D Point Tracking. Scene flow extends per-pixel motion from 2D to 3D, with formulations on point clouds [21, 57] or RGB-D inputs [75, 82, 85], and more recently on monocular RGB [54, 103]. Long-range 3D point tracking [110] generalizes this signal across long horizons, predicting 3D trajectories of arbitrary pixels through occlusions. Early methods [67, 100] lift 2D tracks into 3D using monocular depth [3, 29, 42, 107, 108], but do not disentangle camera from scene motion. More recently, geometric foundation models [34, 50, 55, 90, 92, 111] like VGGT [87] and π^3 [93] estimate cameras and dense geometry in a feed-forward pass. These geometric foundations enable feed-forward world-space 3D tracking, previously only possible through per-video optimization [49, 89, 109]. Building on pairwise reconstruction models [50, 92], methods like POMATO [113], St4RTrack [18], and DPM [77] enable world-space 3D tracking. More recent methods [58, 77, 101], including SpatialTrackerV2 [101], Any4D [41], Track4World [61], leverage feed-forward multi-view reconstruction models [87, 93] to directly predict long-range 3D tracks on continuous videos with improved accuracy and temporal consistency. However, their primary output remains point translation over time. In contrast, MoSE3 predicts per-pixel $\mathrm{SE}(3)$ motion, capturing rotation, translation, and grouping as a more structured motion representation.

$\mathrm{SE}(3)$ **Motion Estimation.** The most common setting for recovering $\mathrm{SE}(3)$ motion is 6-DoF object pose estimation [4, 47, 80, 86, 96, 97, 99], which regresses a single rigid transform per object instance. These methods inherently assume a strictly rigid object, and often require a CAD model [47, 97] or a depth sensor [86, 97]. For deformable objects, many approaches rely on category-specific priors, from low-rank shape bases [5, 68] to parametric models like SMPL [60] and SMAL [118]. Some 4D object [105, 106] and scene [49, 56, 89] reconstruction methods can recover local $\mathrm{SE}(3)$ transforms without category priors, but rely on expensive per-video optimization rather than feed-forward prediction. On scene-level $\mathrm{SE}(3)$ prediction, piecewise-rigid scene flow [83], RAFT-3D [82], and RigidMask [104] estimate per-pixel or per-region rigid motion between frame pairs, while multi-body structure-from-motion [10, 46] and dynamic visual odometry [30] jointly recover grouping and motion over sparse feature tracks through geometric optimization. Like RAFT-3D, we use learned rigidity embeddings to softly group pixels with shared rigid motion, which yields a finer, part-level decomposition than object-level motion segmentation [43, 51]. Whereas RAFT-3D iteratively refines pairwise $\mathrm{SE}(3)$ transforms through reprojection-based optimization, we recover world-frame $\mathrm{SE}(3)$ motion through closed-form alignment of predicted 3D point tracks. Earlier learned approaches such as SE3-Nets [7] and SE3-Pose-Nets [8] jointly segment scene parts and predict their per-part $\mathrm{SE}(3)$ motion, but operate on point clouds in robotic settings rather than monocular in-the-wild videos. SpatialTracker [100] learns rigidity embeddings for feed-forward 3D point tracking, but does not supervise or output per-pixel $\mathrm{SE}(3)$ motion, and its predicted trajectories entangle camera and scene motion. In summary, MoSE3 is the first to predict per-pixel, world-frame $\mathrm{SE}(3)$ motion feed-forward from monocular RGB video, naturally modeling articulated and deformable objects as collections of local rigidities, without requiring CAD models, depth or stereo input, category templates, or per-video optimization.

3 Method

We introduce MoSE3, a feed-forward network that predicts per-pixel $\mathrm{SE}(3)$ motion from a monocular video. Built on the π^3 [93] backbone, MoSE3 introduces two additional predictions: dense point tracks and rigidity embeddings. From these predictions, per-pixel $\mathrm{SE}(3)$ can be recovered in closed form. Sec. 3.1 presents our $\mathrm{SE}(3)$ prediction formulation, Sec. 3.2 introduces the training objective, and Sec. 3.3 describes the model architecture. Fig. 2 provides an overview.

3.1 $\mathrm{SE}(3)$ Prediction Formulation

Given N input frames $\{I_i\}_{i=1}^N$ at resolution $H \times W$ and a designated query frame index $q \in \{1, \dots, N\}$ to track from, our goal is to predict a sequence of dense world-frame rigid transformations $\{\mathbf{T}_{q\to i}\}_{i=1}^N$, where each $\mathbf{T}_{q\to i} = (\mathbf{R}_{q\to i}, \mathbf{t}_{q\to i}) \in \mathrm{SE}(3)^{H\times W}$. Let $\mathbf{X}_q^w \in \mathbb{R}^{H\times W\times 3}$ denote the world-frame 3D location of the scene point at each pixel of I_q. Then $\mathbf{T}_{q\to i}$ maps $\mathbf{X}_q^w$ to its location at time i, $\mathbf{X}_i^w = \mathbf{R}_{q\to i}\mathbf{X}_q^w + \mathbf{t}_{q\to i}$.

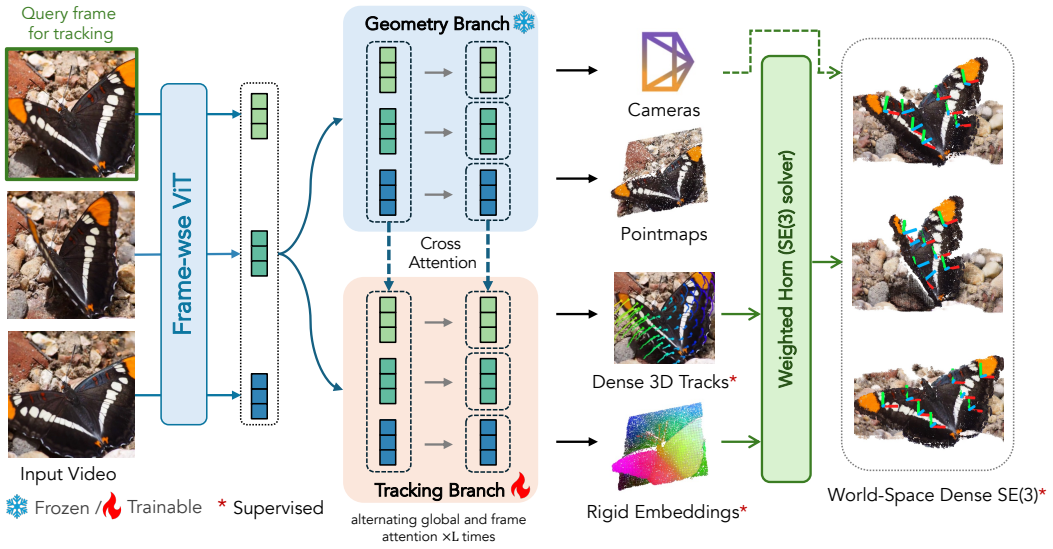

Figure 2: **Method overview.** MoSE3 extends the frozen π^3 backbone with a trainable tracking branch that predicts dense 3D point tracks $\mathbf{X}_i^c$ and rigidity embeddings $\mathbf{E}_i$. Per-pixel $\mathrm{SE}(3)$ motion is then recovered in closed form via a weighted Horn fit on the world-frame tracks, with weights given by rigidity-embedding similarity. Because the Horn fit is differentiable, the dense $\mathrm{SE}(3)$ prediction is supervised end-to-end together with the tracks and rigidity embeddings.

A natural approach is to directly regress per-pixel rotation and translation. Although continuous rotation representations support neural regression, learning accurate dense $\mathrm{SE}(3)$ motion remains challenging, particularly when supervision is scarce. We therefore investigate a decomposition that exploits more widely available point-tracking supervision and recovers valid rigid transforms analytically. Our ablation in Sec. 5.3 compares this formulation with direct $\mathrm{SE}(3)$ regression.

Instead, we make two key observations. First, recent 3D point trackers trained solely on synthetic data generalize well to real-world videos [18, 58]. Second, given dense 3D point tracks and a rigid grouping of pixels, $\mathrm{SE}(3)$ motion is recoverable in closed form via Procrustes alignment [1, 27]. Our key insight is thus that predicting $\mathrm{SE}(3)$ can be reduced to predicting two intermediate quantities: 3D point tracks $\mathbf{X}_i^w$, and an L2-normalized *rigidity embedding* $\mathbf{E}_q \in \mathbb{R}^{H\times W\times D}$ trained so that pixels on the same rigid body produce similar embeddings. Both targets are easier to learn and generalize than direct $\mathrm{SE}(3)$ regression (we defer their learning objectives to Sec. 3.2). From these, the $\mathrm{SE}(3)$ motion of a given pixel $\mathbf{u}$, $\mathbf{T}_{q\to i}(\mathbf{u})$, can be recovered in closed form via Horn's method [27]:

$$\mathbf{T}_{q\to i}(\mathbf{u}) \;=\; \arg\min_{\hat{\mathbf{T}}\in \mathrm{SE}(3)} \sum_{\mathbf{v}\in\mathcal{G}} w(\mathbf{u},\mathbf{v}) \left\| \hat{\mathbf{T}}\cdot \mathbf{X}_q^w(\mathbf{v}) - \mathbf{X}_i^w(\mathbf{v}) \right\|_2^2, \tag{1}$$

where the soft weights $w(\mathbf{u},\mathbf{v})$ are a temperature-τ softmax over rigidity-embedding inner products,

$$w(\mathbf{u},\mathbf{v}) = \frac{\exp\big(\mathbf{E}_q(\mathbf{u})^\top \mathbf{E}_q(\mathbf{v})/\tau\big)}{\sum_{\mathbf{v}'\in\mathcal{G}} \exp\big(\mathbf{E}_q(\mathbf{u})^\top \mathbf{E}_q(\mathbf{v}')/\tau\big)}, \tag{2}$$

$\mathbf{v},\mathbf{v}'\in\mathcal{G}$ are sampled reference pixels on the query frame. Intuitively, $w(\mathbf{u},\mathbf{v})$ measures how rigidly $\mathbf{v}$ moves with $\mathbf{u}$, so the alignment in Eq. 1 softly fits $\mathbf{T}_{q\to i}(\mathbf{u})$ to the trajectory of $\mathbf{u}$'s rigid neighbors. The optimization via weighted Horn is closed-form and differentiable, so $\mathrm{SE}(3)$ supervision flows back into both the point-track and rigidity-embedding predictions.

3.2 Joint Supervision of Tracks, Rigidity Embeddings, and $\mathrm{SE}(3)$

Given a query frame q, our model predicts a 3D point-track map $\mathbf{X}_i^c$ in each target frame i's local camera coordinates, visibility logits V_i, and rigidity embeddings $\mathbf{E}_i$. Tracks and visibility are indexed by query pixels, whereas $\mathbf{E}_i$ is defined on frame i's image grid. π^3 additionally predicts pointmaps $\mathbf{M}_i$ and camera-to-world poses $\mathbf{P}_i$, giving $\mathbf{X}_i^w = \mathbf{P}_i\cdot\mathbf{X}_i^c$. We jointly supervise tracking, visibility, rigidity affinities, and the recovered $\mathrm{SE}(3)$ motion.

3.2.1 Tracking Losses

We internally parameterize each 3D track position by normalized image coordinates $\mathbf{p}_i = (u_i, v_i)$ and signed depth z_i. These are lifted to $\mathbf{X}_i^c$ using focal lengths fitted from π^3's predicted pointmaps; training uses ground-truth focal lengths for the 3D losses. We supervise normalized image coordinates with an ℓ_1 loss $\mathcal{L}_{uv}$ and scene-scale-aligned depth with an inverse-depth-weighted ℓ_1 loss $\mathcal{L}_z$. We additionally apply a query-frame surface-gradient loss $\mathcal{L}_\nabla$ and a temporal-displacement loss $\mathcal{L}_\Delta$, weighted by λ_∇ and λ_Δ, giving four geometric tracking terms:

$$\mathcal{L}_{\text{track}} = \mathcal{L}_{uv} + \mathcal{L}_z + \lambda_\nabla \mathcal{L}_\nabla + \lambda_\Delta \mathcal{L}_\Delta. \tag{3}$$

The parameterization, lifting, and scale alignment are detailed in Appendix A.3, and the loss definitions in Appendix A.4. Separately, a binary cross-entropy loss $\mathcal{L}_{\text{vis}}$ supervises visibility logits V_i against the ground-truth visibility flags on valid annotated tracks.

3.2.2 Rigidity Embedding Loss

We supervise rigidity embeddings through two soft-affinity objectives, using $\mathrm{SE}(3)$ annotations when available and 3D track geometry on all eligible clips. Both objectives can apply to the same clip and match a predicted similarity distribution to a row-normalized target affinity. We draw observations across all input frames. An observation a at pixel $\mathbf{u}$ of frame i has embedding $\widetilde{\mathbf{E}}_a = \mathbf{E}_i(\mathbf{u}) \in \mathbb{R}^D$. Let $\mathcal{J}_a$ hold the other observations b with valid target affinity A_{ab}. With temperature τ_e, define

$$\pi_{ab} = \frac{\exp(\widetilde{\mathbf{E}}_a^\top \widetilde{\mathbf{E}}_b / \tau_e)}{\sum_{b' \in \mathcal{J}_a} \exp(\widetilde{\mathbf{E}}_a^\top \widetilde{\mathbf{E}}_{b'} / \tau_e)}, \qquad \pi^\star_{ab} = \frac{A_{ab}}{\sum_{b' \in \mathcal{J}_a} A_{ab'}}. \tag{4}$$

Let $\mathcal{O}_A$ contain the observations whose target rows have positive mass. We minimize the mean row-wise divergence

$$\mathcal{K}(A) = \frac{1}{|\mathcal{O}_A|} \sum_{a \in \mathcal{O}_A} \sum_{b \in \mathcal{J}_a} \pi^\star_{ab} \log \frac{\pi^\star_{ab}}{\pi_{ab}}, \tag{5}$$

excluding self-pairs and taking $\mathcal{K}(A) = 0$ when $\mathcal{O}_A$ is empty. Both objectives have this form, differing only in the target affinity A. We define its two choices below, and $\mathcal{L}_{\text{emb}}$ combines the resulting losses.

$\mathrm{SE}(3)$ **affinity loss.** This term applies only where $\mathrm{SE}(3)$ labels are available. Let $g(a)$ index the motion annotation associated with observation a, which can correspond to an object, articulated part, or mesh face. For query-relative ground-truth world transforms $(\mathbf{R}^\star_{i,m}, \mathbf{t}^\star_{i,m})$, define

$$\begin{aligned} D^2_{mn} &= \max_i \left[\frac{\mathrm{ang}(\mathbf{R}^{\star\top}_{i,m} \mathbf{R}^\star_{i,n})^2}{\sigma_R^2} + \frac{\|\mathbf{t}^\star_{i,m} - \mathbf{t}^\star_{i,n}\|_2^2}{\sigma_t^2} \right], \\ A^{\mathrm{SE}(3)}_{ab} &= \exp\left(-D^2_{g(a)g(b)} / \tau_{\text{aff}}\right), \qquad \mathcal{L}^{\mathrm{SE}(3)}_{\text{aff}} = \mathcal{K}(A^{\mathrm{SE}(3)}). \end{aligned} \tag{6}$$

Here σ_R and σ_t are angular and translational bandwidths, τ_{aff} sets how fast the affinity decays, and $\mathrm{ang}(\cdot)$ is the geodesic angle. Rotations are measured in radians and translations in the original metric world coordinates, before scene-scale normalization. Only annotation trajectories valid across the sampled clip are used. Unlike hard rigid-class targets, these affinities vary continuously with the difference between motion trajectories.

Tracking affinity loss. This term requires no $\mathrm{SE}(3)$ or rigid-body labels. For ground-truth tracks a, b, let $\overline{\mathbf{X}}^\star_i$ denote scene-normalized ground-truth positions and $d_i(a,b) = \|\overline{\mathbf{X}}^\star_i(a) - \overline{\mathbf{X}}^\star_i(b)\|_2$. Over their jointly valid frames $\mathcal{V}_{ab}$, define

$$\Delta_{ab} = \max_{i \in \mathcal{V}_{ab}} d_i(a,b) - \min_{i \in \mathcal{V}_{ab}} d_i(a,b), \qquad A^{\text{trk}}_{ab} = \exp(-\Delta^2_{ab} / \sigma^2_{\text{trk}}). \tag{7}$$

Here Δ_{ab} is the spread of the pairwise distance over time, and σ_{trk} its tolerance. Pairs require at least two jointly valid frames. To supervise the per-frame embedding maps, we draw visible track–frame observations. We assign each observation pair the affinity of its underlying tracks and set $\mathcal{L}^{\text{trk}}_{\text{aff}} = \mathcal{K}(A^{\text{trk}})$. This supplies a distance-preservation cue across frames on tracking data with visibility and camera annotations, including clips without $\mathrm{SE}(3)$ labels. Visibility gates the embedding reads, whereas track validity gates the geometric target.

3.2.3 $\mathrm{SE}(3)$ Losses

When $\mathrm{SE}(3)$ ground truth $\mathbf{T}^{\star}_{q\to i}(\mathbf{u}) = (\mathbf{R}^{\star}_{q\to i}, \mathbf{t}^{\star}_{q\to i})$ is available (on Art-Kubric, and post-processed Kubric and Syn4D), we additionally supervise the recovered $\mathbf{T}_{q\to i}(\mathbf{u}) = (\mathbf{R}_{q\to i}, \mathbf{t}_{q\to i})$ directly:

$$\mathcal{L}_{\mathrm{SE}(3)} = \sum_{\mathbf{u}} \Big[\operatorname{ang}\left(\mathbf{R}_{q\to i}\, \mathbf{R}^{\star\top}_{q\to i}\right) + \alpha\, \rho\big(\big\|\mathbf{T}_{q\to i}(\mathbf{u})\cdot\mathbf{c}(\mathbf{u}) - \mathbf{T}^{\star}_{q\to i}(\mathbf{u})\cdot\mathbf{c}(\mathbf{u})\big\|\big)\Big], \tag{8}$$

where $\rho(\cdot)$ is a Huber penalty, α balances the two terms, and $\mathbf{c}(\mathbf{u})$ is the weighted query-frame centroid of the anchor's Procrustes support. The second term compares where the predicted and ground-truth transforms place $\mathbf{c}(\mathbf{u})$.

3.2.4 Full Loss

The full objective combines the tracking-branch, rigidity-embedding, and $\mathrm{SE}(3)$ losses described above. On clips that lack $\mathrm{SE}(3)$ or rigidity-embedding labels, the corresponding terms are zeroed. The full multi-task objective is the weighted sum,

$$\mathcal{L} = \lambda_{\text{track}}\,\mathcal{L}_{\text{track}} + \lambda_{\text{vis}}\,\mathcal{L}_{\text{vis}} + \lambda_{\text{emb}}\,\mathcal{L}_{\text{emb}} + \lambda_{\mathrm{SE}(3)}\,\mathcal{L}_{\mathrm{SE}(3)}, \tag{9}$$

where all loss weights are fixed. Training runs in a single stage with all losses applied jointly, proceeding in three consecutive phases that differ only in their learning-rate schedule. Full training details, including loss weights and optimization, are in Appendix A.

3.3 Geometry-Grounded Tracking Architecture

Having specified what MoSE3 predicts and how it is trained, we now describe the model architecture. We attach a trainable tracking branch alongside π^3's frozen geometry branch. We keep the geometry branch frozen to preserve its pretrained geometric priors while learning motion from tracking data. The tracking tokens attend to π^3's geometry tokens at every tracking-branch layer after the shared frozen layers. This gives the tracking branch access to a concrete geometric signal throughout the network: predicted tracks in frame i should align with π^3's pointmap in frame i for co-visible regions.

Tracking Decoder. Concretely, we reuse π^3's image encoder and first few decoder layers, then fork into the frozen geometry branch and a trainable tracking branch whose blocks attend with $\mathrm{Q} =$ tracking tokens and $\mathrm{K}, \mathrm{V} = [\text{tracking tokens}, \text{geometry tokens}]$. Both task heads below consume the tracking branch's output features.

Point-Tracking Head. The point-tracking head produces dense $\mathbf{X}^{c}_{i}$ and per-pixel visibility logits V_i over all N frames. Its blocks alternate frame-wise and global self-attention and are conditioned on query-frame identity through Adaptive Layer Norm [70], applying one set of normalization parameters to I_q's tokens and another to the rest.

Rigidity-Embedding Head. The rigidity-embedding head outputs $\mathbf{E}_i$ from the same tracking-branch features. Unlike the point-tracking head, it is query-frame *invariant* (no AdaLN for the query/non-query split), since rigid-body identity is a property of the scene rather than the query. Output embeddings are L2-normalized so that the inner products in Eq. 2 measure cosine similarity.

Additional architecture details, including the layer-wise frozen/trainable split in the tracking branch and the point-tracking head design, are provided in Appendix B.

4 The Art-Kubric Dataset

We introduce **Articulated Kubric (Art-Kubric)**, a synthetic dataset and generation pipeline built around six capabilities that, to the best of our knowledge, no existing public corpus combines: (1) *articulated assets* from 46 PartNet-Mobility [66, 98] categories, 226 procedurally authored Articraft [117] categories, and 7 Infinigen-Articulated [37] categories; (2) *dense long-range 2D tracks and metric world-space 3D tracks*; (3) *link-level* $\mathrm{SE}(3)$ *pose ground truth* for every kinematic link at every frame; (4) *a per-pixel partition of the image by shared rigid motion*; (5) *multi-body object interaction* under physics simulation rather than isolated-asset playback, with rigid and articulated bodies sharing every scene; and (6) a *configurable rendering pipeline* that randomizes HDRI lighting, camera intrinsics, and object composition per scene.

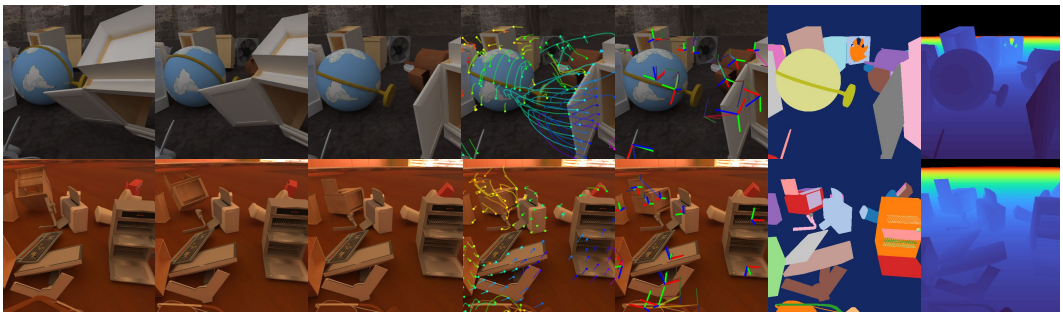

Figure 3: **Art-Kubric overview.** Two example scenes (one per row). *Columns 1–3:* three RGB frames sampled from the clip. *Columns 4–7 (annotations rendered at the third frame):* sampled long-range 3D tracks colored by their parent rigid body, per-pixel $\mathrm{SE}(3)$ motion as rotated coordinate frames at sparse anchors, rigid-part segmentation (per-pixel partition by shared rigid transform), and depth. Articulated objects from PartNet-Mobility [66, 98], Articraft [117], and Infinigen-Articulated [37] interact with rigid Google Scanned Objects [16] under physics simulation while a moving camera observes the scene under varying HDRI lighting.

The pipeline lets the community re-render unlimited new clips with the same supervision channels. The release contains $5{,}000$ scenes of 13–30 rigid and articulated objects, about two-thirds of them in motion, dropped into a textured environment and observed by a moving camera, exported with depth, surface normals, three-granularity segmentation, and full camera parameters. Across the release this comes to 163M tracks, each carrying its rigid-group label, its per-frame occlusion flag, and its link's $\mathrm{SE}(3)$ transform at every frame. Resolution and file layout follow Kubric MOVi-F [20] so that existing tracking loaders ingest the data without modification. Tab. 7 in the appendix contrasts the supervision channels with prior corpora.

Existing synthetic tracking corpora are either rigid-only, like Kubric [12, 20], or dominated by skinned humans and animals with no piecewise-rigid parts, like PointOdyssey [115] and Dynamic Replica [38]. Rigid-only scenes can provide per-object $\mathrm{SE}(3)$ labels by fitting transforms from tracks and instance masks, but this breaks under articulation: links of one object share a single instance label, so the fitted transform collapses sub-object motion. The rigidity supervision degenerates for the same reason: when every body is a single rigid piece, the partition by shared transform coincides with instance segmentation, and an embedding trained only on such scenes is never asked to split a body into parts. Articulated datasets [17, 59, 114] either cover few object templates or stage one articulated object at a time and drive its joints by sampling angles within their limits, leaving no second body that the embedding has to keep apart. Art-Kubric is the first to combine articulated multi-link motion, physics-driven inter-body contact, and exact per-link $\mathrm{SE}(3)$ ground truth at scale, including for links that are occluded or too small in the image to fit a transform to. Its partition departs from instance segmentation in both directions in every scene: the links of one cabinet fall into separate groups, while every body at rest merges with the floor into one. It provides the articulated $\mathrm{SE}(3)$ supervision that our framework requires, and the ablation in Sec. 5.3 confirms its contribution to model performance. Full pipeline parameters and the on-disk schema are provided in Appendix D.

5 Experiments

We evaluate MoSE3 on $\mathrm{SE}(3)$ estimation (Sec. 5.1) and 3D point tracking (Sec. 5.2), and ablate key design choices in Sec. 5.3.

5.1 $\mathrm{SE}(3)$ Evaluation

We evaluate on HO3D [22, 23], which contains real-world hand-object interactions with rigid objects, and iTACO [71], which contains synthetic videos of articulated objects with moving parts. We additionally report results on YCBInEOAT [95], which contains real-world videos of a robot arm manipulating rigid objects, in Tab. 6. We evaluate at both per-pixel and object/part-level $\mathrm{SE}(3)$ granularities.

Table 1: $\mathrm{SE}(3)$ **estimation on HO3D and iTACO.** ADD is reported in centimeters, and IoU is measured against ground-truth rigid masks. AUC_r uses a 30° threshold, while $\mathrm{AUC}_{\mathrm{ADD}}$ uses 10 cm on HO3D and 20 cm on iTACO. Best in **bold**, second underlined.

(a) Per-pixel	HO3D				iTACO			
Method	RRE↓	ADD↓	AUC_r↑	$\mathrm{AUC}_{\mathrm{ADD}}$↑	RRE↓	ADD↓	AUC_r↑	$\mathrm{AUC}_{\mathrm{ADD}}$↑
SpatialTrackerV2 [101]	36.45	3.73	0.376	0.633	36.71	28.45	0.408	0.309
POMATO [113]	60.53	6.59	0.216	0.405	46.42	46.66	0.319	0.121
ZeroMSF [54]	61.53	6.70	0.224	0.409	46.52	43.68	0.298	0.128
Any4D [41]	57.02	5.24	0.263	0.504	43.98	31.50	0.364	0.292
V-DPM [78]	40.37	3.65	0.342	0.641	32.70	22.60	0.460	0.378
TraceAnything [58]	61.56	5.69	0.245	0.467	44.73	35.47	0.347	0.249
4RC [63]	38.62	3.46	0.367	0.667	28.60	20.91	0.471	0.410
Track4World-Pi3x [61]	33.21	3.37	0.371	0.669	28.07	24.82	0.447	0.312
Track4World-DA3 [61]	34.61	3.34	0.361	0.671	32.09	25.69	0.408	0.321
Ours (k-NN)	24.09	2.61	0.509	0.744	20.78	14.47	0.570	0.501
Ours (rigid clust.)	**18.28**	**2.16**	**0.569**	**0.785**	**12.30**	**11.37**	**0.741**	**0.594**
(b) Object/part-level	HO3D				iTACO			
Method	RRE↓	ADD↓	$\mathrm{AUC}_{\mathrm{ADD}}$↑	IoU↑	RRE↓	ADD↓	$\mathrm{AUC}_{\mathrm{ADD}}$↑	IoU↑
SpatialTrackerV2 [101]	32.56	3.46	0.670	0.385	27.25	23.01	0.421	0.239
POMATO [113]	59.99	6.18	0.446	0.197	41.53	44.48	0.152	0.190
ZeroMSF [54]	62.29	6.43	0.442	0.230	43.47	41.83	0.158	0.187
Any4D [41]	55.44	5.16	0.523	0.298	37.23	28.56	0.399	0.185
V-DPM [78]	38.34	3.56	0.660	0.421	28.25	19.92	0.448	0.258
TraceAnything [58]	61.20	5.65	0.488	0.221	38.45	32.47	0.341	0.142
4RC [63]	36.08	3.34	0.688	0.409	18.70	17.26	0.526	0.349
Track4World-Pi3x [61]	30.46	3.17	0.698	0.414	18.80	20.14	0.397	0.353
Track4World-DA3 [61]	30.98	3.20	0.694	0.383	23.78	22.07	0.399	0.319
ProxyPose† [112]	31.22	–	0.490	–	28.51	–	0.300	–
Ours (k-NN)	20.04	2.26	0.783	0.504	**9.87**	**10.27**	**0.631**	0.505
Ours (rigid clust.)	**17.96**	**2.06**	**0.802**	**0.709**	10.76	10.49	0.625	**0.784**

†We evaluate ProxyPose at the object/part level with its single-query protocol and ground-truth intrinsics. It provides no masks for IoU. We omit its ADD because a few tracking failures dominate the mean (Appendix C.2).

To the best of our knowledge, no prior method directly predicts dense world-space $\mathrm{SE}(3)$ motions from monocular videos. We therefore adapt existing 3D tracking methods into baselines using a shared post-hoc protocol. Specifically, for each query pixel, we find its 3D K-nearest neighbors at the query frame, treat neighboring pixels as belonging to the same rigid group, and recover a rigid transform from the predicted 3D tracks within that neighborhood via Horn [27]. Since performance depends strongly on the neighborhood size, we sweep K separately for each method on each clip and report the best result. In contrast, MoSE3 predicts a rigidity embedding, allowing us to utilize the learned rigid similarities rather than relying on the hard k-NN grouping. We additionally compare against ProxyPose [112], a 6-DoF pose tracker that needs no such adaptation but predicts a pose per query rather than a dense per-pixel field. Because its video-generation backbone makes many queries expensive, we evaluate its single-query variant and report it only at the object/part level.

For *per-pixel* $\mathrm{SE}(3)$, we report relative rotation error (RRE, °) and ADD [26] (cm). For *object/part-level* $\mathrm{SE}(3)$, we cluster pixels into rigid bodies with HDBSCAN [65]. We match each predicted cluster to a ground-truth part mask by IoU and re-fit a single $\mathrm{SE}(3)$ per frame for this part. Methods differ in the clustering feature: MoSE3 clusters directly on the rigidity embedding, while baselines do not predict rigid grouping explicitly. For baselines, we instead form a feature by stacking each pixel's recovered $\mathrm{SE}(3)$ across time and cluster based on that feature. We report the matched cluster's mask IoU together with its RRE, ADD, and $\mathrm{AUC}_{\mathrm{ADD}}$ across the video. See Appendix C.2 for additional evaluation details.

Results are shown in Tab. 1. MoSE3 achieves the best per-pixel and object/part-level $\mathrm{SE}(3)$ estimation on both rigid (HO3D) and articulated (iTACO) benchmarks across all metrics. At the per-pixel level, MoSE3 consistently reduces rotation error and ADD relative to prior methods, with larger gains

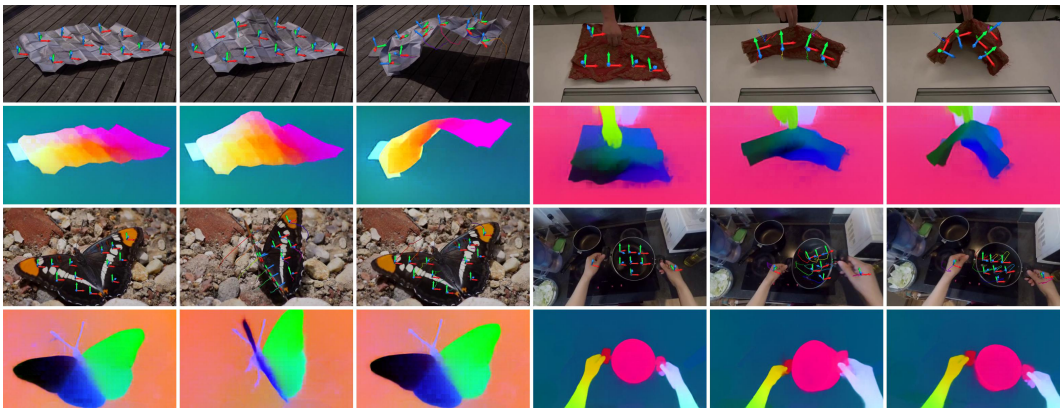

Figure 4: **Qualitative results on in-the-wild videos.** MoSE3 generalizes to out-of-distribution real-world scenes spanning both rigid and non-rigid cases. For each clip, the three columns are three time steps: the top row shows the predicted per-pixel $\mathrm{SE}(3)$ motion and the bottom row the predicted rigidity embedding.

Table 2: World-coordinate 3D point tracking on PointOdyssey, ADT, and PStudio. We follow the evaluation protocol of Track4World [61]. We report tracking accuracy at horizons L-16 (↑) and L-50 (↑). Best in **bold**, second underlined. Avg. is the unweighted mean across the three datasets.

	PointOdyssey [115]		ADT [69]		PStudio [36]		Avg.	
Method	L-16	L-50	L-16	L-50	L-16	L-50	L-16	L-50
SpatialTrackerV2 [101]	0.1925	0.1763	0.2456	0.2163	0.3790	0.3689	0.2724	0.2538
POMATO [113]	0.4425	0.3905	0.3611	0.3548	0.5166	0.4713	0.4401	0.4055
ZeroMSF [54]	0.4053	0.3505	0.4530	0.3563	0.4828	0.4386	0.4471	0.3818
Any4D [41]	0.4769	0.4174	0.4460	0.3717	0.5707	0.5066	0.4979	0.4319
V-DPM [78]	0.4848	0.4233	0.4783	0.3759	0.6084	0.5795	0.5238	0.4596
Track4World [61]	0.5345	0.5162	**0.6250**	0.5622	0.5946	0.5422	0.5847	0.5402
Ours	**0.5635**	**0.5650**	0.5789	**0.5953**	**0.6400**	**0.6235**	**0.5941**	**0.5946**

on the more challenging iTACO dataset. The improvement carries over to the part and object level, where MoSE3 achieves higher cluster IoU together with lower RRE and ADD. Even when using the same k-NN pipeline as the baselines, MoSE3 still outperforms every baseline on every metric, indicating that joint $\mathrm{SE}(3)$ supervision improves the underlying $\mathrm{SE}(3)$ predictions. The rigidity embedding further improves per-pixel $\mathrm{SE}(3)$ and yields substantially higher cluster IoU by grouping rigidly co-moving pixels, providing a more reliable neighborhood for $\mathrm{SE}(3)$ recovery. See Fig. 4 for qualitative results on in-the-wild videos.

5.2 3D Tracking Evaluation

Following the protocol of Track4World [61], we evaluate on PointOdyssey [115], ADT [69], and PStudio [36] from the TAPVid-3D benchmark [45], reporting tracking accuracy at horizons of 16 and 50 frames (L-16, L-50) in world coordinates. We omit DriveTrack [2] as its evaluation split has not been publicly released.

Tab. 2 shows that MoSE3 achieves the best average accuracy at both horizons, and the best result on every dataset and horizon except ADT L-16, where it ranks second behind Track4World. We provide the camera-coordinate tracking comparisons with additional baselines in Appendix C.3.

5.3 Ablation Study

We ablate four sets of design choices against the full MoSE3 model (Tab. 3). Due to compute constraints, we run all ablations (including our method) at small scale with identical settings other than the element being ablated. Training setup is provided in Appendix C.1.

(a) Architecture. *Direct* $\mathrm{SE}(3)$ *regr.* directly regresses $\mathrm{SE}(3)$, instead of through our track and rigidity-embedding decomposition. This degrades $\mathrm{SE}(3)$ accuracy and generalizes poorly, as direct regression must learn a high-dimensional, manifold-constrained output from scarce $\mathrm{SE}(3)$ labels. Our decomposition reduces the problem to two easier targets with broader supervision coverage, recovering $\mathrm{SE}(3)$ in closed form.

(b) Training data. Without Art-Kubric, training on existing datasets degrades $\mathrm{SE}(3)$ accuracy on iTACO and lowers only cluster IoU on HO3D (Tab. 4). Kubric's dense $\mathrm{SE}(3)$ and rigid-body labels cover single rigid objects like those in HO3D. Art-Kubric adds supervision for articulated multi-object scenes like those in iTACO.

Table 3: **Ablation study** on iTACO (per-pixel $\mathrm{SE}(3)$). Trained with a reduced schedule, detailed in Appendix C.1. ADD in cm; AUC_r at 30° and $\mathrm{AUC}_{\mathrm{ADD}}$ at 20 cm. Full results in Tab. 4.

Variant	RRE↓	AUC_r ↑	ADD↓	$\mathrm{AUC}_{\mathrm{ADD}}$ ↑
(a) Architecture				
Direct $\mathrm{SE}(3)$ regr.	24.57	0.525	19.05	0.393
(b) Training data				
w/o Art-Kubric	22.89	0.566	17.01	0.462
(c) Training strategy				
Track-only + k-NN	31.11	0.453	20.96	0.382
Track-only + rigid clust.	24.30	0.550	18.01	0.460
(d) Loss design				
w/o rigidity loss	<u>18.97</u>	<u>0.614</u>	<u>15.83</u>	<u>0.480</u>
w/o $\mathrm{SE}(3)$ loss	23.90	0.555	17.82	0.465
Ours (full)	**17.27**	**0.647**	**14.09**	**0.517**

(c) Training strategy. Joint training improves $\mathrm{SE}(3)$ accuracy through better tracks, not only better rigid grouping. We train a track-only variant that omits the rigidity-embedding head along with all $\mathrm{SE}(3)$ and rigidity-embedding supervision, then recover $\mathrm{SE}(3)$ post-hoc in two ways: by k-NN clustering of its tracks (*Track-only + k-NN*), or by grouping its tracks with the rigidity embedding of *Ours (full)* (*Track-only + rigid clust.*). Both underperform *Ours (full)*. The second variant is a controlled comparison: it borrows the full model's embedding, so the grouping weights are identical and only the tracks entering the Procrustes fit differ. The remaining gap is therefore attributable to the tracks alone, showing that supervising tracks jointly with the rigidity and $\mathrm{SE}(3)$ losses makes them more rigid-coherent and in turn yields more accurate $\mathrm{SE}(3)$.

(d) Loss design. Starting from the full model, we remove the rigidity loss and the $\mathrm{SE}(3)$ loss one at a time. Both degrade $\mathrm{SE}(3)$ accuracy, with the $\mathrm{SE}(3)$ loss having the larger impact. This confirms that direct end-to-end supervision is the strongest driver of $\mathrm{SE}(3)$ learning. Removing the rigidity loss instead degrades the rigidity embedding more heavily, lowering cluster IoU (Tab. 4): the $\mathrm{SE}(3)$ loss shapes the embedding enough to weight a local Procrustes fit, but not enough to separate rigid bodies globally.

6 Conclusion

We presented MoSE3, a feed-forward model that predicts per-pixel $\mathrm{SE}(3)$ motion from monocular RGB video by decomposing $\mathrm{SE}(3)$ into two easier-to-supervise intermediates, dense 3D point tracks and per-pixel rigidity embeddings, with $\mathrm{SE}(3)$ recovered analytically by a differentiable closed-form Horn fit. We also release *Art-Kubric*, a large-scale synthetic dataset that closes the supervision gap for multi-object articulated scenes. MoSE3 achieves state-of-the-art per-pixel and object-level $\mathrm{SE}(3)$ estimation in rigid and articulated scenes, and also sets a new state of the art for 3D point tracking. By turning monocular RGB video into a structured per-pixel motion representation that exposes rotation and rigid grouping alongside translation, MoSE3 provides a foundation for downstream tasks such as physical reasoning, manipulation, and articulated-object understanding.

Limitations. MoSE3 depends on π^3 for camera geometry, using its camera poses to compose world-frame motion and its pointmaps to fit camera intrinsics, so errors in either propagate into the recovered $\mathrm{SE}(3)$. Moreover, because $\mathrm{SE}(3)$ is recovered from point tracks and rigidity embeddings rather than regressed directly, accuracy degrades when these intermediates degrade, most notably under fast motion. Finally, we evaluate deformable scenes only qualitatively and leave a quantitative benchmark to future work.

References

[1] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on pattern analysis and machine intelligence*, (5):698–700,

1987.

[2] Arjun Balasingam, Joseph Chandler, Chenning Li, Zhoutong Zhang, and Hari Balakrishnan. Drivetrack: A benchmark for long-range point tracking in real-world videos. In *CVPR*, 2024.

[3] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

[4] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *European conference on computer vision*, pages 536–551. Springer, 2014.

[5] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3d shape from image streams. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pages 690–696, 2000.

[6] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Xindong He, Xu Huang, et al. AgiBot World Colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025.

[7] Arunkumar Byravan and Dieter Fox. Se3-nets: Learning rigid body motion using deep neural networks. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 173–180. IEEE, 2017.

[8] Arunkumar Byravan, Felix Leeb, Franziska Meier, and Dieter Fox. Se3-pose-nets: Structured deep dynamics models for visuomotor planning and control. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3339–3346, 2018.

[9] Seokju Cho, Jiahui Huang, Jisu Nam, Honggyu An, Seungryong Kim, and Joon-Young Lee. Local all-pair correspondence for point tracking. In *European Conference on Computer Vision*, pages 306–325. Springer, 2024.

[10] João Paulo Costeira and Takeo Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[11] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for EPIC-KITCHENS-100. *International Journal of Computer Vision*, 130:33–55, 2022.

[12] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. *Advances in Neural Information Processing Systems*, 35:13610–13626, 2022.

[13] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10061–10072, 2023.

[14] Carl Doersch, Pauline Luc, Yi Yang, Dilara Gokay, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ignacio Rocco, Ross Goroshin, Joao Carreira, et al. Bootstap: Bootstrapped training for tracking-any-point. In *Proceedings of the Asian Conference on Computer Vision*, pages 3257–3274, 2024.

[15] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015.

[16] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google Scanned Objects: A high-quality dataset of 3D scanned household items. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2553–2560, 2022.

[17] Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J Black, and Otmar Hilliges. ARCTIC: A dataset for dexterous bimanual hand-object manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12943–12954, 2023.

[18] Haiwen Feng, Junyi Zhang, Qianqian Wang, Yufei Ye, Pengcheng Yu, Michael J Black, Trevor Darrell, and Angjoo Kanazawa. St4rtrack: Simultaneous 4d reconstruction and tracking in the world. *arXiv preprint arXiv:2504.13152*, 2025.

[19] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. In *Advances in Neural Information Processing Systems*, 2022.

[20] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3749–3761, 2022.

[21] Xiuye Gu, Yijie Wang, Chongruo Wu, Yong Jae Lee, and Panqu Wang. Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3254–3263, 2019.

[22] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*, 2020.

[23] Shreyas Hampali, Sayan Deb Sarkar, Mahdi Rad, and Vincent Lepetit. Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation. In *CVPR*, 2022.

[24] Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *European Conference on Computer Vision*, pages 59–75. Springer, 2022.

[25] Adam W Harley, Yang You, Xinglong Sun, Yang Zheng, Nikhil Raghuraman, Yunqi Gu, Sheldon Liang, Wen-Hsuan Chu, Achal Dave, Suya You, et al. Alltracker: Efficient dense point tracking at high resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5253–5262, 2025.

[26] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Asian Conference on Computer Vision (ACCV)*, pages 548–562. Springer, 2012.

[27] Berthold K P Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.

[28] Berthold K P Horn and Brian G Schunck. Determining optical flow. *Artificial Intelligence*, 17 (1–3):185–203, 1981.

[29] Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan. Depthcrafter: Generating consistent long depth sequences for open-world videos. *arXiv preprint arXiv:2409.02095*, 2024.

[30] Jiahui Huang, Sheng Yang, Tai-Jiang Mu, and Shi-Min Hu. ClusterVO: Clustering moving instances and estimating visual odometry for self and surroundings. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 2165–2174, 2020.

[31] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer: A transformer architecture for optical flow. In *European Conference on Computer Vision*, pages 668–685. Springer, 2022.

[32] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2462–2470, 2017.

[33] Hanxiao Jiang, Hao-Yu Hsu, Kaifeng Zhang, Hsin-Ni Yu, Shenlong Wang, and Yunzhu Li. PhysTwin: Physics-informed reconstruction and simulation of deformable objects from videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7219–7230, October 2025.

[34] Zeren Jiang, Chuanxia Zheng, Iro Laina, Diane Larlus, and Andrea Vedaldi. Geo4d: Leveraging video generators for geometric 4d scene reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20658–20671, 2025.

[35] Zeren Jiang, Yushi Lan, Yihang Luo, Yufan Deng, Zihang Lai, Edgar Sucar, Christian Rupprecht, Iro Laina, Diane Larlus, Chuanxia Zheng, and Andrea Vedaldi. Syn4d: A multiview synthetic 4d dataset, 2026. URL https://arxiv.org/abs/2605.05207.

[36] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE international conference on computer vision*, pages 3334–3342, 2015.

[37] Abhishek Joshi, Beining Han, Jack Nugent, Max Gonzalez Saez-Diez, Yiming Zuo, Jonathan Liu, Hongyu Wen, Stamatis Alexandropoulos, Karhan Kayan, Anna Calveri, Tao Sun, Gaowen Liu, Yi Shao, Alexander Raistrick, and Jia Deng. Procedural generation of articulated simulation-ready assets, 2025. URL https://arxiv.org/abs/2505.10755.

[38] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. DynamicStereo: Consistent dynamic depth from stereo videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13229–13239, 2023.

[39] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024.

[40] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European Conference on Computer Vision*, pages 18–35. Springer, 2024.

[41] Jay Karhade, Nikhil Keetha, Yuchen Zhang, Tanisha Gupta, Akash Sharma, Sebastian Scherer, and Deva Ramanan. Any4d: Unified feed-forward metric 4d reconstruction. *arXiv preprint arXiv:2512.10935*, 2025.

[42] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9492–9502, 2024.

[43] M. Keuper, B. Andres, and T. Brox. Motion trajectory segmentation via minimum cost multicuts. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. URL http://lmb.informatik.uni-freiburg.de/Publications/2015/KB15b.

[44] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems (RSS)*, 2024.

[45] Skanda Koppula, Ignacio Rocco, Yi Yang, Joe Heyward, Joao Carreira, Andrew Zisserman, Gabriel Brostow, and Carl Doersch. Tapvid-3d: A benchmark for tracking any point in 3d. *Advances in Neural Information Processing Systems*, 37:82149–82165, 2024.

[46] Suryansh Kumar, Yuchao Dai, and Hongdong Li. Multi-body non-rigid structure-from-motion. In *Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016*, pages 148–156, December 2016.

[47] Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. Megapose: 6d pose estimation of novel objects via render & compare. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

[48] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. Dense optical tracking: Connecting the dots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[49] Jiahui Lei, Yijia Weng, Adam Harley, Leonidas Guibas, and Kostas Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. *arXiv preprint arXiv:2405.17421*, 2024.

[50] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024.

[51] Chun-Guang Li and Rene Vidal. Structured sparse subspace clustering: A unified optimization framework. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[52] Hongyang Li, Hao Zhang, Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, and Lei Zhang. Taptr: Tracking any point with transformers as detection. In *European Conference on Computer Vision*, pages 57–75. Springer, 2024.

[53] Xinyu Lian, Zichao Yu, Ruiming Liang, Yitong Wang, Li Ray Luo, Kaixu Chen, Yuanzhen Zhou, Qihong Tang, Xudong Xu, Zhaoyang Lyu, et al. Infinite mobility: Scalable high-fidelity synthesis of articulated objects via procedural generation. *arXiv preprint arXiv:2503.13424*, 2025.

[54] Yiqing Liang, Abhishek Badki, Hang Su, James Tompkin, and Orazio Gallo. Zero-shot monocular scene flow estimation in the wild. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21031–21044, 2025.

[55] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

[56] Junru Lin, Chirag Vashist, Mikaela Angelina Uy, Colton Stearns, Xuan Luo, Leonidas Guibas, and Ke Li. Global motion corresponder for 3d point-based scene interpolation under large motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7884–7893, October 2025.

[57] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 529–537, 2019.

[58] Xinhang Liu, Yuxi Xiao, Donny Y Chen, Jiashi Feng, Yu-Wing Tai, Chi-Keung Tang, and Bingyi Kang. Trace anything: Representing any video in 4d via trajectory fields. *arXiv preprint arXiv:2510.13802*, 2025.

[59] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. HOI4D: A 4D egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21013–21022, 2022.

[60] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, 2015.

[61] Jiahao Lu, Jiayi Xu, Wenbo Hu, Ruijie Zhu, Chengfeng Zhao, Sai-Kit Yeung, Ying Shan, and Yuan Liu. Track4world: Feedforward world-centric dense 3d tracking of all pixels. *arXiv preprint arXiv:2603.02573*, 2026.

[62] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, 1981.

[63] Yihang Luo, Shangchen Zhou, Yushi Lan, Xingang Pan, and Chen Change Loy. 4rc: 4d reconstruction via conditional querying anytime and anywhere. *arXiv preprint arXiv:2602.10094*, 2026.

[64] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016.

[65] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205, 2017.

[66] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 909–918, 2019.

[67] Tuan Duc Ngo, Peiye Zhuang, Chuang Gan, Evangelos Kalogerakis, Sergey Tulyakov, Hsin-Ying Lee, and Chaoyang Wang. Delta: Dense efficient long-range 3d tracking for any video. *arXiv preprint arXiv:2410.24211*, 2024.

[68] David Novotny, Nikhila Ravi, Benjamin Graham, Natalia Neverova, and Andrea Vedaldi. C3dpo: Canonical 3d pose networks for non-rigid structure from motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7688–7697, 2019.

[69] Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng Carl Ren. 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, 2023.

[70] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[71] Weikun Peng, Jun Lv, Cewu Lu, and Manolis Savva. iTACO: Interactable Digital Twins of Articulated Objects from Casually Captured RGBD Videos. In *3DV 2026*, 2025.

[72] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[73] Michael Rubinstein, Ce Liu, and William T Freeman. Towards longer long-range motion trajectories. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 53.1–53.11, 2012.

[74] Peter Sand and Seth Teller. Particle video: Long-range motion estimation using point trajectories. *International Journal of Computer Vision*, 80(1):72–91, 2008.

[75] Jakob Schmid, Azin Jahedi, Noah Berenguel Senn, and Andrés Bruhn. Ms-raft-3d: A multi-scale architecture for recurrent image-based scene flow. In *2025 IEEE International Conference on Image Processing (ICIP)*, pages 1570–1575. IEEE, 2025.

[76] Yunzhou Song, Jiahui Lei, Ziyun Wang, Lingjie Liu, and Kostas Daniilidis. Track everything everywhere fast and robustly. In *European Conference on Computer Vision*, pages 343–359. Springer, 2024.

[77] Edgar Sucar, Zihang Lai, Eldar Insafutdinov, and Andrea Vedaldi. Dynamic point maps: A versatile representation for dynamic 3d reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

[78] Edgar Sucar, Eldar Insafutdinov, Zihang Lai, and Andrea Vedaldi. V-dpm: 4d video reconstruction with dynamic point maps. *arXiv preprint arXiv:2601.09499*, 2026.

[79] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018.

[80] Jiaming Sun, Zihao Wang, Siyu Zhang, Xingyi He, Hongcheng Zhao, Guofeng Zhang, and Xiaowei Zhou. Onepose: One-shot object pose estimation without cad models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6825–6834, 2022.

[81] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.

[82] Zachary Teed and Jia Deng. Raft-3d: Scene flow using rigid-motion embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8375–8384, 2021.

[83] Christoph Vogel, Konrad Schindler, and Stefan Roth. Piecewise rigid scene flow. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[84] Homer Rich Walke, Kevin Black, Tony Z. Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, Abraham Lee, Kuan Fang, Chelsea Finn, and Sergey Levine. BridgeData V2: A dataset for robot learning at scale. In Jie Tan, Marc Toussaint, and Kourosh Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 1723–1736. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/walke23a.html.

[85] Bo Wang, Jian Li, Yang Yu, Li Liu, Zhenping Sun, and Dewen Hu. Scenetracker: Long-term scene flow estimation network. *arXiv preprint arXiv:2403.19924*, 2024.

[86] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3343–3352, 2019.

[87] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. *arXiv preprint arXiv:2503.11651*, 2025.

[88] Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. Tracking everything everywhere all at once. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19795–19806, 2023.

[89] Qianqian Wang, Vickie Ye, Hang Gao, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. *arXiv preprint arXiv:2407.13764*, 2024.

[90] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. *arXiv preprint arXiv:2501.12387*, 2025.

[91] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5261–5271, 2025.

[92] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.

[93] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025.

[94] Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *European Conference on Computer Vision*, 2024.

[95] Bowen Wen, Chaitanya Mitash, Baozhang Ren, and Kostas E Bekris. se (3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10367–10373. IEEE, 2020.

[96] Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Müller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[97] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17868–17879, 2024.

[98] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X Chang, Leonidas J Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11097–11107, 2020.

[99] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *Robotics: Science and Systems (RSS)*, 2018.

[100] Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20406–20417, 2024.

[101] Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Yuri Makarov, Bingyi Kang, Xing Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. Spatialtrackerv2: Advancing 3d point tracking with explicit camera motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6726–6737, 2025.

[102] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8121–8130, 2022.

[103] Gengshan Yang and Deva Ramanan. Upgrading optical flow to 3d scene flow through optical expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1334–1343, 2020.

[104] Gengshan Yang and Deva Ramanan. Learning to segment rigid motions from two frames. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1266–1275, 2021.

[105] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Huiwen Chang, Deva Ramanan, William T Freeman, and Ce Liu. Lasr: Learning articulated shape reconstruction from a monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15980–15989, 2021.

[106] Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. Banmo: Building animatable 3d neural models from many casual videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2863–2873, 2022.

[107] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024.

[108] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024.

[109] David Yifan Yao, Albert J Zhai, and Shenlong Wang. Uni4d: Unifying visual foundation models for 4d modeling from a single video. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1116–1126, 2025.

[110] Bowei Zhang, Lei Ke, Adam W Harley, and Katerina Fragkiadaki. Tapip3d: Tracking any point in persistent 3d geometry. *arXiv preprint arXiv:2504.14717*, 2025.

[111] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arXiv:2410.03825*, 2024.

[112] Ruihang Zhang, Felix Taubner, Pooja Ravi, Kiriakos N. Kutulakos, and David B. Lindell. Proxypose: 6-dof pose tracking via video-to-video translation. *arXiv preprint arXiv:2607.06555*, 2026.

[113] Songyan Zhang, Yongtao Ge, Jinyuan Tian, Guangkai Xu, Hao Chen, Chen Lv, and Chunhua Shen. Pomato: Marrying pointmap matching with temporal motion for dynamic 3d reconstruction. *arXiv preprint arXiv:2504.05692*, 2025.

[114] Weiguang Zhao, Haoran Xu, Xingyu Miao, Qin Zhao, Rui Zhang, Kaizhu Huang, Ning Gao, Peizhou Cao, Mingze Sun, Mulin Yu, Tao Lu, Linning Xu, Junting Dong, and Jiangmiao Pang. SynthVerse: A large-scale diverse synthetic dataset for point tracking. *arXiv preprint arXiv:2602.04441*, 2026.

[115] Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. 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, 2023.

[116] Artem Zholus, Carl Doersch, Yi Yang, Skanda Koppula, Viorica Patraucean, Xu Owen He, Ignacio Rocco, Mehdi SM Sajjadi, Sarath Chandar, and Ross Goroshin. Tapnext: Tracking any point (tap) as next token prediction. *arXiv preprint arXiv:2504.05579*, 2025.

[117] Matt Zhou, Ruining Li, Xiaoyang Lyu, Zhaomou Song, Zhening Huang, Chuanxia Zheng, Christian Rupprecht, Andrea Vedaldi, and Shangzhe Wu. Articraft: An agentic system for scalable articulated 3d asset generation, 2026. URL https://arxiv.org/abs/2605.15187.

[118] Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Appendix

A Implementation Details

A.1 Training Setup

We train MoSE3 by jointly supervising tracks, visibility, rigidity embeddings, and per-pixel $\mathrm{SE}(3)$ transforms. Training runs for 200 epochs in a single stage, proceeding in three consecutive phases that differ only in their learning-rate schedule, each resuming from the previous phase's checkpoint.

Initialization. We initialize MoSE3 from the pretrained π^3 [93] large checkpoint, keeping the π^3 encoder and geometry-branch decoder frozen throughout training. The trainable tracking branch is initialized from the corresponding frozen π^3 weights, with the within-block trainable/frozen split detailed in Sec. B.1. The transformer layers of both the point-tracking head and the rigidity-embedding head are deep-copied from π^3's pointmap head transformer layers. The additional global-wise self-attention layers in the point-tracking head are initialized from the corresponding frame-wise self-attention weights. The linear point-track and visibility outputs are deep-copied from π^3's pointmap and confidence linear heads. Because the point-track head predicts pixel offsets and a depth channel rather than a pointmap, we re-initialize that depth channel so that it predicts unit depth at every pixel at the start of training. Supervision is scale-normalized, so one unit is the mean scene scale of the reconstruction rather than one meter. Only the 16-dimensional rigidity-embedding linear output is trained from scratch.

Optimization. We optimize with AdamW at peak learning rate 8×10^{-5}, weight decay 0.05, and betas $(0.9, 0.999)$, excluding biases and one-dimensional parameters (e.g., LayerNorm) from weight decay. For the learning rate schedule, it first warms up over the first epoch (800 steps) to 8×10^{-5}. Then, we use a constant learning rate until the validation performance converges. We then gradually decay the learning rate, and use the cosine schedule to 4×10^{-8} in the last 40 epochs. In total, the model is trained in 200 epochs. We use per-GPU batch 1 with no gradient accumulation, yielding an effective batch of 64 sequences per optimizer step on $64\times$ NVIDIA H100-80GB GPUs (800 steps per epoch, 160,000 in total). Gradients are clipped at norm 1.0 and the total loss value is clipped at 100 to suppress occasional spikes. Mixed precision is applied selectively: all transformer blocks (encoder, decoder layers, and the head transformer layers) run in bf16, while the linear output heads, the post-processing math, and the entire loss module remain in FP32 for numerical stability.

Objective. We train with $\lambda_{\text{track}} = 1.0$ on both the pixel and depth components of the track loss, $\lambda_{\text{vis}} = 0.1$, $\lambda_{\text{emb}} = 0.1$, and $\lambda_{\mathrm{SE}(3)} = 1.0$. Within the track loss, the surface-gradient and temporal-displacement terms use $\lambda_{\nabla} = 1.0$ and $\lambda_{\Delta} = 2.0$. The rigidity term combines the two affinity losses as $\mathcal{L}_{\text{emb}} = \mathcal{L}_{\text{aff}}^{\mathrm{SE}(3)} + \frac{1}{2}\mathcal{L}_{\text{aff}}^{\text{trk}}$, giving the $\mathrm{SE}(3)$ affinity and tracking affinity losses effective weights of 0.1 and 0.05. The $\mathrm{SE}(3)$ affinity loss uses $\tau_e = 0.07$, $\tau_{\text{aff}} = 5.0$, $\sigma_R = 0.12$, and $\sigma_t = 0.06$, with up to $4{,}096$ observations sampled per sequence, and the tracking affinity loss uses $\sigma_{\text{trk}} = 0.005$. The $\mathrm{SE}(3)$ loss combines a geodesic-angle loss on rotation with a Huber loss ($\delta = 0.1$) on the norm of the residual at the anchor's support centroid $\bar{\mathbf{s}}$. The two terms are balanced by $\alpha = 20$. Both are computed from the per-anchor weighted Horn fit of Sec. A.2, which uses $4{,}096$ shared anchor and reference points, weights them with $\tau = 0.07$ in Eq. 2, and requires at least 8 valid points per fit.

A.2 Per-Anchor $\mathrm{SE}(3)$ Recovery via Weighted Horn

For the closed-form $\mathrm{SE}(3)$ fit of Eq. 1, we use an *anchor* set $\mathcal{A}$ of query-frame pixels for which we predict per-anchor $\mathrm{SE}(3)$, and a *reference* set $\mathcal{G}$ of query-frame pixels that supply the Procrustes support points. We set $\mathcal{G} = \mathcal{A}$, so every selected query pixel plays a dual role: it is an anchor whose $\mathrm{SE}(3)$ we predict and a Procrustes support point for every other anchor's fit. Both sets are re-sampled per frame pair (q, i) with $i \neq q$ from the query-frame pixels with ground-truth track supervision from q to i. We deterministically sub-sample every k-th valid pixel in raster order, yielding $|\mathcal{A}| = |\mathcal{G}| = 4{,}096$ with even spatial coverage.

For an anchor $\mathbf{u} \in \mathcal{A}$ and target frame i, the rigidity embeddings produce soft weights $\{w_{\mathbf{v}}\}_{\mathbf{v}\in\mathcal{G}}$ via Eq. 2, where we abbreviate $w_{\mathbf{v}} = w(\mathbf{u}, \mathbf{v})$. The anchor is excluded from its own support set, since its unit self-similarity would otherwise dominate the temperature-τ softmax. Writing the source/target

point pairs as $(\mathbf{s_v}, \mathbf{t_v}) = (\mathbf{X}_q^w(\mathbf{v}), \mathbf{X}_i^w(\mathbf{v}))$, we solve Eq. 1 in closed form by Horn's method [27]. We first form the weighted centroids and the weighted cross-covariance,

$$\bar{\mathbf{s}} = \sum_{\mathbf{v}\in\mathcal{G}} w_{\mathbf{v}}\, \mathbf{s_v}, \qquad \bar{\mathbf{t}} = \sum_{\mathbf{v}\in\mathcal{G}} w_{\mathbf{v}}\, \mathbf{t_v}, \qquad \Sigma = \sum_{\mathbf{v}\in\mathcal{G}} w_{\mathbf{v}}\, (\mathbf{s_v} - \bar{\mathbf{s}})(\mathbf{t_v} - \bar{\mathbf{t}})^\top. \tag{10}$$

We use Horn's quaternion representation rather than the SVD-based solution because it empirically provides more stable gradients in this differentiable training loop. We assemble the Horn matrix H from the entries of Σ; the optimal rotation $\mathbf{R}_{q\to i}(\mathbf{u})$ is the unit quaternion that solves an eigenproblem on H, and the optimal translation follows from centroid alignment,

$$\mathbf{t}_{q\to i}(\mathbf{u}) = \bar{\mathbf{t}} - \mathbf{R}_{q\to i}(\mathbf{u})\, \bar{\mathbf{s}}. \tag{11}$$

The fit is differentiable, so the resulting per-anchor $\mathbf{T}_{q\to i}(\mathbf{u})$ propagates gradients into both the rigidity-embedding head (through the weights) and the point-tracking head (through the source and reference positions).

A.3 Coordinate and Scale Conventions

Tracking parameterization and lifting. For a query pixel $\mathbf{u}$, the tracking head outputs normalized image coordinates $\mathbf{p}_i = (p_{i,x}, p_{i,y})$ and a raw depth parameter η_i for every target frame. Let $d = \frac{1}{2}\sqrt{H^2 + W^2}$. This (u, v, z) parameterization uses centered image coordinates and decoded depth, not direct XYZ regression or camera-bearing coordinates:

$$\begin{aligned} \mathbf{p}_i &= ((u_i^{\text{px}} - c_{x,i})/d,\ (v_i^{\text{px}} - c_{y,i})/d)\,, \qquad z_i = \sinh\big(\,\text{clip}(\eta_i, -7, 7)\big), \\ \mathbf{X}_i^c &= z_i\, [d\, p_{i,x}/f_x \quad d\, p_{i,y}/f_y \quad 1]^\top. \end{aligned} \tag{12}$$

Depth is signed, so valid tracks behind the camera can remain supervised. During training, ground-truth focal lengths $f_{x,i}^\star, f_{y,i}^\star$ replace f_x, f_y when decoding the tracks used by the 3D losses. No ground-truth intrinsics are required at inference.

Focal fitting. One focal-length pair is fitted per video from π^3's frozen pointmap bearings; there is no separately learned intrinsics head. The fit uses image pixel centers $(j + \frac{1}{2}, k + \frac{1}{2})$ and principal point $(W/2, H/2)$. For the horizontal bearing b_x and centered pixel coordinate x, it minimizes $\sum m(b_x - a_x x)^2$ and sets $f_x = 1/a_x$; the vertical fit is analogous. Here m selects finite bearings with predicted confidence above 0.1. Each focal is clamped to $[0.25d, 10d]$. Invalid or non-positive estimates, or fewer than 256 valid samples, trigger a fallback of d on each affected axis. The fitted calibration is detached. Ground-truth principal points are used when projecting ground-truth tracks to read target-frame embeddings for tracking affinity.

Scene normalization and scale alignment. Let $\mathcal{R}$ contain valid reconstruction pixels across the sequence. The predicted reconstruction $\mathbf{M}_i$ is in camera i's local frame. The prediction normalizer and the ground-truth normalizer are

$$\begin{aligned} \widehat{\nu} &= \frac{1}{|\mathcal{R}|} \sum_{(i,\mathbf{u})\in\mathcal{R}} \|\mathbf{M}_i(\mathbf{u})\|_2, \\ \nu^\star &= \frac{1}{|\mathcal{R}|} \sum_{(i,\mathbf{u})\in\mathcal{R}} \|(\mathbf{P}_1^\star)^{-1} \cdot \mathbf{M}_i^{w\star}(\mathbf{u})\|_2. \end{aligned} \tag{13}$$

Thus the ground-truth reconstruction norm uses the fixed first-camera frame, whereas the prediction norm uses per-frame local coordinates. Predicted and ground-truth 3D quantities, including camera translations, are divided by their respective normalizers. A further detached scalar s aligns normalized predicted local reconstruction to normalized ground-truth local reconstruction using the inverse-depth-weighted ℓ_1 scale solver. It uses $4{,}096$ nearest-resampled valid reconstruction points; the absolute solution is clamped to $[0.1, 5]$. Define the aligned training tracks

$$\widetilde{\mathbf{X}}_i^c = (s/\widehat{\nu})\mathbf{X}_i^c, \qquad \widetilde{\mathbf{X}}_i^{c\star} = \mathbf{X}_i^{c\star}/\nu^\star. \tag{14}$$

The same scalar s is used for depth, surface-gradient, temporal-displacement, and $\mathrm{SE}(3)$ supervision; it is estimated from reconstruction, not from the tracks. UV coordinates are scale invariant. We adopt the per-sequence scale-alignment solver of MoGE [91] (also used in π^3 [93]).

A.4 Tracking Loss Details

UV and depth losses. Let $\mathbf{p}_i^\star$ denote the normalized image-coordinate target. Let Ω_{uv} contain valid tracks with finite predicted and target image coordinates and $\|\mathbf{p}_i^\star\|_2 < 20$. Let Ω_z contain valid tracks with finite predicted and target 3D coordinates and finite depth weights. Then

$$\begin{aligned} \mathbf{p}_i^\star &= \operatorname{diag}(f_{x,i}^\star/d, f_{y,i}^\star/d)\, \mathbf{X}_{i,xy}^{c\star}/z_i^\star, \\ \mathcal{L}_{uv} &= \frac{1}{2|\Omega_{uv}|} \sum_{(i,\mathbf{u})\in\Omega_{uv}} \|\mathbf{p}_i(\mathbf{u}) - \mathbf{p}_i^\star(\mathbf{u})\|_1, \\ \mathcal{L}_z &= \frac{1}{|\Omega_z|} \sum_{(i,\mathbf{u})\in\Omega_z} w_i^z(\mathbf{u}) \left| \frac{s}{\widehat{\nu}} z_i(\mathbf{u}) - \frac{z_i^\star(\mathbf{u})}{\nu^\star} \right|. \end{aligned} \tag{15}$$

The depth weight w_i^z is the capped inverse-depth weight of Eq. 19. The UV term is not inverse-depth weighted, and its two coordinates are averaged. Off-screen and behind-camera targets are retained when valid; the radius threshold removes near-zero-depth projection outliers. There is no additional XYZ position loss in the training objective.

Query-frame surface-gradient loss. Only the query-frame track map is a surface pointmap on its own image grid. We compare its finite differences against dense ground-truth reconstruction, not against the frozen predicted pointmap. With residual $\mathbf{r}_q(\mathbf{u}) = \widetilde{\mathbf{X}}_q^c(\mathbf{u}) - \mathbf{M}_q^{c\star}(\mathbf{u})/\nu^\star$, let $\mathcal{E}_q$ contain horizontal and vertical adjacent pixel pairs with valid reconstruction and finite predictions and targets at both endpoints. Writing each edge as $(\mathbf{u}, \boldsymbol{\delta})$, where $\boldsymbol{\delta} \in \{(1,0),(0,1)\}$,

$$\mathcal{L}_\nabla = \frac{1}{3|\mathcal{E}_q|} \sum_{(\mathbf{u},\boldsymbol{\delta})\in\mathcal{E}_q} w_q^\nabla(\mathbf{u}) \|\mathbf{r}_q(\mathbf{u}+\boldsymbol{\delta}) - \mathbf{r}_q(\mathbf{u})\|_1. \tag{16}$$

The weight is taken at the left/top endpoint. This term requires dense query-frame reconstruction ground truth; sparse tracking labels alone do not suffice.

Temporal-displacement loss. We compare consecutive sampled-frame displacements in a fixed query-camera frame. Let $(\mathbf{R}_{qi}^\pi, \mathbf{t}_{qi}^\pi) = \mathbf{P}_q^{-1}\mathbf{P}_i$ and $(\mathbf{R}_{qi}^{\pi\star}, \mathbf{t}_{qi}^{\pi\star}) = (\mathbf{P}_q^\star)^{-1}\mathbf{P}_i^\star$ denote predicted and ground-truth relative cameras before scale normalization. The transported tracks are

$$\mathbf{Y}_i = \frac{s}{\widehat{\nu}}(\mathbf{R}_{qi}^\pi \mathbf{X}_i^c + \mathbf{t}_{qi}^\pi), \qquad \mathbf{Y}_i^\star = \frac{1}{\nu^\star}(\mathbf{R}_{qi}^{\pi\star}\mathbf{X}_i^{c\star} + \mathbf{t}_{qi}^{\pi\star}). \tag{17}$$

Camera translations are scaled together with the point coordinates. With Ω_Δ containing consecutive sampled-frame pairs with valid tracks and finite transported points at both endpoints,

$$\mathcal{L}_\Delta = \frac{1}{3|\Omega_\Delta|} \sum_{(i,\mathbf{u})\in\Omega_\Delta} w_i^\Delta(\mathbf{u}) \big\|(\mathbf{Y}_{i+1} - \mathbf{Y}_i)(\mathbf{u}) - (\mathbf{Y}_{i+1}^\star - \mathbf{Y}_i^\star)(\mathbf{u})\big\|_1. \tag{18}$$

The weight uses the earlier frame. This matches ground-truth displacement, not zero motion, and is neither a moving-camera-frame difference nor a velocity divided by the sampling interval.

Depth weights and reductions. Depth and temporal supervision use capped inverse-depth weights:

$$\begin{aligned} a_i(\mathbf{u}) &= \max(|z_i^\star(\mathbf{u})|/\nu^\star, 10^{-3}), \qquad \mu_i = \frac{\sum_{\mathbf{u}} m_i(\mathbf{u}) a_i(\mathbf{u})}{\sum_{\mathbf{u}} m_i(\mathbf{u}) + HW \cdot 10^{-7}}, \\ w_i(\mathbf{u}) &= \frac{1}{\max(a_i(\mathbf{u}), 0.1\mu_i) + 10^{-6}}. \end{aligned} \tag{19}$$

For w^z, the mask m uses positive-depth valid tracks when any such track exists in the batch, and otherwise all valid tracks. For w^Δ, it uses valid finite transported tracks. For w^∇, the same construction uses positive-clamped normalized dense reconstruction depth at q and valid finite surface pixels, rather than absolute track depth. Surface and temporal residuals average their three coordinates. Each loss is divided by the number of valid observations or edges, not by the sum of depth weights. Empty eligible sets contribute zero. The shared UV/depth/visibility helper returns zero when the batch has no valid tracking or reconstruction points.

Visibility supervision. Let Ω_{vis} contain valid annotated tracks with finite visibility logits, and let $V_i^\star$ be the binary visibility flag. With σ the sigmoid,

$$\mathcal{L}_{\text{vis}} = -\frac{1}{|\Omega_{\text{vis}}|} \sum_{(i,\mathbf{u})\in\Omega_{\text{vis}}} \left[V_i^\star \log \sigma(V_i) + (1 - V_i^\star) \log(1 - \sigma(V_i))\right]. \tag{20}$$

Indices are suppressed inside the brackets. No class-balancing or depth weight is applied, and the term is omitted when visibility annotations are absent.

A.5 Dataset

For each training sequence, we randomly sample 16 to 24 frames at a stride drawn uniformly from $\{1, 2, 3, 4\}$, and load up to 24 frames of the sampled sequence per GPU. Resolution is randomized per batch, with aspect ratio in $[0.5, 2.0]$ and pixel count in $[100\text{k}, 255\text{k}]$. The query frame index also varies per batch.

The training mix draws from the Kubric CoTracker3 split [20, 39], Art-Kubric, Syn4D [35], SynthVerse [114], PointOdyssey [115], and Dynamic Replica [38], sampled per sequence at 26%, 26%, 20%, 16%, 8%, and 4% respectively. Among these, the Kubric CoTracker3 split, Art-Kubric, and Syn4D provide a per-pixel rigid-body partition and per-rigid-body $\mathrm{SE}(3)$ labels. On Syn4D we derive these labels through post-processing, including for deformable objects and human bodies. The $\mathrm{SE}(3)$ affinity loss and the $\mathrm{SE}(3)$ loss are supervised on these three sources only, while the tracking affinity loss needs no such labels and applies to all six. The track and visibility losses are likewise supervised on all six sources.

B Architecture Details

B.1 Tracking Branch

π^3-Large has 36 decoder layers. The encoder and decoder layers 0–9 are shared and frozen. Layers 10–35 are paralleled by a 26-layer trainable tracking branch, with each tracking-branch block initialized from its same-layer π^3 decoder counterpart. Specifically, the fused QKV is duplicated into two copies of π^3's QKV weights: a trainable tracking-side query/key/value triple that processes the tracking tokens, and a frozen geometry-side key/value pair that reads the same-layer geometry tokens from the frozen geometry branch. The remaining components — LayerNorms, layer scaling, and MLP — are deep-copies of the π^3 block, with the geometry-side LayerNorms held frozen and the tracking-side trainable. Each block runs a single attention where the queries come from the tracking tokens alone, while the keys and values concatenate the tracking-side and geometry-side projections along the token axis, so one softmax allocates attention between the two streams without an explicit gate.

B.2 Point-Tracking Head

The point-tracking head consists of 10 transformer blocks alternating frame-wise and global self-attention. Each block is conditioned on query-frame identity through Adaptive Layer Norm [70]: AdaLN routes each token through one of two learned (γ, β) pairs based on whether it comes from the query frame I_q or from one of the other $N - 1$ frames. Two parallel linear pointmap-style outputs follow the block stack: one predicts the normalized image coordinates $\mathbf{p}_i(\mathbf{u})$ and raw depth parameter $\eta_i(\mathbf{u})$ at every pixel, which Eq. 12 lifts to the 3D track $\mathbf{X}_i^c(\mathbf{u})$, and the other predicts the per-pixel visibility logit $V_i(\mathbf{u})$. $\mathbf{p}_i$ and η_i are absolute target-frame values, not offsets from the query pixel.

C Evaluation

C.1 Ablation Setup and Full Results

Tab. 4 reports the full ablation across both datasets and both evaluation granularities. The main paper (Tab. 3) reports only per-pixel $\mathrm{SE}(3)$ on iTACO.

All ablations in Tab. 4 share an identical training schedule, so any difference in the reported metrics reflects only the variable we change. Each variant is trained as a single stage with all losses enabled,

Table 4: **Full ablation study.** We ablate four design choices, each compared against our full model (bottom row). **(a) Architecture:** directly regressing $\mathrm{SE}(3)$ from an $\mathrm{SE}(3)$ head, instead of predicting rigidity embeddings and point tracks. **(b) Training data:** training without Art-Kubric dataset. **(c) Training strategy:** predicting tracks alone and recovering $\mathrm{SE}(3)$ posthoc, via either k-NN clustering of tracks or clustering of our rigidity embeddings, whose cluster IoU is inherited from our full model and therefore omitted (–). **(d) Loss design:** removing individual loss terms. ADD is reported in centimeters; AUC_r uses a $30°$ threshold, while $\mathrm{AUC}_{\mathrm{ADD}}$ uses a 10 cm threshold on HO3D and 20 cm on iTACO. Best in each column in **bold**, second underlined.

	Per-pixel $\mathrm{SE}(3)$								Object/Part-level $\mathrm{SE}(3)$							
	HO3D				iTACO				HO3D				iTACO			
Variant	RRE↓	AUC_r ↑	ADD↓	$\mathrm{AUC}_{\mathrm{ADD}}$ ↑	RRE↓	AUC_r ↑	ADD↓	$\mathrm{AUC}_{\mathrm{ADD}}$ ↑	RRE↓	ADD↓	$\mathrm{AUC}_{\mathrm{ADD}}$ ↑	IoU↑	RRE↓	ADD↓	$\mathrm{AUC}_{\mathrm{ADD}}$ ↑	IoU↑
(a) Architecture																
Direct $\mathrm{SE}(3)$ regr.	29.79	0.449	3.65	0.655	24.57	0.525	19.05	0.393	36.18	3.21	0.701	0.600	22.49	15.01	0.511	0.620
(b) Training data																
w/o Art-Kubric	**25.07**	**0.480**	2.94	0.716	22.89	0.566	17.01	0.462	**23.24**	2.86	0.734	0.643	19.06	15.32	0.502	0.670
(c) Training strategy																
Track-only + k-NN	41.71	0.334	3.61	0.652	31.11	0.453	20.96	0.382	39.32	3.79	0.650	0.450	27.61	18.86	0.461	0.322
Track-only + rigid clust.	42.59	0.333	3.60	0.657	24.30	0.550	18.01	0.460	41.71	3.56	0.671	–	23.44	17.00	0.488	–
(d) Loss design																
w/o rigidity loss	27.38	0.456	2.94	**0.720**	18.97	0.614	15.83	0.480	26.09	3.13	0.715	0.579	15.95	13.12	0.562	0.663
w/o $\mathrm{SE}(3)$ loss	43.28	0.332	3.59	0.655	23.90	0.555	17.82	0.465	41.79	3.60	0.667	0.656	21.09	15.98	0.516	**0.702**
Ours (full)	25.53	0.473	**2.90**	**0.720**	**17.27**	**0.647**	**14.09**	**0.517**	24.57	**2.85**	**0.735**	**0.689**	**14.66**	**12.49**	**0.564**	0.685

for 80 epochs of 800 iterations on $4\times$ NVIDIA H200 GPUs, with per-GPU batch 1 and no gradient accumulation, giving an effective batch of 4 sequences per optimizer step. We optimize with AdamW at peak learning rate 2×10^{-5}, following a cosine schedule that decays to 2×10^{-10} over training. Dataset mix and loss weights are otherwise identical to those in Sec. A.1. For the *Track-only + rigid clust.* variant in group (c), the track-only model itself does not produce a rigidity embedding, so we cluster on the rigidity embedding from our full-loss model rather than training a separate embedding network.

C.2 $\mathrm{SE}(3)$ Evaluation Detail

Ground-truth processing. We evaluate on 54 clips from HO3D, 55 clips from iTACO, and 9 clips from YCBInEOAT. On HO3D, ground-truth labels are restricted to the B-channel object segmentation, so hand pixels are excluded from all per-pixel and clustering metrics. On iTACO, each clip has multiple actors; we select a single *moving* actor per clip via the largest end-to-start pose difference (matching the iTACO release's heuristic), restrict per-pixel metrics to that actor, and use the binary $\{\text{moving}, \text{not-moving}\}$ partition as the cluster-IoU GT mask. On YCBInEOAT, each clip contains a single manipulated object with ground-truth pose; we restrict per-pixel and clustering metrics to that object, excluding robot-arm pixels, and use its segmentation mask as the cluster-IoU GT mask. Pixels excluded from these metrics remain in the input — they serve as Procrustes neighbors for baselines and as clustering negatives for MoSE3. Before computing any metric, we align predicted tracks to ground truth once per clip by fitting a global scale and translation over the pixels that are visible and have ground-truth $\mathrm{SE}(3)$ labels, then apply the same alignment to all predicted points.

Per-pixel $\mathrm{SE}(3)$. We compute per-pixel $\mathrm{SE}(3)$ by solving a weighted Horn-Procrustes problem [27] for each labeled query pixel. What differs across methods is the support set and weighting. Baselines have only point tracks, so for each query pixel we take its K nearest 3D neighbors at frame 0 and weight them uniformly. The optimal K depends on scene composition and object scale, so we sweep $K \in \{4, 8, 16, 32, 64, 128, 256\}$ and select per clip the K that minimizes a z-score composite of mean RRE, mean ADD, $-\mathrm{AUC}_r@30°$, and $-\mathrm{AUC}_{\mathrm{ADD}}$ at the dataset-scaled ADD threshold. MoSE3, in contrast, predicts a rigidity embedding that directly identifies which pixels are moving rigidly together. For each query pixel, we weight *every* other pixel by a softmax over its cosine similarity to the query's rigidity embedding at temperature $\tau{=}0.01$, sharper than the 0.07 used during training, with no explicit neighborhood truncation, so each Procrustes fit pools tracks that the embedding has already grouped as rigidly co-moving. We aggregate the resulting per-pixel transforms over labeled pixels at $t{>}0$ and report mean RRE (degrees), mean ADD (cm), AUC_r up to $30°$, and $\mathrm{AUC}_{\mathrm{ADD}}$ up to a dataset-scaled threshold (10 cm on HO3D and YCBInEOAT, 20 cm on iTACO). ADD is measured on ground-truth surface samples at frame 0, transformed by the predicted and the ground-truth pose.

Table 5: 3D point tracking results on PointOdyssey, ADT, and PStudio in both camera and world coordinates. We follow the evaluation protocol of Track4World [61]. We report tracking accuracy at horizons L-16 (↑) and L-50 (↑). Best in **bold**, second underlined. Avg. is the unweighted mean across the three datasets.

	PointOdyssey [115]		ADT [69]		PStudio [36]		Avg.	
Method	L-16	L-50	L-16	L-50	L-16	L-50	L-16	L-50
Camera coordinate 3D tracking								
SpatialTracker [100]	0.3116	0.2977	0.4962	0.4692	0.5390	0.4991	0.4489	0.4220
DELTA [67]	0.3529	0.3412	0.5116	0.4952	0.5922	0.5533	0.4856	0.4632
SpatialTrackerV2 [101]	0.1864	0.1785	0.2400	0.2330	0.3784	0.3690	0.2683	0.2602
MASt3R [50]	0.3546	0.3253	0.3368	0.3029	0.3293	0.2956	0.3402	0.3079
MonST3R [111]	0.3912	0.3860	0.3694	0.3429	0.3511	0.3381	0.3706	0.3557
POMATO [113]	0.4816	0.4623	0.5338	0.5299	0.5163	0.4726	0.5106	0.4883
ZeroMSF [54]	0.4214	0.3887	0.5382	0.4635	0.5083	0.4524	0.4893	0.4349
Track4World [61]	0.5397	0.5268	**0.6501**	0.6091	0.5948	0.5423	0.5949	0.5594
Ours	**0.5666**	**0.5688**	0.5951	**0.6146**	**0.6391**	**0.6207**	**0.6003**	**0.6014**
World coordinate 3D tracking								
SpatialTrackerV2 [101]	0.1925	0.1763	0.2456	0.2163	0.3790	0.3689	0.2724	0.2538
POMATO [113]	0.4425	0.3905	0.3611	0.3548	0.5166	0.4713	0.4401	0.4055
ZeroMSF [54]	0.4053	0.3505	0.4530	0.3563	0.4828	0.4386	0.4471	0.3818
Any4D [41]	0.4769	0.4174	0.4460	0.3717	0.5707	0.5066	0.4979	0.4319
V-DPM [78]	0.4848	0.4233	0.4783	0.3759	0.6084	0.5795	0.5238	0.4596
Track4World [61]	0.5345	0.5162	**0.6250**	0.5622	0.5946	0.5422	0.5847	0.5402
Ours	**0.5635**	**0.5650**	0.5789	**0.5953**	**0.6400**	**0.6235**	**0.5941**	**0.5946**

For object-level evaluation, we cluster a per-pixel feature with HDBSCAN (min_cluster_size $= 30$, EOM cluster selection) and match the resulting clusters to the ground-truth mask by maximum IoU. We then re-fit a single $\mathrm{SE}(3)$ per frame on the matched cluster by pooling *all* the tracks it contains. At each target frame t, we solve one weighted Procrustes between the frame-0 positions and frame-t positions of every track assigned to the cluster, yielding a single $(\hat{R}_t, \hat{\mathbf{t}}_t)$ that summarizes the object's motion. We construct the per-pixel feature fed to HDBSCAN differently for each method. For MoSE3, we use the L2-normalized rigidity embedding directly as the cluster feature, since it is trained to group rigidly co-moving pixels, and weight each track in the cluster Procrustes fit by softmax-cosine similarity to the cluster centroid (τ=0.1), so tracks more confidently inside the object contribute more. For baselines, we form the feature by stacking each pixel's recovered $\mathrm{SE}(3)$ at two reference frames ($T/2$ and $T-1$) into a 24-D vector, and weight all tracks in the cluster fit uniformly. We report mask IoU and the matched cluster's per-frame RRE, ADD, and $\mathrm{AUC}_{\mathrm{ADD}}$, averaged across clips.

ProxyPose. ProxyPose [112] predicts a pose per query rather than dense per-pixel motion, so we report it only at the object/part level and evaluate it under its own protocol with ground-truth intrinsics. Because its video generation is computationally expensive, we evaluate its one-query variant. We omit its ADD, because its PnP-based proxy-cube tracker diverges on 5 of 54 HO3D clips and 7 of 55 iTACO clips, producing translations of hundreds of meters that dominate the mean. $\mathrm{AUC}_{\mathrm{ADD}}$ and RRE remain comparable, since diverged clips simply fall outside the AUC threshold and rotation is unaffected.

C.3 Additional Results

Tab. 5 reports 3D point tracking results in both camera and world coordinates. The main paper (Tab. 2) reports only world-coordinate results. For comparability with the results reported by Track4World [61], we use its released evaluation code unchanged. Before the standard median scaling of TAPVid-3D [45], this code fits a per-sequence scale and shift to the predicted tracks using only the points that are occluded in the ground truth.

Table 6: $\mathrm{SE}(3)$ **estimation on YCBInEOAT [95].** Nine clips of robot manipulation with rigid objects, evaluated with the protocol of Tab. 1. ADD is reported in centimeters; AUC_r uses a 30° threshold and $\mathrm{AUC}_{\mathrm{ADD}}$ a 10 cm threshold. Best in **bold**, second underlined.

	Per-pixel $\mathrm{SE}(3)$				Object/Part-level $\mathrm{SE}(3)$			
Method	RRE↓	ADD↓	AUC_r ↑	$\mathrm{AUC}_{\mathrm{ADD}}$ ↑	RRE↓	ADD↓	$\mathrm{AUC}_{\mathrm{ADD}}$ ↑	IoU↑
SpatialTrackerV2 [101]	21.93	4.93	0.461	0.576	22.93	6.55	0.508	0.251
POMATO [113]	46.47	10.78	0.223	0.336	50.11	10.89	0.309	0.054
ZeroMSF [54]	44.79	11.57	0.243	0.353	46.71	10.74	0.387	0.073
Any4D [41]	42.30	10.28	0.251	0.432	44.67	10.45	0.455	0.267
V-DPM [78]	30.42	7.58	0.329	0.504	31.60	8.18	0.501	0.234
TraceAnything [58]	44.79	9.61	0.249	0.415	48.47	10.18	0.392	0.066
4RC [63]	21.18	4.85	0.492	0.614	24.23	4.52	0.625	0.322
Track4World-Pi3x [61]	19.92	7.68	0.441	0.562	19.91	7.53	0.549	0.360
Track4World-DA3 [61]	22.46	5.59	0.399	0.560	28.89	6.11	0.542	0.278
ProxyPose† [112]	–	–	–	–	17.01	–	0.515	–
Ours (k-NN)	17.97	4.50	0.546	0.595	14.02	**3.67**	**0.658**	0.171
Ours (rigid clust.)	**14.89**	**3.98**	**0.588**	**0.628**	**11.93**	4.00	0.638	**0.498**

†ProxyPose predicts a pose per query rather than a dense per-pixel field; we evaluate its single-query variant under its own protocol with ground-truth intrinsics, so it is reported at the object/part level only and has no predicted mask for IoU. We omit its ADD, whose mean is dominated by a few clips where its tracker diverges (Appendix C.2).

Tab. 6 reports $\mathrm{SE}(3)$ estimation on YCBInEOAT [95], nine real-world clips of a robot arm manipulating rigid objects, following the protocol of Sec. 5.1. MoSE3 achieves the best result on every metric at both granularities.

C.4 Qualitative Visualizations

Fig. 5 compares MoSE3 against ProxyPose [112] on in-the-wild videos, including deformable cases. MoSE3 produces more stable and coherent axes across both rigid and deformable scenes. We further show additional in-the-wild qualitative results in Fig. 6, complementing Fig. 4 in the main paper. The in-the-wild videos in Figs. 1, 4, 5, and 6 and on our project page are drawn from DAVIS [72], DyCheck [19], EPIC-KITCHENS-100 [11], PhysTwin [33], AgiBot World [6], BridgeData V2 [84], and DROID [44], as well as from stock footage on Pexels.

D Art-Kubric: Additional Details

Production Setting. 5,000 scenes at 512×512, 120 frames at 60 fps, 32,768 tracks per scene with 7% background reserve. Built on the SAPIEN [98] simulator with 3,018 articulated URDFs pooled from PartNet-Mobility [66, 98] (1,034 models / 46 categories), Infinigen-Articulated [37] (1,199 / 7), and Articraft [117] (785 / 226), 1,033 Google Scanned Objects [16] as the rigid pool, and HDRI Haven [20] environment maps (509) for image-based lighting plus a tonemapped circular ground projection. Every asset is load-tested with the simulator's own URDF builders before it enters the pool.

Per-Scene Randomization. Vertical field of view is drawn per scene over 18–90°. Replayed through our loader's crop augmentation, that band reaches below SynthVerse's p05 of 15.9° and past PointOdyssey's p95 of 75.4°, which no single render FOV can, since cropping only narrows a view. Camera radii compensate by $\left(\tan(57.3^\circ/2)/\tan(\mathrm{fov}/2)\right)^{\alpha}$ with $\alpha \sim \mathcal{U}(0.75, 1)$ drawn per scene: at $\alpha = 1$ the camera distance is an exact function of the field of view ($R^2 = 0.915$), so a model could read the focal length off apparent object size rather than perspective; the random exponent breaks that dependence ($R^2 = 0.884$). Distance variety comes instead from the 4–7 m camera shell. A per-scene lighting level $u \sim \mathcal{U}(0, 0.9)$ runs from a strong directional key to a flat environment-only look, and spawn clearance and footprint are drawn below the median object size so that objects can make contact. Each quantity has its own random stream derived from the scene seed, so enabling one leaves the others bit-identical.

Annotations Exported per Scene. RGB; float32 and uint16 depth; surface normals; three segmentation granularities (*link* / *rigid-part* / *object*); per-link world-frame poses $\mathbf{T}_l^w(t) \in \mathrm{SE}(3)$; oriented 3D bounding boxes; full camera intrinsics and extrinsics; object-coordinate maps (canonical-frame

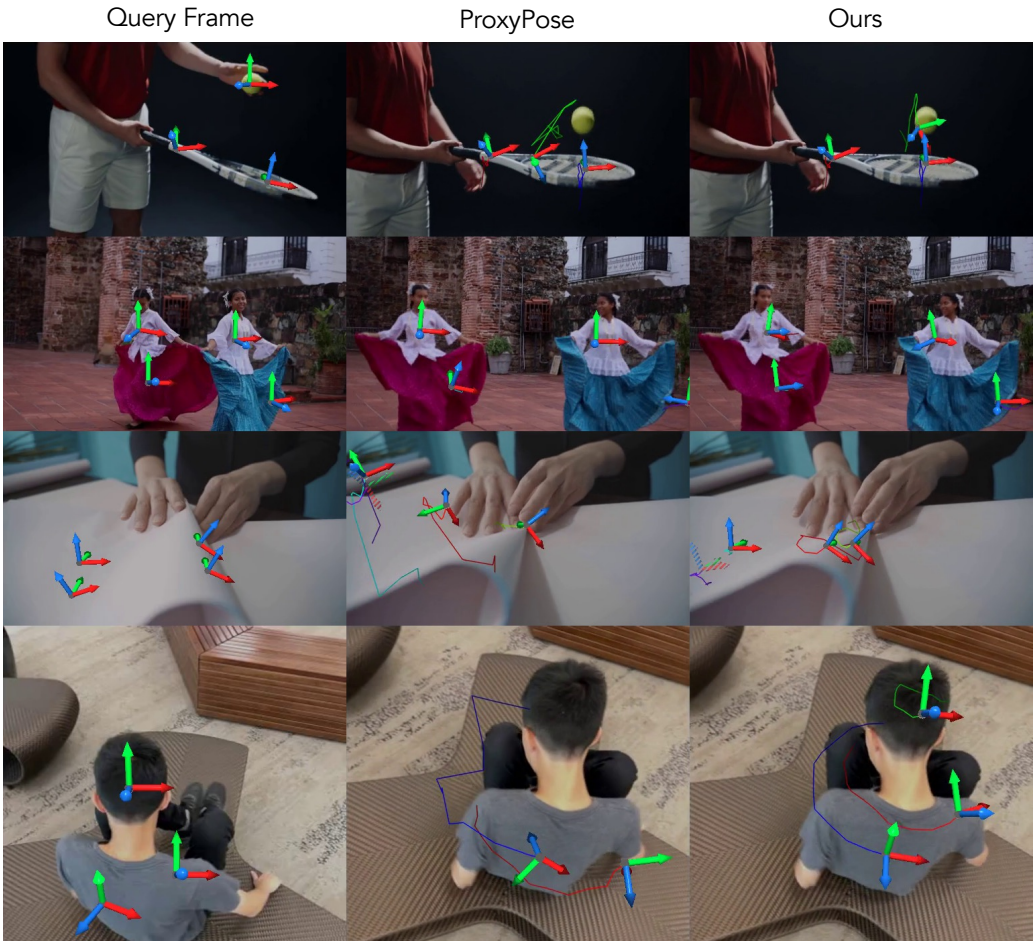

Figure 5: **Qualitative comparison on in-the-wild videos.** We compare MoSE3 against ProxyPose [112] on out-of-distribution real-world scenes, including deformable cases. For each method, we visualize the estimated per-pixel $\mathbf{T}_{q\to i}(\mathbf{u})$ by drawing its $\mathrm{SE}(3)$ coordinate axes at sampled pixels.

surface positions); 2D and 3D long-range trajectories with occlusion flags. The rigid-part (rp) segmentation collapses links sharing identical rigid motion (with 10^{-3} tolerance on translation and rotation) into a single class, so the supervision target for rigid grouping is a partition of pixels by *shared transform*, not by instance.

Statistics. A typical scene contains 22 assets ($\sim$14 dynamic) decomposed into $\sim$100 kinematic links that collapse into $\sim$ 85 rigid-part segments; 90% of scenes contain at least one articulated object, and rigid assets account for 30% of all placed objects. With $32{,}768$ tracks per clip across $5{,}000$ scenes this yields $\sim$163M long-range tracks, each carrying a per-pixel rigid-group label and a link-level $\mathrm{SE}(3)$ transform at every one of its 120 frames.

Extensibility. The released generation pipeline allows the community to re-render new clips under custom lighting, scenes, and asset distributions; a single dataset-level knob sets the realized share of each asset source, which the pipeline solves for from measured per-source placement rates.

License. Because some Art-Kubric scenes render PartNet-Mobility objects, whose terms permit only non-commercial research and educational use, we will release the Art-Kubric data (renders, depth, segmentation, tracks, and $\mathrm{SE}(3)$ annotations) under CC BY-NC 4.0. The generation pipeline does not redistribute any upstream asset, so users who re-render scenes obtain each asset source

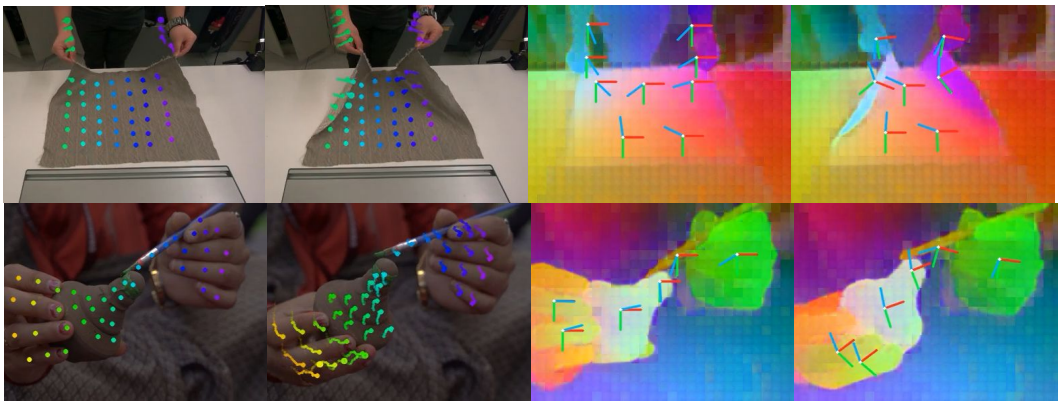

Figure 6: **Additional qualitative results on in-the-wild videos.** In addition to Fig. 4, we show further out-of-distribution real-world scenes spanning both rigid and non-rigid cases, here also visualizing the predicted 3D point tracks alongside the per-pixel rigidity embeddings and the per-pixel $\mathrm{SE}(3)$ motion overlaid on them.

Table 7: **Comparison of supervision signals across point-tracking and articulation datasets.** The point of this table is *which* annotation channels and scene regimes each dataset exposes, not raw scale. "Phys. multi-body" = multiple independently moving bodies that interact under physics simulation (collisions, contact). "Articul." = bodies whose motion is a kinematic tree of multiple rigidly-moving links. "3D tracks" = dense 3D point trajectories over > 2 frames. "Per-link $\mathrm{SE}(3)$" = world-frame 4×4 pose for each sub-object rigid link, every frame; ○ = only an object-level (single-rigid-body) pose. "Rigid-grouping" = per-pixel partition of pixels by shared transform, beyond instance segmentation. "Cam. variety" = multiple distinct camera trajectory modes (mixing moving and stationary), in the spirit of Kubric's MOVi setups. Among the datasets compared here, Art-Kubric is the only one that provides all seven properties.

Dataset	Source	Phys. multi-body	Articul. (multi-link)	Long-range tracks	3D track GT	Per-link $\mathrm{SE}(3)$ GT	Rigid-grouping	Cam. variety
FlyingThings3D [64]	Synth	✗	✗	✗	✓$^{\flat}$	✗	✗	✓
Kubric MOVi-F [20]	Synth	✓	✗	✓	✓$^{\sharp}$	○	✗	✓
TAP-Vid Kubric [12]	Synth	✓	✗	✓	✗	✗	✗	✓
PointOdyssey [115]	Synth	○$^{\diamond}$	✗†	✓	✓	✗	✗	✓
Dynamic Replica [38]	Synth	✗†	✗†	✓	✓	✗	✗	✓
SynthVerse [114]	Synth	✗‡	✓‡	✓	✓	✗	✗	✗‡
ARCTIC [17]	Real	✗	✓	○*	○*	✓	○*	✗
HOI4D [59]	Real	✗	✓	○*	✗	✓	✗	✗
Art-Kubric (ours)	Synth	✓	✓	✓	✓	✓	✓	✓

$^{\flat}$ frame-pair scene flow only, not long-range. $^{\sharp}$ not released as track files; computable from depth + per-instance pose.

† foreground content is dominated by skinned (not piecewise-rigid) humans/animals; the "Articul." column requires kinematic trees of *rigidly-moving* links, which skinned characters do not satisfy. Dynamic Replica's backgrounds are static scanned environments and have no rigid-body physics. $^{\diamond}$ PointOdyssey applies random forces with realistic collisions on auxiliary GSO/PartNet rigid objects, but this multi-body physics is secondary to the character animation that dominates each scene. ‡ SynthVerse's articulated split renders one articulated asset per scene in an HDR surround (no other dynamic objects), drives motion by random joint-angle sampling within valid limits rather than by interaction forces (executed in Isaac Sim), and uses a fixed four-view rig plus one orbiting camera ([114], §3.3). The articulated source pool combines PartNet-Mobility [66, 98] and Infinite Mobility [53]; released annotations contain only point-track fields and instance masks (no per-link $\mathrm{SE}(3)$). * not packaged as track/grouping files; derivable from released per-frame mesh + per-part 6D pose. ARCTIC's hands are MANO-parametric (not piecewise-rigid), so its rigid-grouping covers only the two rigid object parts.

themselves and must comply with its license, namely PartNet-Mobility (non-commercial research and educational use under the ShapeNet terms), Infinigen-Articulated (BSD 3-Clause), Articraft (CC BY 4.0), Google Scanned Objects (CC BY 4.0), and HDRI Haven (CC0).

E Broader Impacts

MoSE3 converts monocular RGB video into a structured per-pixel $\mathrm{SE}(3)$ motion field, with positive applications in 4D reconstruction, robotics, and physical reasoning. Potential negative uses stem from fine-grained motion analysis of humans (e.g., surveillance), but our model outputs geometric motion only and is trained without human identity supervision. We do not foresee additional harms beyond those already present in monocular geometry/tracking models built on π^3-class foundations.

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: The abstract and the contribution list in Sec. 1 state four claims: (1) MoSE3, the first feed-forward model for dense $\mathrm{SE}(3)$ motion from monocular RGB video, (2) the decomposition of $\mathrm{SE}(3)$ into 3D point tracks and rigidity embeddings connected by a differentiable closed-form fit, (3) the Art-Kubric dataset and generation pipeline, and (4) state-of-the-art $\mathrm{SE}(3)$ estimation at pixel, part, and object levels together with state-of-the-art average 3D point tracking accuracy. Sec. 4 and Appendix D support (3), Sec. 5.1 and Sec. 5.2 support (4), and the ablations in Sec. 5.3 support (2) and the contribution of Art-Kubric.

   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: Sec. 6 has a dedicated Limitations paragraph. MoSE3 depends on π^3 for camera poses and intrinsics, so errors in either propagate into the recovered $\mathrm{SE}(3)$. Its $\mathrm{SE}(3)$ accuracy degrades when the predicted tracks or rigidity embeddings degrade, most notably under fast motion. Deformable scenes are evaluated only qualitatively, and a quantitative benchmark is left to future work.

   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: [N/A]

   Justification: The paper does not introduce new theoretical results. The closed-form $\mathrm{SE}(3)$ fit used in our pipeline is the standard Horn quaternion solution [27], and its concrete instantiation (weighted centroids, cross-covariance, and quaternion eigenproblem) is given in Appendix A.2.

   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: Sec. 3 describes the architecture and losses. Appendix A gives the full training recipe (initialization, optimizer and learning-rate schedule, loss weights, mixed precision, coordinate and scale conventions, tracking-loss details, and training-data mix). Appendix A.2 gives the closed-form $\mathrm{SE}(3)$ fit, Appendix B the tracking branch and point-tracking head, Appendix C the ablation and evaluation protocols, and Appendix D the Art-Kubric generation settings.

   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: [Yes]

   Justification: We provide open access to the code, the Art-Kubric dataset, and its generation pipeline, with the data under CC BY-NC 4.0 as stated in Appendix D. The implementation, training, and evaluation details in Appendices A–C accompany the release with what is needed to reproduce the main results.

   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: Sec. 5 gives the benchmarks, baseline adaptation, and metrics. Appendix A.1 gives the optimizer, learning-rate schedule, loss weights, batch size, and mixed precision. Appendix A.5 gives the training mix and frame sampling, Appendix C.1 the ablation schedule, and Appendix C.2 the evaluation clips, ground-truth processing, and $\mathrm{SE}(3)$ protocols, including the per-clip K selection rule for baselines.

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: Following common practice in recent 3D tracking and motion-estimation work [39, 61, 93], we report single-run results because a full training run takes 160,000 optimizer steps on 64 NVIDIA H100 GPUs. The $\mathrm{SE}(3)$ margins in Tab. 1 are large (e.g., per-pixel RRE on HO3D falls from 33.21° for the strongest baseline to 18.28°). The average 3D tracking margin over Track4World in Tab. 2 is small at L-16 (0.009), so we do not claim that it exceeds run-to-run variance.

   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: Appendix A.1 reports compute for the main model, which trains on 64 NVIDIA H100-80GB GPUs with per-GPU batch 1 and no gradient accumulation for 200 epochs of 800 steps (160,000 optimizer steps). Appendix C.1 reports compute for each ablation variant, which trains on 4 NVIDIA H200 GPUs for 80 epochs of 800 iterations.

   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 reviewed the NeurIPS Code of Ethics, and this work conforms with it. We train on synthetic data, evaluate on public benchmarks, show qualitative results on publicly available research datasets and stock footage, and conduct no human-subjects research.

   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: Appendix E discusses positive applications in 4D reconstruction, robotics, and physical reasoning, as well as potential negative uses such as surveillance through fine-grained analysis of human motion.

    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: MoSE3 is a per-pixel $\mathrm{SE}(3)$ motion estimator, not a generative model, a language model, or a model trained on scraped data. Art-Kubric is fully synthetic, rendered from 3D assets and environment maps, and contains no human imagery or scraped Internet content. Neither poses the risks this question addresses.

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 the original sources of the external assets we use. These include the π^3 checkpoint (Appendix A.1), the training datasets (Appendix A.5), the evaluation benchmarks and baselines (Sec. 5), the 3D assets and environment maps that Art-Kubric renders, listed with their licenses in Appendix D, and the sources of the in-the-wild videos (Appendix C.4). We use each asset under its stated terms. PartNet-Mobility permits only non-commercial research and educational use, which is why the Art-Kubric data is released under CC BY-NC 4.0.

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: Art-Kubric is documented in Sec. 4 and Appendix D, which cover its generation settings, per-scene randomization, exported annotations, statistics, and license. The data is released under CC BY-NC 4.0. The generation pipeline redistributes no upstream assets, so users who re-render scenes obtain each asset under its own license.

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: The paper does not involve crowdsourcing or research with human subjects. Training uses synthetic data, evaluation uses public benchmarks released by their original authors, and qualitative results use publicly available videos.

    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: The paper does not involve human-subjects research, so IRB approval is not applicable.

    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: LLMs are not used as an important, original, or non-standard component of the core method. Any LLM usage in this work is limited to writing, editing, and formatting, which the policy states does not require declaration.

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.