3DPoV: Improving 3D understanding via Patch Ordering on Videos

Ioana Simion [*1,2] **Mohammadreza Salehi** [*3] **Shashanka Venkataramanan** [4] **Cees G. M. Snoek** [3] **Yuki M Asano** [5]

Abstract

Visual foundation models have achieved remarkable progress in scale and versatility, yet understanding the 3D world remains a fundamental challenge. While 2D images contain cues about 3D structure that humans readily interpret, deep models often fail to exploit them, underperforming on tasks such as multiview semantic consistency–crucial for applications including robotics and autonomous driving. We propose a self-supervised approach to enhance the 3D understanding of vision foundation models by (i) introducing a temporal nearest-neighbor consistency loss that finds corresponding points across video frames and enforces consistency between their nearest neighbors, (ii) incorporating reference-guided ordering that requires patch-level features to be not only expressive but also consistently aligned, and (iii) constructing a mixture of video datasets tailored to these objectives, thereby leveraging rich 3D information. Our method, 3DPoV, achieves state-of-the-art performance in keypoint matching under viewpoint variation, as well as in depth and surface normal estimation, and consistently improves a diverse set of backbones, including DINOv3. Code is available at https://github.com/Ioana-Simion/3DPoV.

1. Introduction

Recent advances in dense self-supervised learning have yielded feature representations that are remarkably effective for a variety of vision tasks, including object part recognition, dense retrieval, and semantic matching. Models like DINO (Caron et al., 2021) and its successors demonstrate that fine-grained correspondence can emerge even without explicit labels. However, a critical shortcoming remains: robustness to viewpoint change. When the camera pose shifts, these representations often degrade substantially, revealing a lack of true 3D spatial understanding.

This challenge is especially important in real-world scenarios where objects are seen from multiple perspectives, and consistent recognition across views is crucial. Existing self-supervised approaches based on static images or temporally adjacent frames–while effective in learning texture and semantics–struggle to capture geometric cues like depth, structure, or object permanence under motion. This gap has been increasingly highlighted by benchmarks like Probe3D (El Banani et al., 2024), which systematically exposes these limitations across keypoint matching, depth prediction, and surface normal estimation tasks.

To address this, we propose **3DPoV (3D understanding via Patch Ordering on Videos)**, a post-training strategy for enhancing multiview spatial consistency by enforcing temporal alignment across tracked patches. Our method builds on the insight that viewpoint changes induce systematic deformations in patch-level similarity patterns. By supervising the relative ranking of features extracted along point tracks over time, 3DPoV encourages the network to learn descriptors that remain consistent across large temporal and viewpoint shifts.

Unlike prior approaches such as TimeTuning (Salehi et al., 2023) and MoSiC (Salehi et al., 2024), which operate through temporal propagation of segmentation maps, or NeCo (Pariza et al., 2025), which focuses on intra-image part ordering, our framework directly aligns patch-wise relationships across frames. It leverages differentiable sorting (Petersen et al., 2022) to compare similarity structures over reference patches, and uses a teacher-student setup grounded in explicit temporal tracking to provide stable supervision under motion and occlusion.

By leveraging video sequences and lightweight fine-tuning, 3DPoV instills emergent 3D reasoning, with consistent gains across all Probe3D tasks–particularly under large viewpoint changes, occlusion, and lighting variation. Our approach narrows the gap between 2D feature learning and 3D understanding, offering an efficient and scalable path to enhance foundation models for geometry-aware visual rea-

[*]Equal contribution [1]Quantitative Healthcare Analysis (qurAI) group, University of Amsterdam [2]Amsterdam University Medical Center, department of Biomedical Engineering and Physics [3]Video & Image Sense Lab, University of Amsterdam [4]Valeo.ai [5]Fundamental AI Lab, University of Technology Nuremberg. Correspondence to: Ioana Simion <i.simion@uva.nl>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

soning, requiring only 20 hours of training on A100 GPUs (detailed in Section D). The main contributions of 3DPoV are as follows:

- We introduce a temporal permutation loss anchored by point tracks, which supervise the relative ordering of patch features across frames. This directly trains the model to produce viewpoint-invariant descriptors without relying on crops or masks.
- We propose a more stable teacher-student setup by also passing reference frames through the student network yielding features that are both discriminative and sortable under motion and occlusion; stability is further ensured through a reference pool mixing external frames with internal samples from the same video.
- We demonstrate that 3DPoV achieves consistent improvements across all Probe3D difficulty regimes with lightweight training. Unlike prior approaches that trade robustness at large viewpoint shifts for small-viewpoint gains, our method improves uniformly across viewpoint variation, occlusion, and lighting changes.

2. Background

Self-supervised learning on videos has leveraged temporal coherence to improve semantic consistency, but often without explicitly modeling spatial alignment. TimeTuning (Salehi et al., 2023) propagates cluster assignments across frames to stabilize semantics, while MoSiC (Salehi et al., 2024) strengthens this with point tracks for improved consistency. However, both methods remain centered on propagating semantic groupings rather than directly optimizing for viewpoint-robust spatial understanding.

Spatially-aware ordering methods such as NeCo (Pariza et al., 2025) address viewpoint sensitivity in images by supervising the relative ordering of patch similarities via differentiable sorting. This approach enhances local spatial structure and yields more context-aware representations, making it particularly relevant to our work. However, NeCo is restricted to static images and overlapping crops, which assume a fixed viewpoint and discard the global context that intrinsically encodes spatial structure. These assumptions limit its applicability to videos, where motion and viewpoint changes dominate.

Evaluation frameworks such as Probe3D (El Banani et al., 2024) expose these gaps by probing robustness under viewpoint changes across tasks like keypoint matching, depth estimation, and surface normal prediction. Existing models tend to perform well under small viewpoint differences but suffer a sharp drop in accuracy as the viewpoint gap increases, highlighting the need for methods that improve consistently across all regimes. Leveraging multiview supervision has recently emerged as a promising direction for enhancing 3D correspondence (You et al., 2024; Ruan et al., 2024). More recently, models such as DINOv3 (Siméoni et al., 2025) have explicitly targeted these evaluations, reporting strong results and emphasizing the growing role of geometry-aware benchmarks in guiding self-supervised learning. In addition, Probe3D combines quantitative metrics with qualitative inspection, offering a diagnostic lens into whether models truly encode intrinsic 3D structure rather than relying on priors, appearance, or texture cues. The systematic gaps highlighted by Probe3D motivate our approach, which is designed to improve spatial consistency across viewpoint variation.

3. Method

We propose **3DPoV**, a framework for learning temporally consistent dense features from videos by leveraging point tracks and patch-level ordering. The method builds on a teacher–student architecture, where the student processes video frames independently and the teacher provides a stable anchor frame for supervision (Figure 1). To enforce temporal consistency, we track a grid of points across frames and extract features at aligned locations.

Rather than matching features directly, we align the *relative similarity structure* of tracked patches over time. For each frame, we compute similarity rankings with respect to a shared set of reference features and use differentiable sorting to obtain soft permutation matrices. The student is then trained to match the teacher's anchor-frame permutations, encouraging viewpoint-invariant descriptors that remain consistent under motion, occlusion, and appearance changes.

Preliminaries Given a video clip $X \in \mathbb{R}^{h \times w \times 3 \times t}$, where $h \times w$ is the spatial resolution and t the number of frames, we extract dense patch-level features using a Vision Transformer (ViT) (Dosovitskiy et al., 2021) backbone. Each frame is encoded independently by a student network Ψ_S, while the teacher network Ψ_T encodes a designated anchor frame that serves as the temporal reference for supervision and is updated using the exponential moving average (EMA) of the student's parameters.

Establishing spatiotemporal correspondence To align features across time, we require reliable pixel-level correspondences between frames. Rather than relying on optical flow or feature matching, which can be noisy in the presence of occlusion or ambiguous camera motion (Salehi et al., 2024), we employ an off-the-shelf point tracker that provides stable trajectories and visibility estimates (i.e., where the tracked point disappears from the frame). Such tracks allow the model to observe how the same object evolves under motion and viewpoint change, serving as an essential

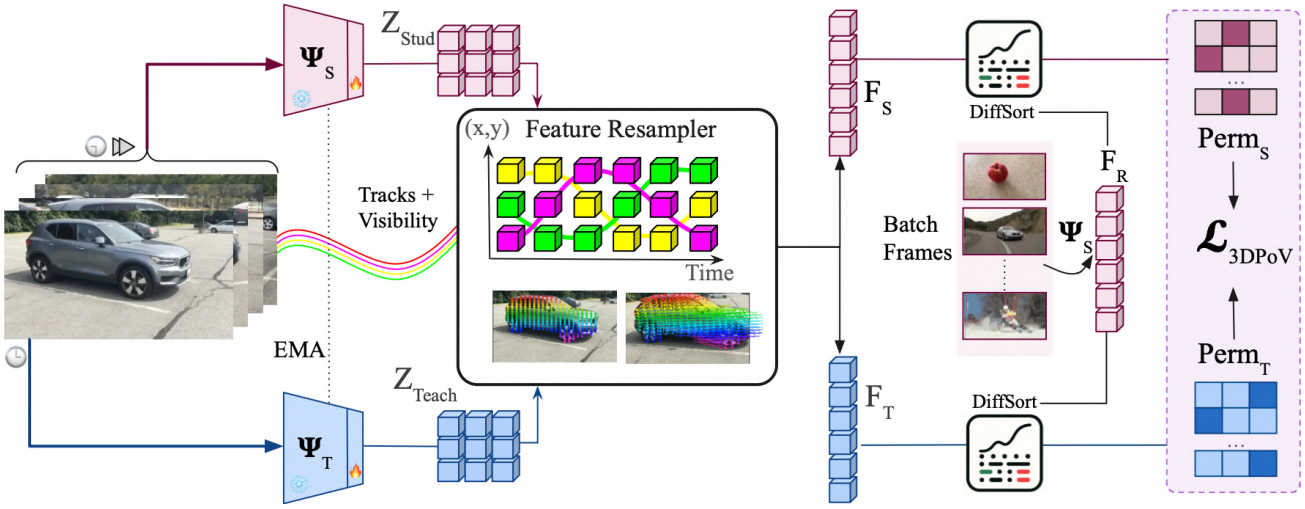

Figure 1. **3DPoV**: Learning 3D-aware representations via Patch Ordering in Videos. We begin by extracting motion trajectories $\mathrm{T}_{f,i}$ from raw video clips using CoTrackerV3. The video is parsed into frames, and each is processed by the student and teacher networks to produce feature maps $Z^{\text{stu}}, Z^{\text{teach}} \in \mathbb{R}^{p\times d\times t}$. Using the tracked coordinates from $\mathrm{T}_{f,i}$, we resample features to obtain patch sequences F_s (student) and F_t (teacher). Reference features F_r are extracted from other batch frames using the student network. Pairwise cosine distances $D_{i,j}$ are computed between F_s and F_r, and between F_t and F_r. These distances are sorted via a Differentiable Sorting module, producing permutation matrices Perm $\in \mathbb{R}^{np_q\times np_r\times np_r}$ that enforce consistent patch ordering across time.

cue for enforcing temporal consistency in feature space.

We use CoTrackerv3 (Karaev et al., 2024), an off-the-shelf point tracker, to estimate trajectories $T_{t,i}$ and visibility masks $V_{t,i}$ for a set of points initialized on an anchor frame. Here, t denotes the frame index and i denotes the point index on the initialized grid. Concretely, we initialize a regular grid of size $g \times g$ on the first frame, resulting in $N = g^2$ points with coordinates $\{(x_i, y_i)\}_{i=1}^{N}$. Given the video clip X and this grid, the tracker predicts the trajectories of all N points across the sequence as:

$$\mathrm{T}_{t,i} := \texttt{Tracker}(X, (x_i, y_i)) \in \mathbb{R}^{t\times N\times 2}, \qquad (1)$$

We initialize the tracker at the first frame $t = 0$, where all points are guaranteed to be visible. This frame is designated as the anchor, and its features are extracted using the teacher network, which provides a stable reference during training. Subsequent frames, encoded by the student, may contain occlusions, motion blur, or appearance changes. Aligning their representations with the clean anchor frame encourages the learning of features that remain consistent across viewpoint and occlusion variations.
Since tracking is initialized on the first frame, all points are guaranteed to be visible at $t = 0$. We therefore designate frame 0 as the anchor and extract its features with the teacher network, which provides a stable reference throughout training. Later frames, processed by the student, may contain occlusions or appearance changes; aligning them with the clean anchor frame encourages viewpoint- and occlusion-invariant representations.

Figure 2. CoTrackerV3 maintains high tracking accuracy across lighting changes, viewpoint shifts, and forward camera motion in a CO3D sample.

Feature Extraction and Alignment Features $Z^{\text{stu}}, Z^{\text{teach}} \in \mathbb{R}^{p\times d\times t}$ are extracted from raw frames using the student Ψ_S and teacher Ψ_T networks, where p denotes the number of patches, d is the feature dimension, and t the number of frames per video. While NeCo (Pariza et al., 2025) leverages ROI align to extract overlapping patches between paired crops of a single image, our approach instead samples full-frame patches and uses point tracks to extract aligned patch trajectories throughout time. Given a tracked trajectory $\mathrm{T}_{f,i}$ for the i^{th} point, we sample back corresponding patch features from Z^{stu} and Z^{teach} to obtain temporally aligned patch sequences F_s and F_t.

To balance generalization and alignment quality, we consider two strategies for retrieving patch features from tracked coordinates. In resized sampling, feature maps are upsampled to the input resolution and features are retrieved via nearest-neighbor indexing. In latent-space sampling,

features are extracted directly from the native feature grid using bilinear interpolation.

Similarity via Differentiable Sorting To supervise the temporal consistency of patch features, we adopt a differentiable sorting mechanism that aligns the relative similarity structure of patches over time. Rather than enforcing direct feature similarity between frames, we compare the *ranking distributions* of each patch with respect to a shared set of reference features. This encourages the model to learn a structured, viewpoint-invariant representation of similarity– crucial for robust dense correspondence.

For each video clip, we construct a reference feature bank by sampling N_{ref} number of local crops from frames of other videos in the batch, as well as from a non-anchor frame $k \neq 0$ of the current video. These reference crops preserve spatial layout and introduce both intra-video and inter-video diversity. Instead of cropping raw input images, we apply spatial cropping in feature space after forwarding the reference frames through the student network Ψ_S. This yields a reference feature bank of patch features $F_r \in \mathbb{R}^{B\times np_r\times d}$, where np_r is the number of reference patches per sample and d is the feature dimension.

As such, the Differentiable Sorting module operates on a per-sample basis. It receives a set of query patch features $F_q \in \mathbb{R}^{B\times np_q\times d}$, extracted from either the student at a future frame $t>0$ or the teacher at the anchor frame $t=0$, where np_q denotes the number of query patches. It also receives a set of reference features $F_r \in \mathbb{R}^{B\times np_r\times d}$ obtained from the reference bank.

To compare the query features with the reference features, we compute cosine similarity:

$$S_{i,j} = \frac{\langle F_q^i, F_r^j\rangle}{\|F_q^i\|\cdot\|F_r^j\|}, \quad D_{i,j} = 1 - S_{i,j} \tag{2}$$

for $i \in [1, np_q]$, $j \in [1, np_r]$. Each row of S encodes the similarity between one query patch and all reference patches. Since our goal is to capture relative ordering rather than absolute scores, we pass the distance matrix $D = 1 - S$ to the differentiable sorting module (Petersen et al., 2022) which outputs soft permutation matrices P that approximate the ranking distribution of each query patch over the reference set. Full details of the sorting procedure are provided in Appendix A.

Patch-Wise Permutation Loss for Temporal Alignment To supervise the temporal consistency of patch-level features, we compare the sorting behavior of the student network across time to that of the teacher network at a fixed anchor frame. Rather than enforcing direct similarity in feature space, we align their respective soft permutation matrices over a shared set of reference patches. This encourages the student to match the teacher's viewpoint-invariant similarity structure, even under occlusions and appearance shifts.

Let $F_t^S \in \mathbb{R}^{B\times np_q\times d}$ denote student features at a future frame $t>0$, and F_0^T the teacher features at the anchor frame $t=0$. For each of the N_{ref} reference crops $F_r^R \in \mathbb{R}^{B\times np_r\times d}$, we compute soft permutation matrices via differentiable sorting:

$$P_{t,r}^S = \text{DiffSort}(F_t^S, F_r^R),\ P_{0,r}^T = \text{DiffSort}(F_0^T, F_r^R) \tag{3}$$

Each soft permutation matrix $P \in \mathbb{R}^{B\times np_q\times np_r\times np_r}$ encodes, for every query patch, a distribution over the ranked positions of reference patches.

Patch-wise Cross-Entropy Loss We supervise the student permutation matrix $P_{t,r}^S$ with respect to the teacher matrix $P_{0,r}^T$. For each query patch i, we compute the cross-entropy where the student distribution provides the weighting:

$$\mathcal{L}_{\text{CE}}^i = -\sum_{j=1}^{np_r} P_{0,r}^S[i,j]\cdot\log\left(P_{t,r}^T[i,j]+\epsilon\right) \tag{4}$$

This formulation encourages the student to place probability mass in regions where the teacher also provides support, while simultaneously promoting disentangled and confident predictions. In practice, this leads to sharper spatial rankings and improves patch-level discrimination. We then average this loss across all query patches i and samples b in the batch:

$$\mathcal{L}_{\text{CE}}^{(t,r)} = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{np_q}\sum_{i=1}^{np_q}\mathcal{L}_{\text{CE}}^{(b,i)} \tag{5}$$

Visibility-Weighted Loss To account for occlusion and tracking failures, we weight each patch by its visibility at both the anchor and current frames. Let $v_{0,t}^{(b,i)} = V_{0,i}^{(b)}\cdot V_{t,i}^{(b)}$ denote the joint visibility of patch i in sample b.

The visibility-weighted cross-entropy becomes:

$$\mathcal{L}_{\text{CE}}^{(t,r)} = \frac{1}{B}\sum_{b=1}^{B}\sum_{i=1}^{np_q}\frac{v_{0,t}^{(b,i)}}{\sum_j v_{0,t}^{(b,j)}+\epsilon}\cdot\mathcal{L}_{\text{CE}}^{(b,i)} \tag{6}$$

Final Loss Across Time and References To enforce alignment throughout the sequence, we apply the patch-wise loss across all compared frames $t-1$ and all references $r=1,\ldots,N_{\text{ref}}$. The final permutation alignment loss is:

$$\mathcal{L}_{\text{3DPoV}} = \frac{1}{t-1}\sum_{t=t_{\text{start}}}^{T}\frac{1}{N_{\text{ref}}}\sum_{r=1}^{N_{\text{ref}}}\mathcal{L}_{\text{CE}}^{(t,r)} \tag{7}$$

This objective encourages the student network to produce temporally aligned, viewpoint-consistent patch-level rankings relative to shared reference crops (Figure 7)–anchored by the teacher signal–while softening the contribution of low-confidence or occluded regions via visibility weighting. Further details on loss formulation and design decisions are provided in Appendix B.

An equally important factor is the choice of fine-tuning data, which plays a central role in shaping the model's ability to learn viewpoint-invariant and geometry-aware representations from videos. To capture complementary aspects of variability, we fine-tune on a blend of three datasets: (i) CO3D (Reizenstein et al., 2021), which provides long object-centric multiview sequences with large viewpoint shifts; (ii) DL3DV (Ling et al., 2024), which offers diverse dynamic scenes and spatial layouts; and (iii) YouTube-VOS (Xu et al., 2018), which introduces unconstrained motion, occlusions, and real-world camera trajectories. Together, this mixture spans single-object, scene-level, and natural video variability, supporting robust learning of dense, temporally consistent features. Full dataset descriptions and preprocessing details are provided in Appendix C.

4. Experiments

We evaluate our method on the Probe3D benchmark (El Banani et al., 2024), which assesses 3D spatial understanding through keypoint matching, depth estimation, and surface normal estimation. We fine-tune the last two Transformer blocks (blocks 10 and 11) while keeping the rest of the network frozen. Unless otherwise stated, we use the DINOv2-R backbone. All comparisons are made against models with identical backbone architectures, isolating the effect of our method.

Although we focus on DINO-based baselines for fair comparison, 3DPoV is a backbone-agnostic post-training framework and can be applied to any model that produces dense visual features. To ensure fair placement of results, we reproduce all DINO baselines from (El Banani et al., 2024) and use publicly available checkpoints for prior post-training baselines (TimeTuning, NeCo, MoSiC). Dataset details and reproduction studies are provided in Appendix D and Appendix H.

Keypoint Matching We evaluate on SPair-71k (2D human-annotated keypoints) and Navi (synthetic data with 3D geometry and calibrated cameras). On SPair, recall is measured by predicting target keypoints from feature similarity. Results are reported across viewpoint bins (small, medium, large) as well as the "All" split, which aggregates all pairs but is biased toward small-viewpoint cases. In contrast, Navi enables 3D-aware evaluation by matching correspondences directly in 3D, assessing both by Euclidean error in 3D space and reprojection error in 2D. We report recall at multiple thresholds and analyze results as a function of relative camera rotation. Full experimental details are deferred to Appendix D.

Table 1 reports SPair-71k recall across increaasing viewpoint differences (small, medium, large), as well as an aggregated 'All' split. 3DPoV achieves the strongest overall performance across all viewpoint bins, maintaining gains even under larger viewpoint shifts. This consistent behaviour indicates stronger spatial consistency under diverse transformations. NeCo improves over its baseline primarly for small viewpoint differences, but its performance degrades sharply as viewpoint variation increases, suggesting limited robustness to large geometric changes. Segmentation-focused approaches such as TimeTuning and MoSiC achieve temporal semantic propagation but fail to retain the spatial discrimination required for robust keypoint matching. This highlights that improvements in semantic consistency over time do not directly translate into stronger spatial semantic correspondence.

Table 1. SPair-71k viewpoint difference. 0: No significant view difference (same view or minimal changes), 1: Moderate viewpoint difference, 2: Large viewpoint difference. DINOv2R: DINOv2 with registers. 3DPoV improves on the baseline, including consistent improvements on challenging viewpoint settings

Model	Backbone	S / 0	M / 1	L /2	All
DINO	ViT-S/16	**28.34**	**23.38**	**24.44**	**25.63**
TimeTuning	DINOv1-S/16	26.76	22.48	23.45	23.96
MoSiC	DINOv1-S/16	26.73	21.97	22.98	23.76
DINO	ViT-B/16	30.19	24.22	24.35	26.39
NeCo	DINOv1-B/16	30.24	24.45	23.10	26.32
3DPoV	DINOv1-B/16	**31.77**	**25.74**	**25.80**	**28.16**
DINOv2R	ViT-B/14	58.20	51.56	53.41	53.47
NeCo	DINOv2R-B/14	59.57	49.06	52.35	54.42
MoSiC	DINOv2R-B/14	56.37	50.70	51.75	51.72
3DPoV	DINOv2R-B/14	**60.16**	**52.79**	**54.50**	**55.40**
DINOv3	ViT-B/16	61.99	**48.67**	46.77	55.76
3DPoV	DINOv3-B/16	**62.24**	48.56	**46.81**	**55.84**

A similar pattern is observed on Navi (Table 2). While NeCo shows gains on SPair, its improvements do not transfer as effectively, reflecting the added difficulty of enforcing 3D-consistent correspondences. In contrast, MoSiC achieves better results in the θ_{60}^{180} range of the DinoV2-reg variant, but all other viewpoints are degraded. 3DPoV, on the other hand, consistently improves across all relative viewpoint bins, underscoring its robustness under large viewpoint changes.

Finally, the breakdown in Table 13a and Table 13b shows that 3DPoV achieves consistent gains in both 3D correspondence accuracy and 2D reprojection alignment across all thresholds. This dual improvement highlights that the learned features are geometrically faithful in 3D space while also preserving accurate alignment in 2D.

Table 2. Navi Performance Comparison Across Models with performance binned for different relative viewpoint changes between image pairs. Best results are in bold. DINOv2R: DINOv2 with registers

Model	Backbone	θ_0^{15}	θ_{15}^{30}	θ_{30}^{60}	θ_{60}^{120}
DINO	ViT-S/16	**84.36**	**55.17**	**34.58**	**20.48**
TimeTuning	DINOv1-S/16	80.81	52.61	33.93	19.96
MoSiC	DINOv1-S/16	80.21	52.07	33.37	19.59
DINO	ViT-B/16	86.13	56.92	33.37	19.74
NeCo	DINOv1-B/16	84.94	53.52	31.80	18.47
3DPoV	DINOv1-B/16	**86.42**	**57.18**	**33.77**	**20.42**
DINOv2R	ViT-B/14	87.87	67.45	47.15	31.58
NeCo	DINOv2R-B/14	88.76	65.10	43.88	28.96
MoSiC	DINOv2R-B/14	87.13	66.46	46.95	**31.78**
3DPoV	DINOv2R-B/14	**89.18**	**68.98**	**47.64**	**31.63**
DINOv3	ViT-B/16	94.40	74.73	48.64	**31.45**
3DPoV	DINOv3-B/16	**94.47**	**74.74**	**48.65**	31.36

Depth and Surface normal estimation We evaluate our model's geometric understanding using depth and surface normal estimation on the Navi benchmark, following the standardized protocol introduced in Probe3D. This evaluation tests whether the learned features encode meaningful 3D spatial geometry beyond keypoint-level correspondences.

Since the backbone models do not inherently predict depth or surface normals, we follow the Probe3D protocol on training lightweight linear probes on top of frozen features for each task. This setup isolates the quality of the learned representations, ensuring that performance reflects spatial awareness embedded in the features rather than downstream training capacity.

In line with (El Banani et al., 2024), we conduct both quantitative evaluation using ground-truth 3D signals and qualitative inspection to better interpret the spatial reasoning captured by the features. Full definitions of the evaluation metrics are deferred to Appendix D.

Qualitative depth results (Figure 3) show that 3DPoV produces more coherent maps than the baseline, preserving boundaries and geometric detail across diverse object types. For instance, in the dinosaur example, our method resolves the lower leg despite heavy shadow, and on the tractor it avoids interpreting a painted stroke as spurious geometry, yielding a more plausible depth map. These improvements align with the quantitative gains in Table 3.

For surface normals (Figure 4), we visualize both predictions and angular error maps to highlight regions of divergence between 3DPoV and the baseline. 3DPoV provides more faithful orientation estimates, particularly under challenging conditions: on the eagle, it better recovers fine structure along the body and wing edges, and in the can example it reduces errors caused by reflective surfaces. These

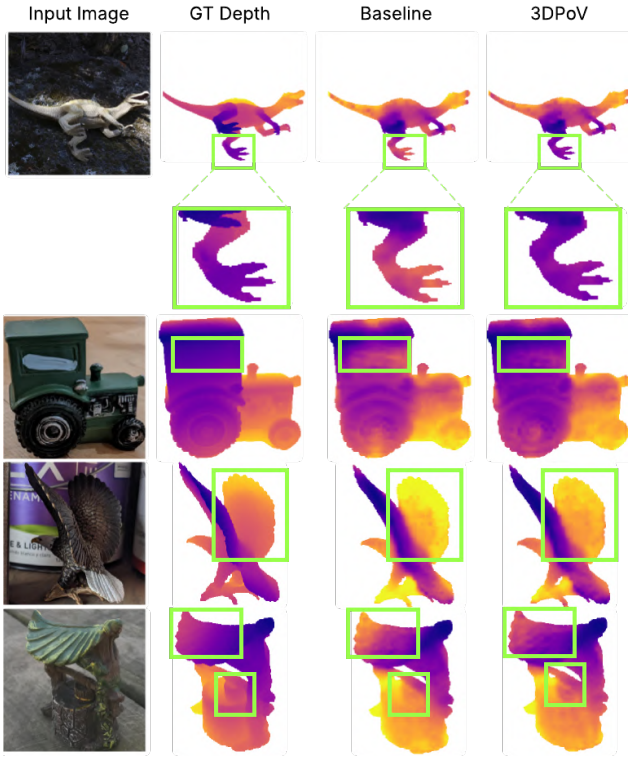

Figure 3. **Depth Qualitative Examples.** Comparing predicted depth maps from Baseline (DINOv2-reg) and **3DPoV**. Ground truth (GT) depth is provided for reference

qualitative trends are consistent with the quantitative improvements reported in Table 4. Additional visualizations, including relative error maps with respect to ground truth, are provided in Appendix L.

Moreover, we probe the proposed framework on downstream tasks outside of probe3D scope and show consistent improvements on video understanding in Table 12.

Light and occlusions robustness

Robustness to lighting conditions emerged as an unexpected finding in our qualitative analysis, while robustness to occlusions is an expected property through our model design. We therefore designed two experiments to quantitatively validate both. First, we crafted two NAVI subsets: one featuring more challenging lighting conditions, and another with normal or optimal lighting (Figure 5). We re-evaluated the model on both subsets and report the scores in Table 5. Results show not only that 3DPoV improves over baseline in both lighting scenarios, but also that score improvements are generally larger in intense lighting conditions. A more detailed breakdown of subset configuration and examples are available in Appendix K.

For the occlusion experiment, we do not use NAVI subsets, as it contains too few occluded samples and its ground-truth masks do not exclude these regions. Instead, we introduce

Table 3. Depth estimation results on Navi. Accuracy is reported using the threshold-based metrics δ_1 (< 1.25), δ_2 ($< 1.25^2$), and δ_3 ($< 1.25^3$), as introduced by (Eigen et al., 2014). We also report RMSE in meters. Both scale-aware and scale-invariant scores are shown for completeness.

Model	Backbone	Scale-Aware				Scale-Invariant			
		$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	RMSE $\downarrow$	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	RMSE $\downarrow$
DINO	ViT-B/16	47.16	73.83	86.70	0.1237	58.64	81.85	90.43	0.1022
3DPoV	DINOv1-B/16	**47.93**	**74.77**	**87.45**	**0.1218**	**58.83**	**82.20**	**90.67**	**0.1014**
DINOv2R	ViT-B/14	57.62	82.49	91.97	0.0960	68.49	87.89	93.95	0.0778
3DPoV	DINOv2R-B/14	**59.17**	**83.61**	**92.59**	**0.0933**	**69.61**	**88.47**	**94.19**	**0.0757**

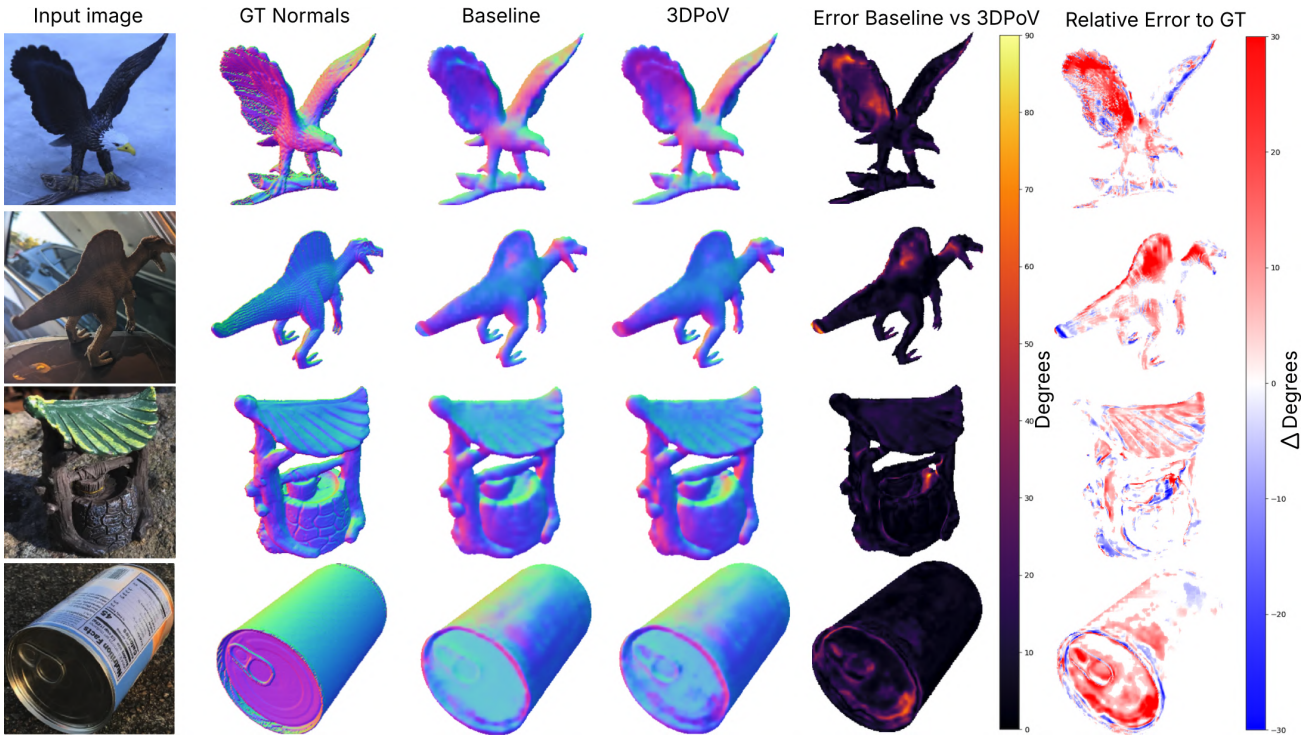

Figure 4. **Surface Normal Qualitative Examples.** To highlight differences between models with shared architecture, we visualize the angular error between the Baseline and 3DPoV predictions, which highlights regions where surface normal estimates differ. ΔError to GT denotes the difference in angular error between baseline and 3DPoV predictions with respect to the ground truth normals, shown only in regions of disagreement (error $> 5°$).Red areas indicate where 3DPoV predictions align more closely with the ground truth, while blue areas indicate where the baseline is closer. For baseline we use DINOv2 with Registers.

Table 4. Surface normal estimation results on Navi. We report accuracy at angular thresholds as well as the RMSE in degrees between predicted and GT normals.

Model	Backbone	11.25° ↑	22.5° ↑	30° ↑	RMSE ↓
DINO	ViT-B/16	31.47	58.61	70.62	31.83
3DPoV	DINOv1-B/16	**31.67**	**58.82**	**70.71**	**31.78**
DINOv2R	ViT-B14	37.10	65.93	77.09	28.07
3DPoV	DINOv2R-B/14	**38.29**	**67.00**	**77.86**	**27.78**

synthetic occlusions by overlaying rectangular masks of varying size. To ensure fair comparison, occlusions are generated deterministically and fixed across runs. We evaluate on SPair and report recall only over non-occluded keypoints (as explained in Appendix K).

Table 5. Comparison of Navi kepoint matching recall across viewpoint bins. Higher is better. Evaluation on light conditions subsets; 3DPoV showcases higher improvements under challenging light conditions.

Model	Backbone	θ_0^{15}	θ_{15}^{30}	θ_{30}^{60}	θ_{60}^{120}
Default NAVI dataset					
DINOv2R	ViT-B/14	87.87	67.45	47.15	31.58
3DPoV	DINOv2R-B/14	**89.18**	**68.98**	**47.64**	**31.63**
Less lighting variation subset					
DINOv2R	ViT-B/14	92.28	67.72	49.24	**33.69**
3DPoV	DINOv2R-B/14	**93.27**	**68.85**	**49.27**	33.52
Intense lighting conditions subset					
DINOv2R	ViT-B/14	89.63	66.98	48.13	33.96
3DPoV	DINOv2R-B/14	**90.91**	**68.45**	**48.82**	**34.11**

Figure 5. Samples form the two light conditions subsets. Entries were selected manually based on visual artifacts such as: shadows, reflections, color variation due to lighting

Table 6. SPair-71k Keypoint Matching; Robustness to occlusions. 3DPoV consistently improves across all difficulties levels on both levels of occlusion.

MODEL	S / 0	M / 1	L /2	ALL
One view occluded				
DINOv2R	59.78	53.84	56.01	55.54
3DPoV	**62.04**	**54.92**	**57.65**	**57.54**
Both views occluded				
DINOv2R	58.84	53.41	56.90	54.88
3DPoV	**61.30**	**54.11**	**58.09**	**56.98**

3DPoV consistently improves over the baseline across all viewpoint bins (Table 6), both when only the target view is occluded (Figure 6) and when both views are occluded. This trend remains stable across all difficulties, indicating robustness under increasing occlusion severity and consistent generalization even when both views are degraded.

5. Ablation Studies

We conduct ablation studies to isolate the impact of key design choices in **3DPoV**. All experiments are based on the DINOv2-Reg backbone. To ensure fair comparisons, we vary only one factor per experiment and report performance at matched training durations.

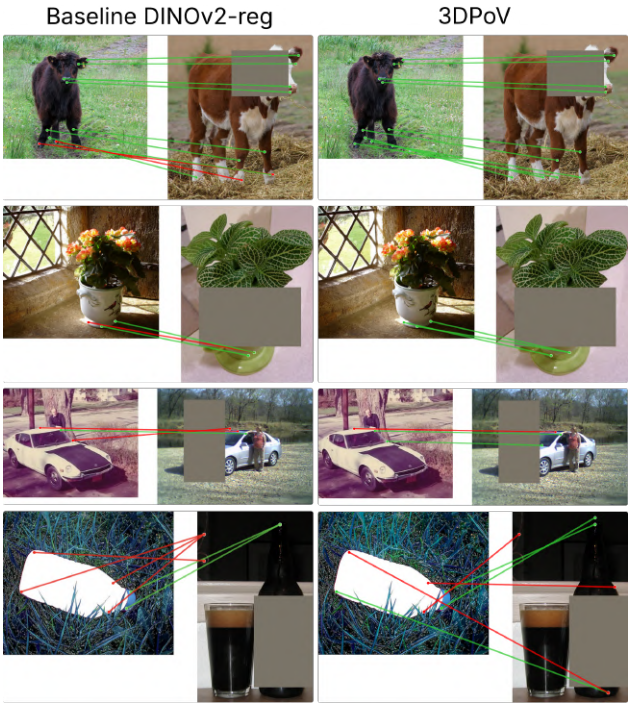

Figure 6. Correspondence quality with one view occluded. 3DPoV manages to match keypoints at different levels of occlusion. As seen in the last row, even if the entire body of the bottle is occluded, 3DPoV still locates the bottom of the object.

Reference extraction. As shown in Table 7a, student-extracted references improve average performance by +2.37%, with the largest gains on small viewpoint differences. This supports our design choice of letting the student process references, as it encourages more discriminative and sortable features.

Number of frames. Table 7b compares training with 1, 2, and 4 frames. Since our flow does not intrinsically operate on a single frame, the 1-frame setup leverages NeCo-style overlapping crops as a proxy. Performance improves steadily with more frames: 4 frames bring +1.01% over 1 frame and +0.13% over 2 frames, indicating that temporal supervision benefits from richer context across all viewpoint regimes.

The chosen setup currently uses 4 frames with step size 4, covering 12 frames, which we found enough to cover dynamics and viewpoint changes. Because our source datasets leverage rotations around an object/scene, more frames will not provide that much dynamic/visual information, and might actually lead to less informative correspondences, as revelant points become invisible (not weighted) due to rotation. The setup allows for using more frames, but due to compute limitations, this was the best configuration we could explore.

Table 7. **Ablation of Key Design Choices in 3DPoV.** We report Keypoint Matching Recall on SPair-71k across viewpoint difficulty levels–Small, Medium, Large, and All.

(a) References Extracted by

MODEL	S / 0	M / 1	L /2	ALL
Teacher	57.95	51.04	53.05	53.03
Student	**60.16**	**52.79**	**54.50**	**55.40**

(b) Number of frames

FRAMES	S / 0	M / 1	L /2	ALL
1	59.26	52.02	54.00	54.39
2	60.04	52.72	54.35	55.27
4	**60.16**	**52.79**	**54.50**	**55.40**

(c) Dataset choice

DATA	S / 0	M / 1	L /2	ALL
CO3D	59.32	52.29	54.08	54.58
YTVoS	59.71	52.56	54.34	55.02
DL3DV	59.84	52.41	54.24	55.02
YT-DL	60.02	52.66	54.24	55.25
CO3-YT-DL	**60.16**	**52.79**	**54.50**	**55.40**

(d) Step size on frame sampling

STEP	S / 0	M / 1	L /2	ALL
2	60.04	52.75	54.50	55.22
4	**60.16**	**52.79**	**54.50**	**55.40**
6	59.70	52.28	54.06	55.02

(e) Type of Resampling

METHOD	S / 0	M / 1	L /2	ALL
Resized	59.83	52.42	54.20	55.04
Latent	**60.16**	**52.79**	**54.50**	**55.40**

(f) Choice of Point Tracker

TRACKER	S / 0	M / 1	L /2	ALL
RAFT	59.81	52.43	54.31	55.07
CoTrackerV3	**60.16**	**52.79**	**54.50**	**55.40**

Dataset choice. Table 7c shows that the CO3-YT-DL mixture outperforms any single dataset, confirming that diversity is key to robustness. While CO3D alone yields the weakest overall scores, adding it to YT–DL still improves the large-viewpoint bin (+0.26), highlighting that object-centric multiview footage provides complementary signal.

Step size. Varying the temporal step between frames (Table 7d) shows that step 2 captures too little variation, while step 6 reduces visibility in multi-view datasets like CO3D and DL3DV, biasing tracks towards uninformative regions (sky/ground). Step 4 achieves the best trade-off, maintaining many visible points while capturing meaningful viewpoint changes.

Resampling strategy. Latent-space interpolation (Table 7e) outperforms resized sampling (+0.36 overall), suggesting that operating directly in the feature grid avoids artifacts from upsampling and preserves finer spatial detail.

Point tracker. Table 7f compares RAFT (Teed & Deng, 2020) and CoTrackerV3 (Karaev et al., 2024). Our method improves with both, showing independence from tracker choice, but CoTrackerV3 performs best (+0.33 overall), likely due to its robustness to occlusions and sudden motion compared to optical flow methods.

Choice of anchor frame Our method assigns the Teacher to the first frame ($t = 0$) and the Student to subsequent frames. We tested the reverse configuration - Table 8. The results confirm our design choice (55.40% vs 55.05%), where CoTracker initializes points at $t = 0$, guaranteeing they are visible and unoccluded. Assigning the Teacher to $t = 0$ ensures the target features are reliable. Using later frames as the anchor introduces occlusion noise into the supervision signal.

Table 8. SPair-71k Keypoint Matching; we ablate the choice of frame being processed by the teacher and consequently used as anchor.

TEACHER FRAME	S / 0	M / 1	L /2	ALL
Last (t-1)	59.85	52.40	54.28	55.05
First (0)	**60.16**	**52.79**	**54.50**	**55.40**

6. Conclusion

In this paper, we introduced 3DPoV, a framework for learning dense, viewpoint-invariant features through temporally anchored permutation supervision. By integrating point tracks with reference-based sorting, our method enforces relative similarity structures that remain stable across time, occlusion, and viewpoint variation. Evaluations across Probe3D tasks demonstrate consistent improvements over all baselines, with balanced gains across both small and large viewpoint shifts and emerging robustness to challenging lighting conditions. These results highlight the value of temporal ranking as a supervisory signal and suggest that point tracking can serve as a powerful tool for geometry-aware representation learning without requiring explicit 3D labels.

Acknowledgements

We thank the Amsterdam ELLIS Unit for their generous funding, which allowed the visits to the FunAI laboratory in Nuremberg. The presentation of this paper at the conference was financially supported by the Amsterdam ELLIS Unit and Qualcomm.

Impact Statement

This paper presents work whose goal is to advance 3D understanding in the field of Machine Learning. Offering a lightweight approach to improving scores, we believe the societal consequences of our work are limited to field advancements.

References

Assran, M., Bardes, A., Fan, D., Garrido, Q., Howes, R., Komeili, M., Muckley, M., Rizvi, A., Roberts, C., Sinha, K., Zholus, A., Arnaud, S., Gejji, A., Martin, A., Robert Hogan, F., Dugas, D., Bojanowski, P., Khalidov, V., Labatut, P., Massa, F., Szafraniec, M., Krishnakumar, K., Li, Y., Ma, X., Chandar, S., Meier, F., LeCun, Y., Rabbat, M., and Ballas, N. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Bae, G., Budvytis, I., and Cipolla, R. Estimating and exploiting the aleatoric uncertainty in surface normal estimation. In *International Conference on Computer Vision (ICCV)*, 2021.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Eigen, D., Puhrsch, C., and Fergus, R. Depth map prediction from a single image using a multi-scale deep network. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/91c56ce4a249fae5419b90cba831e303-Paper.pdf.

El Banani, M., Raj, A., Maninis, K.-K., Kar, A., Li, Y., Rubinstein, M., Sun, D., Guibas, L., Johnson, J., and Jampani, V. Probing the 3D Awareness of Visual Foundation Models. In *CVPR*, 2024.

Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., Hoppe, F., Thurau, C., Bax, I., and Memisevic, R. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Karaev, N., Makarov, I., Wang, J., Neverova, N., Vedaldi, A., and Rupprecht, C. CoTracker3: Simpler and better point tracking by pseudo-labelling real videos. 2024.

Lee, H.-Y., Huang, J.-B., Singh, M. K., and Yang, M.-H. Unsupervised representation learning by sorting sequence, 2017.

Ling, L., Sheng, Y., Tu, Z., Zhao, W., Xin, C., Wan, K., Yu, L., Guo, Q., Yu, Z., Lu, Y., et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22160–22169, 2024.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.-Y., Xu, H., Sharma, V., Li, S.-W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. Dinov2: Learning robust visual features without supervision, 2023.

Pariza, V., Salehi, M., Burghouts, G. J., Locatello, F., and Asano, Y. M. Near, far: Patch-ordering enhances vision foundation models' scene understanding. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Qro97zWC29.

Petersen, F., Borgelt, C., Kuehne, H., and Deussen, O. Monotonic differentiable sorting networks. In *International Conference on Learning Representations (ICLR)*, 2022.

Reizenstein, J., Shapovalov, R., Henzler, P., Sbordone, L., Labatut, P., and Novotny, D. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *International Conference on Computer Vision*, 2021.

Ruan, S., Dong, Y., Liu, H., Huang, Y., Su, H., and Wei, X. Omniview-tuning: Boosting viewpoint invariance of vision-language pre-training models. *arXiv preprint arXiv:2404.12139*, 2024.

Salehi, M., Gavves, E., Snoek, C. G. M., and Asano, Y. M. Time does tell: Self-supervised time-tuning of dense image representations. *ICCV*, 2023.

Salehi, M., Venkataramanan, S., Simion, I., Gavves, E., Snoek, C. G., and Asano, Y. M. Mosic: Optimal-transport motion trajectory for dense self-supervised learning. In *International Conference on Computer Vision*, 2024.

Siméoni, O., Vo, H. V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., Tolan, J., Brandt, J., Couprie, C., Mairal, J., Jégou, H., Labatut, P., and Bojanowski, P. DINOv3, 2025. URL https://arxiv.org/abs/2508.10104.

Teed, Z. and Deng, J. Raft: Recurrent all-pairs field transforms for optical flow. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M. (eds.), *Computer Vision – ECCV 2020*, pp. 402–419, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58536-5.

Xu, N., Yang, L., Fan, Y., Yang, J., Yue, D., Liang, Y., Price, B., Cohen, S., and Huang, T. Youtube-vos: Sequence-to-sequence video object segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

You, Y., Li, Y., Deng, C., Wang, Y., and Guibas, L. Multiview equivariance improves 3d correspondence understanding with minimal feature finetuning, 2024. URL https://arxiv.org/abs/2411.19458.

A. Relaxed Sorting and Soft Permutation Matrices

Sorting is a non-differentiable operation, which prevents gradient-based optimization when comparing ranked outputs. Traditional sorting uses discrete element swaps, such as $d_i' \leftarrow \min(d_i, d_j)$, which introduce discontinuities. To enable smooth learning, we adopt a differentiable sorting approach that relaxes these comparisons into continuous, pairwise soft-sorting operations.

Following (Lee et al., 2017), for any pair of distances d_i, d_j (drawn from a row of the distance matrix D), the relaxed sorting step is defined as:

$$\text{softmin}(d_i, d_j) = d_i f(d_j - d_i) + d_j f(d_i - d_j) \tag{8}$$

$$\text{softmax}(d_i, d_j) = d_i f(d_i - d_j) + d_j f(d_j - d_i) \tag{9}$$

where $f(x) = \frac{1}{\pi}\arctan(\beta x) + 0.5$ is a sigmoid-shaped function centered at $x = 0$, and $\beta > 0$ controls the steepness of the relaxation.

As $\beta \to \infty$, the function $f(x)$ approaches a step function, and the sorting converges to discrete behavior. In practice, we use moderate values ($\beta = 3$ or 20), which result in **soft permutations** that retain uncertainty and allow smooth gradient flow–ideal for ambiguous or occluded regions in video.

These pairwise comparisons are composed into **elementary swap matrices** $P_{\text{swap}}(d_i, d_j) \in \mathbb{R}^{np_r \times np_r}$, each being a near-identity matrix except for a 2×2 block that softly mixes elements i and j. The full differentiable sorting process applies a sequence of these swaps using the Odd-Even Sorting Network (Petersen et al., 2022):

$$P_t = \prod_{(i,j)\in\mathcal{M}_t} P_{\text{swap}}(d_i, d_j), \quad \mathcal{M}_t = \begin{cases} \text{odd indices}, & \text{if } t \text{ odd} \\ \text{even indices}, & \text{if } t \text{ even} \end{cases} \tag{10}$$

After $L = np_r$ steps, the final soft permutation matrix is obtained by composing all swap layers:

$$P = \prod_{t=1}^{L} P_t \in \mathbb{R}^{np_r \times np_r} \tag{11}$$

Each P matrix describes a probabilistic ranking over reference patches. Each row of P encodes a distribution over rank positions for one reference patch, while each column reflects the expected occupant of that rank. This soft structure captures a smooth approximation of the discrete sorting behavior.

In our implementation, we apply this procedure independently to each query patch. The resulting permutation matrices for a batch of size B with np_q query patches form a tensor:

$$P \in \mathbb{R}^{B \times np_q \times np_r \times np_r} \tag{12}$$

These permutation matrices capture the relative ordering of reference patches with respect to each query patch and serve as the foundation for our temporal consistency loss.

B. Further details of Loss formulation

Our loss formulation is intentionally designed to strengthen spatial discrimination in patch correspondences. Concretely, we supervise student permutation distributions using a reversed cross-entropy of the form:

$$\mathcal{L}_{\text{CE}}^{i} = -\sum_{j=1}^{np_r} P_{t,r}^{S}[i,j] \cdot \log\left(P_{0,r}^{T}[i,j] + \epsilon\right), \tag{13}$$

where the student distribution acts as the weighing measure.

If we express cross-entropy in terms of KL divergence and entropy,

$$CE(P_A, P_B) = KL(P_A \parallel P_B) + H(P_A), \tag{14}$$

the direction used in prior work such as NeCo, $CE(P_T, P_S)$, reduces (up to constants) to minimizing $KL(P_T \parallel P_S)$ because $H(P_T)$ is fixed when the teacher is frozen (receives EMA updates).

This corresponds to a mode-covering divergence: the student must distribute mass wherever the teacher assigns probability, encouraging broad, soft distributions that cover the teacher's uncertainty.

In contrast, the loss we apply, $CE(P_S, P_T)$, can be expressed as :

$$CE(P_S, P_T) = KL(P_S \parallel P_T) + H(P_S), \tag{15}$$

where $H(P_S)$ is not constant. Minimizing this loss therefore simultaneously reduces $KL(P_S \parallel P_T)$ while suppressing the entropy of the student, promoting high-confidence, sharply peaked ranking distributions.

The optimum of this loss is a deterministic distribution that assigns all mass to the teacher's highest-probability candidate, illustrating its mode-seeking nature. Thus, in ambiguous correspondence cases, our formulation encourages the student to make confident, spatially discriminative predictions rather than reproducing the teacher's diffuse uncertainty.

Figure 7 illustrates how multiple reference frames contribute to the loss.

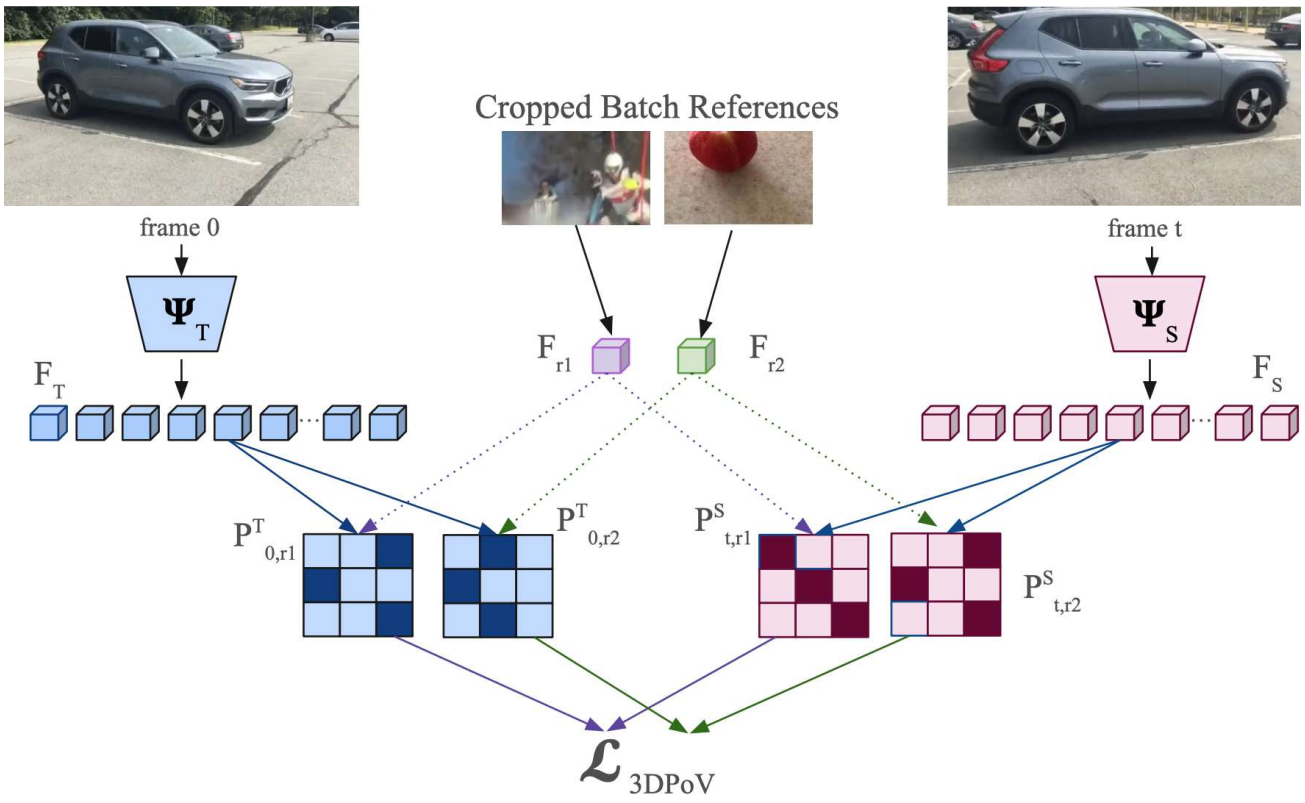

Figure 7. **Multiple reference contribution to the final loss.** Given two reference features F_{r1} and F_{r2} sampled from the feature bank, we compute corresponding permutation matrices $P^T_{0,r}$ and $P^S_{t,r}$ for each reference crop, comparing teacher (anchor frame $t = 0$) and student (future frame t) features. The permutation-based loss is computed for each reference independently by aligning the student and teacher permutations. The final loss is obtained by averaging over all such reference-specific losses.

C. Dataset choice

The choice of fine-tuning data significantly shapes the model's capacity to learn meaningful correspondences and geometric understanding from videos. Using video as a modality introduces variability along several axes: camera motion (static vs. dynamic), object movement, scene composition, occlusion patterns, viewpoint shifts, and lighting conditions. Capturing this diversity is essential for enhancing dense self-supervised learning, particularly when supervision operates at the level of patch correspondences and temporal consistency.

To this end, we fine-tune on a blend of complementary datasets, each contributing to different facets of the video distribution. For learning object-centered 3D structure and viewpoint-invariant patterns, we rely on CO3D (Reizenstein et al., 2021), which provides long video sequences of individual objects viewed under large viewpoint variations, often spanning 180 degrees or more (Figure 14). The dataset spans both indoor and outdoor contexts, and includes challenging factors such as occlusion, background clutter, and varying lighting conditions–making it especially well-suited for learning spatially consistent patch-level features under changing viewpoints and appearances.

For scene-level understanding, we incorporate DL3DV (Ling et al., 2024), a large-scale dataset of RGB-D video sequences captured using a commodity LiDAR-equipped phone. DL3DV contains over 10,000 dynamic scenes recorded in both indoor and outdoor settings, offering a wide range of spatial layouts and motion patterns (Figure 15). While we do not use depth annotations, the diversity in geometry and camera motion supports learning structure-aware features that generalize to complex 3D environments.

To encourage temporal coherence and robustness to real-world motion, we also train on YouTube-VOS (Xu et al., 2018), a large-scale video dataset containing high-resolution clips of everyday activities involving multiple objects, scene changes, occlusion events, and complex camera trajectories. These sequences provide valuable temporal signal, allowing the model to learn how to maintain patch-level consistency across time under natural, unconstrained motion.

Together, these datasets span a wide range of visual conditions–from single-object multiview videos to dynamic, cluttered scenes with complex motion. This diversity supports learning dense, geometry-aware representations that generalize across tasks such as surface normal estimation, depth prediction, and keypoint correspondence.

D. Experimental Setup

Following TimeTuning (Salehi et al., 2023), we initialize our models using publicly available pretrained DINO backbones. Specifically, we experiment with ViT-Base backbones from DINOv1 (Caron et al., 2021), DINOv2 with Registers (Oquab et al., 2023) and DINOv3(Siméoni et al., 2025). Unless otherwise stated, we use DINOv2R as our reference baseline and fine-tune only the final layer of the frozen backbone.

We train all models on 4 NVIDIA A100 GPUs using AdamW with cosine learning rate decay. For DINOv1-based variants, the feature extractor is updated with a learning rate of 1e-5 and the remaining layers with 1e-4, applying a weight decay of 1e-4. For DINOv2-reg models, which converge faster, we use 1e-7 for the extractor and 1e-6 for the rest of the model, with a weight decay of 1e-5. All DINOv3 experiments are conducted using the same training setup and evaluation protocol as DINOv2 to ensure comparability. DINOv3 experiments are performed under the same fine-tuning regime as DINOv2 to ensure comparability, though more tailored settings will be explored in future work.

While the absolute gain for the DINOv3 variant are modest, it represents a meaningful improvement given the high saturation of DINOv3, which is distilled from a 7B-parameter teacher trained on data distributionally similar to NAVI. Crucially, this improvement is statistically consistent across multiple runs (exhibiting very low standard deviation) and is achieved with orders of magnitude less compute than the original pretraining (our finetuning amounts to only 0.028% of a single DINOv3 epoch - 13 epochs of 9,242 samples using 4 frames comapred to 1689M) demonstrating that 3DPoV can efficiently extract geometric signal.

Training is lightweight compared to large-scale pretraining: fine-tuning requires roughly 5 hours on 4×A100 GPUs (≈20 GPU-hours) for 9,242 samples. For perspective, this is less than the cost of a single additional epoch of DINOv2 pretraining, which was conducted on 142M images and demanded multi-week training on large-scale GPU clusters (our finetuning amounts to only 0.34% of a single DINOv2-reg epoch - 13 epochs of 9,242 samples using 4 frames compared to 142M). Thus, the reported improvements are achieved with a negligible fraction of the original pretraining cost. These configurations follow the same optimization strategy as MoSiC (Salehi et al., 2024), with adjustments tailored to each backbone variant.

We don't finetune during evaluation, we follow the protocol from Probe3D (frozen models for correspondence, only finetune DPT heads for the depth evaluations). For our training, we only finetune the last 2 layers as we found to be enough to achieve improvements (refer to Table 20). We leveraged the learned representations and, with minimal compute requirements we are able to improve 3D understanding.

D.1. Dataset configuration

Due to the substantial imbalance in dataset sizes, we subsampled CO3D to ensure a more even distribution of training samples across the three sources. Specifically, with a frame sampling step of 10, the full CO3D dataset yielded 16,345 samples, while YouTube-VOS and DL3DV provided only 3,471 and 1,150 samples, respectively. To avoid training bias, we reduced the CO3D sample count to match that of YouTube-VOS.

Additionally, to compensate for the lower volume and higher complexity of DL3DV scenes–often containing multiple objects and fine structural details–we applied two different preprocessing strategies. One variant followed the standard resizing pipeline used across all datasets (resizing to 224×224). The other employed a center crop to match the 224×224 resolution used in our training pipeline. This center crop was necessary to ensure frame alignment required by the tracking module, and it is particularly favorable for preserving spatial and scene-level details that could otherwise be degraded by uniform resizing. The final training distribution consisted of 3,471 samples from CO3D, 3,471 from YouTube-VOS, and 2,300 from DL3DV.

D.2. Keypoint Matching

On **SPair-71k**, we follow the Probe3D protocol. Dense spatial features are extracted from both images in a pair, and cosine similarity is computed between all spatial locations. For each annotated keypoint in the source image, the target location is predicted as the position with the highest similarity. Recall is then computed based on the spatial distance between predicted and ground-truth keypoints at varying thresholds.

The benchmark categorizes pairs into three viewpoint groups (small, medium, large). The "All" split aggregates these categories and additionally includes pairs that do not fall into any viewpoint-defined subset. Due to the imbalance in dataset distribution, the "All" score is heavily influenced by small-viewpoint pairs and should not be interpreted as a direct average across difficulty regimes.

On **Navi**, evaluation leverages access to ground-truth 3D geometry and calibrated cameras. Following Probe3D, dense features are projected onto a 3D grid, and correspondences are established directly in 3D space. Performance is assessed in two complementary ways:

- **3D error** – the Euclidean distance between predicted and ground-truth 3D points, aligned into a shared coordinate frame using camera pose.
- **2D reprojection error** – the pixel-level distance between the reprojected 3D predictions and the ground-truth 2D keypoints.

We report recall at multiple thresholds (e.g., $<$2cm in 3D, $<$5px in 2D) and break down results by relative camera rotation. This dual evaluation provides a comprehensive test of whether features preserve geometric consistency across views.

D.3. Depth Estimation

Depth evaluation follows the protocol introduced by (Eigen et al., 2014), which includes both error-based and accuracy-based metrics. The primary error metric is the root mean squared error (RMSE), computed between the predicted depth values d_{pred} and ground truth d_{gt}. In addition, accuracy is measured using threshold-based metrics defined as the percentage of pixels for which the ratio between prediction and ground truth is within a multiplicative threshold. More formally, accuracy at threshold is

$$\delta_i(d^{pr}, d^{gt}) = \frac{1}{N} \sum_{j \in N} max\left(\frac{d_j^{pr}}{d_j^{gt}}, \frac{d_j^{gt}}{d_j^{pr}} < 1.25^i\right) \tag{16}$$

where $i \in 1, 2, 3$. The thresholds $\delta_1, \delta_2, \delta_3$ therefore correspond to tolerance levels of 1.25, 1.25^2 and 1.25^3 respectively.

For depth estimation, we report both scale-aware and scale-invariant metrics. The scale-aware RMSE (in meters) reflects absolute depth accuracy and is sensitive to global scale. In contrast, the scale-invariant RMSE normalizes per-frame predictions to account for scale ambiguity, capturing relative depth structure. Both are included for completeness.

As NAVI was not originally created as a depth benchmark, the authors of Probe3D adapt it by leveraging the underlying 3D geometry from multiview reconstructions to define a relative depth signal between pixels across view pairs. In this context, scale-invariant results are more aligned with the intent of the benchmark, as they emphasize relative spatial structure rather than absolute scale.

D.4. Surface normal estimation

For surface normal evaluation, we follow the setup described in .(Bae et al., 2021), where the goal is to assess the angular consistency between predicted normals n_{pred} and ground truth normals n_{gt}. Specifically, we compute the angle θ between the two vectors at each pixel and report the percentage of pixels for which this angular error is below predefined thresholds. Following the benchmark, we report accuracy at $11.25°$, $22.5°$, $30°$ along with RMSE for the angular error.

E. Robustness across random seeds

Our code is deterministic under a fixed training seed. We set the seed to 3 different values: 0, 42 and 100 and train 3 different models. Then, we evaluate these models on SPair keypoint matching and report mean and standard deviation per viewpoint bucket in Table 9.

Table 9. Mean ± std, across 3 seeds

View diff	Mean	Std
0	60.17	0.07
1	52.71	0.09
2	54.31	0.16
All	55.29	0.10

We further evaluate variance across these checkpoints on NAVI correspondence in Table 10. The low variance on both NAVI and SPair datasets further supports the stability of our training pipeline and the consistency of our gains.

Table 10. Mean ± std in NAVI correspondence, across 3 seeds

View diff	Mean	Std
θ_{0}^{15}	89.14	0.11
θ_{15}^{30}	68.85	0.17
θ_{30}^{60}	47.61	0.03
θ_{60}^{180}	31.65	0.02

F. Further Results

We report below the comparison with another available view-aware method (3DCorrEnhance (You et al., 2024)). While both methods improve on the baseline, 3DPoV achieves better scores across all viewpoint differences. We attribute it to the structured nature of the supervision in 3DPoV, which operates on relative relationships between patches rather than individual correspondences. This encourages more globally consistent feature organization, leading to consistent gains in keypoint matching performance.

Table 11. SPair-71k Keypoint Matching; Comparison to new baseline, both models are pretrained with DinoV2 with registers

MODEL	S / 0	M / 1	L /2	ALL
DINOv2R	58.20	51.56	53.41	53.47
3DCorrEnhance	59.61	52.16	54.39	54.64
3DPoV	**60.16**	**52.79**	**54.50**	**55.40**

To complement Probe3D and assess external validity, we also evaluated our encoder using the V-JEPA v2 (Assran et al., 2025) frozen video-understanding protocol on Something-Something V2(SSv2) (Goyal et al., 2017). While 3DPoV is

designed to improve spatial coherence and 3D-awareness, SSv2 contains many samples with inherent depth variation, pose changes, and fine-grained object–object interactions. Because spatial reasoning is a key component in interpreting these actions, SSv2 provides a relevant downstream test of whether our spatial improvements translate into better video understanding.

The official protocol uses 256×256 inputs and a 16×2×3 sampling pattern (16 frames × 2 temporal crops × 3 spatial crops), trained for 20 epochs. For feasibility, we retained the full validation set and multi-view evaluation but used a reduced configuration: 224×224 resolution, 8×2×3 inputs, and a balanced subset of the training data, training the probe for only 5 epochs. Despite this substantially lighter setup, our model achieves higher SSV2 probe accuracy than the DINO baseline (see Table 12), indicating that the spatial regularization introduced by 3DPoV yields more informative representations for downstream action understanding.

MODEL	BACKBONE	ACC %
DINOv2R	ViT-B/14	27.73
3DPoV	DINOv2R-B/14	**28.66**

Table 12. Frozen SSV2 Video Understanding Performance Using a Linear Probe on Top of the Encoder

The binned performance reported in Table 2 combines 3D and 2D evaluation metrics in accordance with the Probe3D benchmark. To provide further insight into these results, Table 13 presents a detailed breakdown of the individual 3D and 2D scores.

(a) 3D keypoint matching

Model	**Backbone**	**0.01m**	**0.02m**	**0.05m**
DINO	ViT-S16	**26.12**	**43.10**	**74.80**
TimeTuning	ViT-S16	24.17	41.44	73.36
MoSiC	ViT-S16	23.69	40.94	72.98
DINO	ViT-B16	26.12	43.10	74.80
NeCo	ViT-B16	24.24	41.20	73.20
3DPoV	ViT-B16	**26.52**	**43.53**	**74.99**
DINOv2-**reg**	ViT-B14	34.04	53.70	82.62
MoSiC-**reg**	ViT-B14	33.28	53.34	82.72
NeCo-**reg**	ViT-B14	32.01	51.47	81.45
3DPoV-**reg**	ViT-B14	**34.89**	**54.45**	**82.74**
DINOv3-**reg**	ViT-B16	38.33	**56.95**	83.69
3DPoV-**reg**	ViT-B16	**38.36**	56.93	**83.72**

(b) 2D keypoint matching

Model	**Backbone**	**5px**	**25px**	**50px**
DINO	ViT-S16	**3.47**	**22.69**	**37.49**
TimeTuning	ViT-S16	2.86	20.33	35.32
MoSiC	ViT-S16	2.78	20.04	34.82
DINO	ViT-B16	3.47	22.69	37.49
NeCo	ViT-B16	3.18	21.05	35.68
3DPoV	ViT-B16	**3.58**	**23.07**	**37.71**
DINOv2-**reg**	ViT-B14	4.43	30.20	47.96
MoSiC-**reg**	ViT-B14	4.18	29.38	47.47
NeCo-**reg**	ViT-B14	4.42	29.60	46.76
3DPoV-**reg**	ViT-B14	**4.63**	**31.03**	**48.65**
DINOv3-**reg**	ViT-B16	5.76	**36.68**	53.44
3DPoV-**reg**	ViT-B16	**5.77**	**36.68**	**53.46**

Table 13. Comparison of Navi Recall for 3D (a) and 2D (b) keypoint matching at different thresholds. Higher is better.

G. Further Ablations

For completeness, we also report ablation results on Navi keypoint matching in Table 14, complementing the SPair analysis presented in the main paper. The overall trends are consistent across the two benchmarks, confirming that our design choices generalize beyond 2D correspondence. On Navi, improvements under large viewpoint changes are smaller in magnitude compared to our preferred setup, yet the performance remains competitive. Taken together, the results across SPair and Navi highlight that 3DPoV delivers consistent benefits across both 2D and 3D correspondence evaluations.

G.1. Reference bank choice

We introduce patches from batch clips (external reference) to ensure diversity in similarity values and scenes. This setup follows the configuration from NeCo. In contrast, the addition of crops from the same clip (internal reference) ensures high-similarity anchors within the broader distribution, sharpening the ranking and ensuring a positive signal for the gradient. Nonetheless, the use of internal crops is limited to the number of frames used in training. We ablated this design choice in Table 15, reducing the number of references to match the number of frames and observe that indeed exclusive use of

Table 14. **Ablation of Key Design Choices in 3DPoV.** We report Keypoint Matching Recall on NAVI. Each experiment isolates one design parameter, with other settings held fixed.

(a) References Extracted by

MODEL	θ_0^{15}	θ_{15}^{30}	θ_{30}^{60}	θ_{60}^{180}
Teacher	87.56	67.61	47.10	**31.35**
Student	**89.22**	**69.23**	**47.48**	31.33

(b) Number of frames

FRAMES	θ_0^{15}	θ_{15}^{30}	θ_{30}^{60}	θ_{60}^{180}
1	88.47	68.34	47.46	**31.52**
2	**89.86**	69.60	47.21	30.67
4	89.80	**69.73**	**47.32**	**30.83**

(c) Dataset choice - Navi eval

DATA	θ_0^{15}	θ_{15}^{30}	θ_{30}^{60}	θ_{60}^{180}
CO3D	**89.96**	**69.77**	47.29	30.62
YTVoS	89.70	69.63	**47.33**	30.92
DL3DV	89.72	69.71	47.39	**30.96**
YT-CO3D	89.76	69.63	47.32	30.85
CO3-YT-DL	89.80	69.73	47.32	30.83

(d) Step size on frame sampling

STEP SIZE	θ_0^{15}	θ_{15}^{30}	θ_{30}^{60}	θ_{60}^{180}
2	89.13	69.20	**47.54**	**31.42**
4	**89.22**	**69.23**	47.48	31.33
6	88.87	68.81	47.32	31.24

(e) Type of Resampling

RESAMPLING	θ_0^{15}	θ_{15}^{30}	θ_{30}^{60}	θ_{60}^{180}
Resized	88.82	68.91	47.60	**31.52**
Latent	**89.22**	**69.23**	**47.48**	31.33

(f) Choice of Point Tracker

TRACKER	θ_0^{15}	θ_{15}^{30}	θ_{30}^{60}	θ_{60}^{180}
RAFT	88.91	69.03	47.46	31.37
CoTrackerV3	**89.22**	**69.23**	**47.48**	**31.33**

internal references result in better performance (+0.32% on 'All'). This suggests that internal reference patches maximize the coverage of the specific dynamic scene, which is more valuable for learning fine-grained 3D correspondence.

REFERENCES	S / 0	M / 1	L /2	ALL
External (4)	59.66	52.27	54.40	54.89
Internal (1) + external (3)	59.87	52.44	54.27	55.07
Internal (4)	**60.24**	**52.71**	**54.36**	**55.39**

Table 15. SPair-71k keypoint matching ablations. For a reference pool of size 4 we compare different internal/external splits

In order to further probe the importance of having a discriminative pool of references, we kept the batch size consistent, but sampled external patches only from one other element from the batch. Results are reported in Table 16.

MODEL	S / 0	M / 1	L /2	ALL
Less discr. sampl.	59.91	52.41	54.31	54.95
3DPoV sampling	**60.16**	**52.79**	**54.50**	**55.40**

Table 16. SPair-71k Keypoint Matching; ablating the sampling of external references, we compare less discriminative sampling where external references are obtained from only one video in the batch

The results strongly indicate that having discriminative references positively impacts the score .

G.2. Choice of tracker

3DPoV is tracker agnostic, as shown in the ablation in Table 17. Our method still improved the performance compared to even weaker point trackers such as RAFT, supporting that 3DPoV is tracker agnostic. This means our method can benefit even further from better trackers according to the field progress, in a plug-and-play manner.

TRACKER	S / 0	M / 1	L /2	ALL
RAFT	59.81	52.43	54.31	55.07
CoTrackerV2	59.03	51.74	53.82	54.18
CoTrackerV3	**60.16**	**52.79**	**54.50**	**55.40**

Table 17. SPair-71k Keypoint Matching; Ablating choice of tracker

G.3. Benefit of the differentiable sorting component

A further experiment focuses on removing the sorting algorithm, and simply applying the cross-entropy loss on the similarity matrices. The results are presented in Table 18

METHOD	S / 0	M / 1	L /2	ALL
Similarity matrix	59.49	52.36	54.42	54.81
Sorted similarity matrix	**60.16**	**52.79**	**54.50**	**55.40**

Table 18. SPair-71k Keypoint Matching; Ablating sorting module by removing the differentiable sorting module and leverage the similarity matrices directly

While directly leveraging similarity matrices showcases a degradation in SPair scores, we observe an even grater drop for the Navi eval in Table 19.

Model	θ_0^{15}	θ_{15}^{30}	θ_{30}^{60}	θ_{60}^{120}
DINOv2R baseline	87.87	67.45	47.15	31.58
Similarity matrix	88.62	68.23	47.26	31.41
Sorted similarity matrix	**89.18**	**68.98**	**47.64**	**31.63**

Table 19. Navi Correspondance.

Without a sorting component, the method struggles to perform, achieving better results in easier viewpoint cases, but degrading baseline score across the last two buckets. The DiffSort modules allows for a more complex mapping, that prioritizes high similarity matches while also weighting in possible alternatives, which proves efficient in more challenging viewpoint scenarios.

UNFROZEN BLOCKS	S / 0	M / 1	L /2	ALL
Blocks 8-11	57.66	49.84	51.48	52.50
Blocks 10-11	**60.16**	**52.79**	**54.50**	**55.40**
Block 11	58.64	51.72	53.41	53.79

Table 20. SPair-71k Keypoint Matching; we unfreeze a number of layers and experiment under the same setup

NO. OF REF	S / 0	M / 1	L /2	ALL
3	59.94	52.43	54.20	55.10
4	59.87	52.44	54.27	55.07
5	**60.16**	**52.79**	**54.50**	**55.40**
7	59.84	52.37	54.28	55.04

Table 21. SPair-71k keypoint matching ablations. Ablating number of references

G.4. Loss direction

We ablate the direction of the loss to further support the formulation described in Appendix B, including symmetric loss as well as KL divergence(variants) in Table 22. For KL we can write it as CE(t,s) - H(s). Also, we ablate different coefficients

of H(s) to make the ablations even more comprehensive. As shown our default achieves SOTA, we attribute this to the chosen direction encouraging sharper target distributions, leading to more discriminative matching for correspondence tasks.

Table 22. SPair-71k Keypoint Matching; loss ablation

LOSS	S / 0	M / 1	L /2	ALL
$CE(t,s)$	58.17	51.29	53.13	53.27
$CE(s,t)$	**60.16**	**52.79**	**54.50**	**55.40**
$CE(t,s)+CE(s,t)$	58.13	51.28	53.12	53.33
$CE(t,s)-0.1*H(s)$	58.11	51.35	53.23	53.35
$CE(t,s)-0.8*H(s)$	58.13	51.35	53.25	53.36

G.5. Batch size

The batch size used in all reported experiments is 32, we further add a breakdown of performance for varying batch sizes in Table 23.

Table 23. SPair-71k Keypoint Matching; batch size ablation

BS	S / 0	M / 1	L /2	ALL
8	59.63	52.44	54.24	54.86
16	59.77	52.40	54.32	54.95
32	**60.16**	**52.79**	**54.50**	**55.40**

As shown, increasing batch size results in better scores across all viewpoints for 3DPoV.

H. Mapping to Probe3D Benchmark

We compare our reproduced baselines and reported results with the original Probe3D study in Table 24, Table 25, Table 26, Table 27. Minor misalignments are expected due to differences in environment and training setup, but overall trends are consistent with the original benchmark.

H.1. Fix in probe3D NAVI keypoint matching flow

We identified and revised a key fault in the NAVI evaluation. The Probe3D evaluation uses center padding to accommodate backbones with different patch sizes. Since NAVI uses 512 x 512 inputs, this padding is triggered only for DINOv2 ViT-14 variants but not ViT-16 variants. The issue is that NAVI evaluation relies on a dense per-pixel 3D correspondence map (xyz_grid) that is defined in the original image coordinate frame. When center padding is applied only to the image before feature extraction, the feature map and the xyz_grid no longer refer to the same spatial coordinates, and this misalignment is further propagated through the subsequent rescaling. In this sense, we fixed the evaluation pipeline by applying the same center-padding transformation to xyz_grid as to the input images, so that the extracted features and the underlying geometry remain spatially aligned.

Table 24. SPair-71k viewpoint difference. 0: No significant view difference (same view or minimal changes), 1: Moderate viewpoint difference, 2: Large viewpoint difference. Here † represents the bechmark reported values

Model	Backbone	Data	S / 0	M / 1	L /2	All
DINO †	ViT-B16	IN-1k	30.4	24.0	24.3	26.8
DINO	ViT-B16	IN-1k	30.19	24.22	24.35	26.39
3DPoV	ViT-B16	CO3-YT-DL	**31.77**	**25.74**	**25.80**	**28.16**
DINOv2-**reg**†	ViT-B14	LVD	58.3	51.4	53.4	53.7
DINOv2-**reg**	ViT-B14	LVD	58.20	51.56	53.41	53.47
3DPoV	ViT-B14	CO3-YT-DL	**60.16**	**52.79**	**54.50**	**55.40**

Table 25. Navi Performance. Here † represents the bechmark reported values. Here, DINOv2 values are reported before NAVI fix to align with the original reported values.

Model	Backbone	Data	θ_0^{15}	θ_{15}^{30}	θ_{30}^{60}	θ_{60}^{180}
DINO†	ViT-B16	IN-1k	86.0	56.0	31.3	20.3
DINO	ViT-B16	IN-1k	86.13	56.92	33.37	19.74
3DPoV	ViT-B16	CO3-YT-DL	**86.42**	**57.18**	**33.77**	**20.42**
DINOv2-**reg**†	ViT-B14	LVD	89.0	67.3	44.8	31.1
DINOv2-**reg**	ViT-B14	LVD	87.92	67.74	47.18	**31.57**
3DPoV	ViT-B14	CO3-YT-DL	**89.22**	**69.23**	**47.48**	31.33

Table 26. Depth estimation results on Navi.

Model	Backbone	Scale-Aware				Scale-Invariant			
		$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	RMSE $\downarrow$	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	RMSE $\downarrow$
DINOv2-**reg**†	ViT-B/14	-	-	-	-	66.56	87.94	94.74	0.0806
DINOv2-**reg**	ViT-B14	57.62	82.49	91.97	0.0960	68.49	87.89	93.95	0.0778
3DPoV-**reg**	ViT-B14	**59.17**	**83.61**	**92.59**	**0.0933**	**69.61**	**88.47**	**94.19**	**0.0757**

Table 27. Surface normal estimation results on Navi.

Model	Backbone	11.25° ↑	22.5° ↑	30° ↑	RMSE ↓
DINOv2-**reg**†	ViT-B/14	45.81	72.00	81.28	25.66
DINOv2--**reg**	ViT-B14	37.10	65.93	77.09	28.0693
3DPoV-**reg**	ViT-B14	**38.29**	**67.00**	**77.86**	**27.7798**

I. Comparing with the most similar model

During our ablation studies, we adopted an image processing strategy similar to NeCo–cropping frames followed by ROI alignment of the crops. This defines the 3DPoV-1frame experiment. As shown in Table 28, when compared directly to the baseline and NeCo, our approach demonstrates stronger ability to learn robust 3D representations, particularly under medium and large viewpoint shifts. This trend is consistent with the central challenge emphasized by the Probe3D benchmark, where performance typically drops sharply at larger viewpoint changes. We also note that the 'All' score–an aggregate over all categories, including samples not belonging to any category–is biased toward easier (small-shift) cases, and therefore differs in interpretation from a category-wise average.

Table 28. SPair71k Keypoint Matching results. Compared to the most relevant prior method (NeCo), 3DPoV attains similar performance in the 'All' category while offering improvements in the more challenging Medium and Large viewpoint shift categories.

Model	Backbone	Data	S / 0	M / 1	L /2	All
DINOv2-**reg**	ViT-B14	LVD	58.20	51.56	53.41	53.47
NeCo-**reg**	ViT-B14	COCO	**59.57**	49.06	52.35	54.42
3DPoV-1Frame-**reg**	ViT-B14	CO3-YT-DL	59.49	**52.18**	**54.22**	**54.66**

J. Failure cases

Since our method inherits correspondences from CoTracker, its limitations can influence supervision quality. We therefore explore such cases in this section.

A representative scenario is shown in Figure 8, where multiple similar subjects (e.g., several blue fish in blue water) move in and out of frame. When the tracked fish exits the view, some points "jump" to a visually similar fish, and the tracker is unable to recover the original correspondence once it reappears. This produces an incorrect trajectory rather than a complete tracking loss.

3DPoV does not fully correct such failures, but its visibility-aware weighting reduces their impact. When a point drifts or becomes unreliable, its predicted visibility decreases, naturally lowering its contribution in the loss. As a result, these ambiguous temporal matches influence supervision less strongly, instead of being propagated as confident signals.

While this mechanism improves robustness in cluttered or out-of-frame motion, scenes with persistent ambiguity across many frames (e.g., long occlusions or repeated textures) remain challenging and are an interesting direction for future improvement.

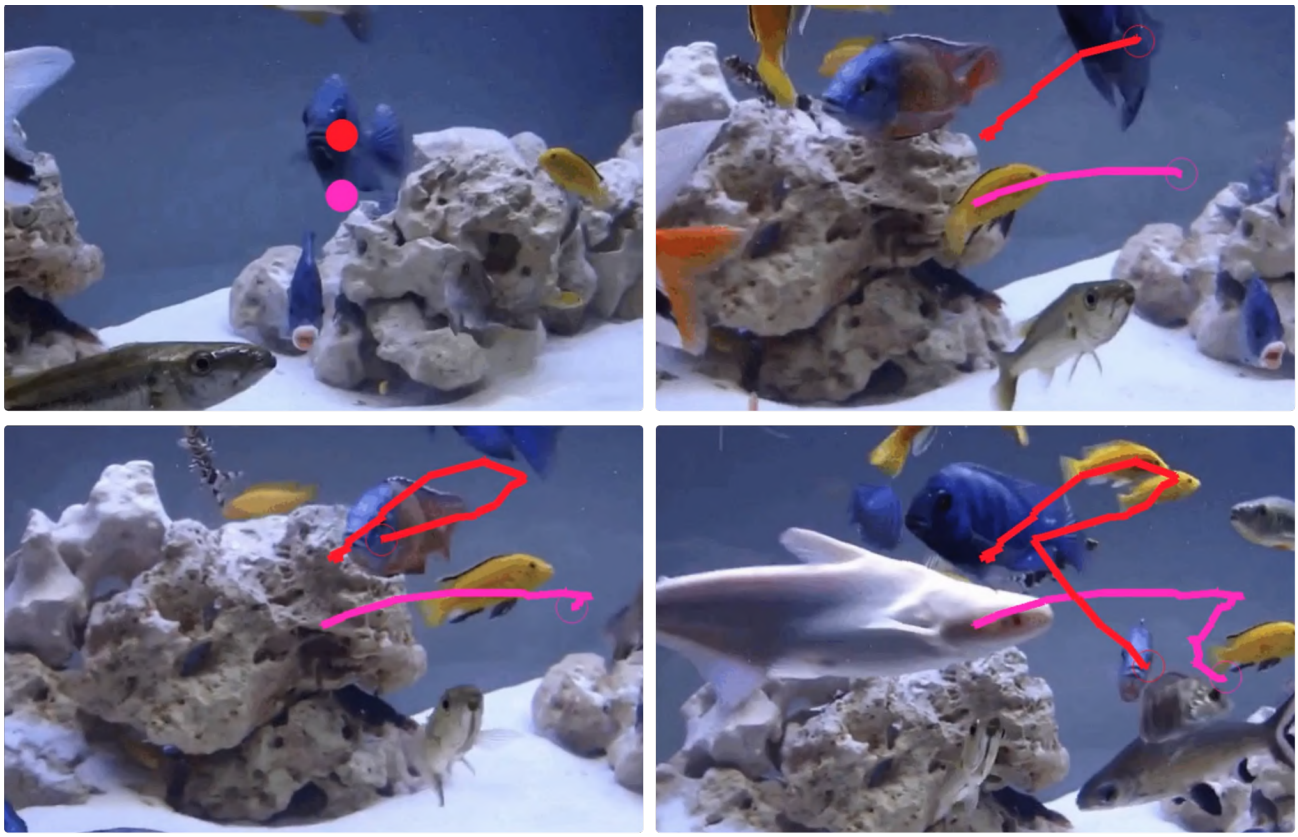

Figure 8. Example of failure cases for tracking. Here we observe how the red tracked point jumps on similar pixels once the subject gets out of frame. The tracked dot now shows only the contour, indicating reduced visibility value which results in less weight during matching

K. Light and occlusions robustness experiments

The normal light and intense lighting condition subsets contain 495 and 498 samples respectively (compared to 552 subsampled at random from 2219 test entries in default NAVI eval), with a distribution over the 0-30, 30-60, 60-90, 90-120 buckets of 84/130/145/136 and 83/125/139/151 samples respectively. Some example samples are showcased in Figure 9

For the occlusion experiment, synthetic boxes were overlaid on the target view, or on both source and target views, before computing correspondences. This means valid keypoints may be covered by the occluding boxes. We handle this by discarding occluded keypoints and evaluating accuracy only on the remaining visible ones, which assesses whether models

Figure 9. Samples selected for the two subgroups of the light experiment

can maintain reliable correspondences in the presence of occlusions that obscure parts of the image without necessarily blocking the keypoints themselves, as well as occlusions that obstruct key components of the object as shown in Figure 10.

L. Additional Visualizations

Figure 10. Example of synthetic occlusions present in the experiment

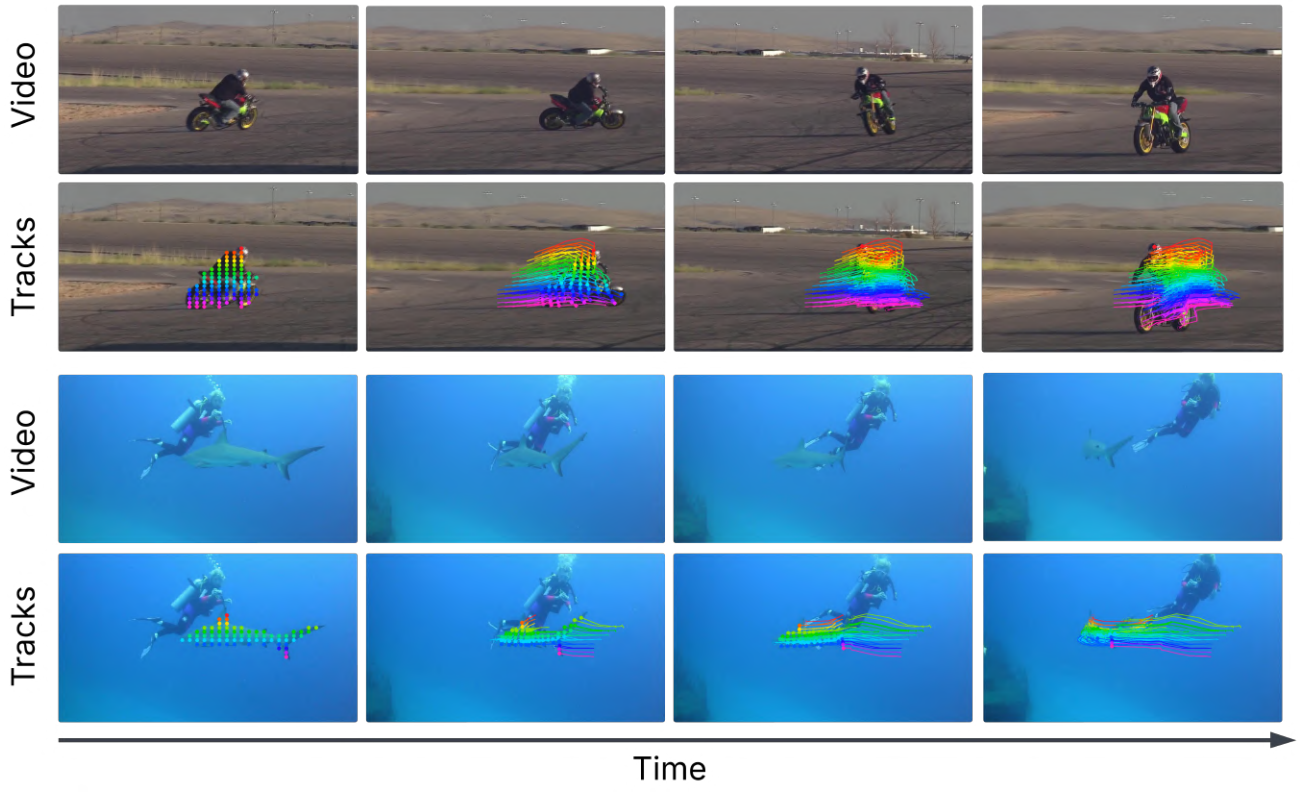

Figure 11. Examples of YTVoS movements and tracking quality

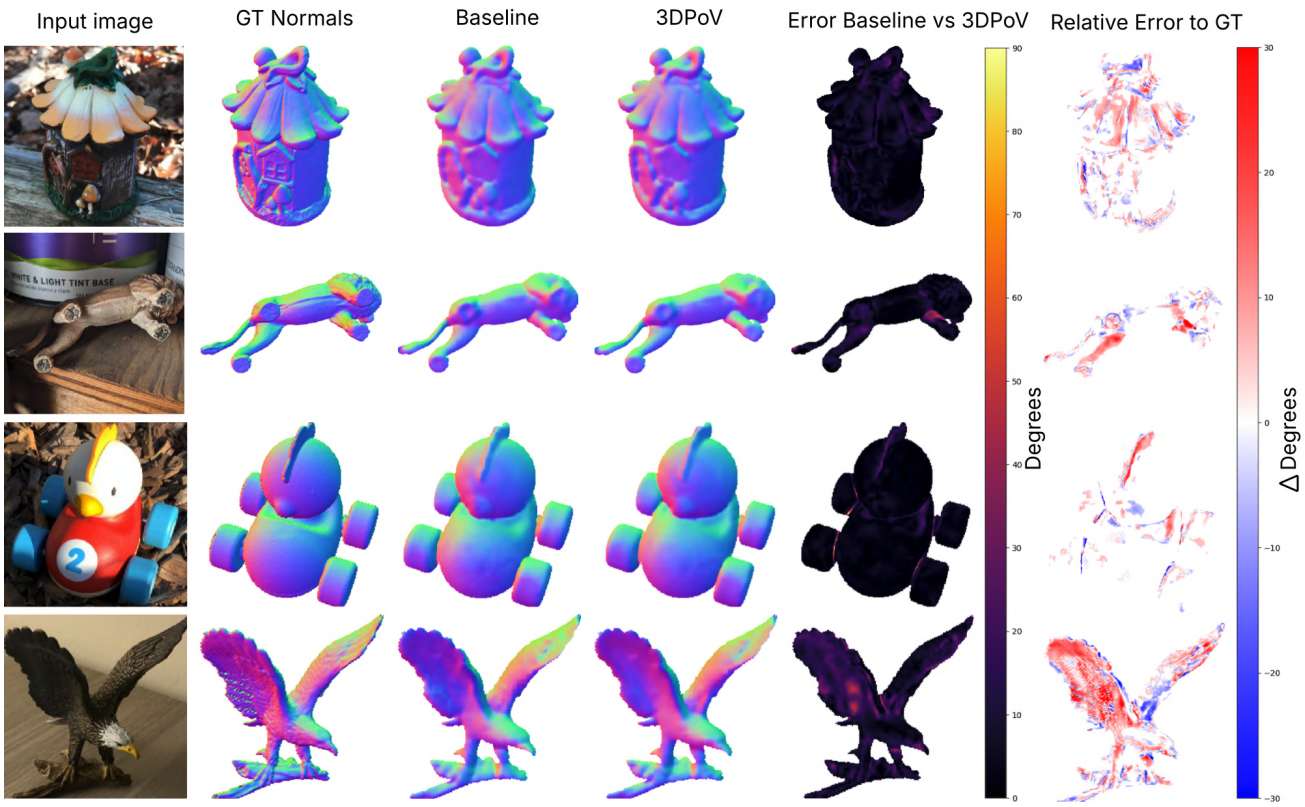

Figure 12. More examples of surface normal qualitative results

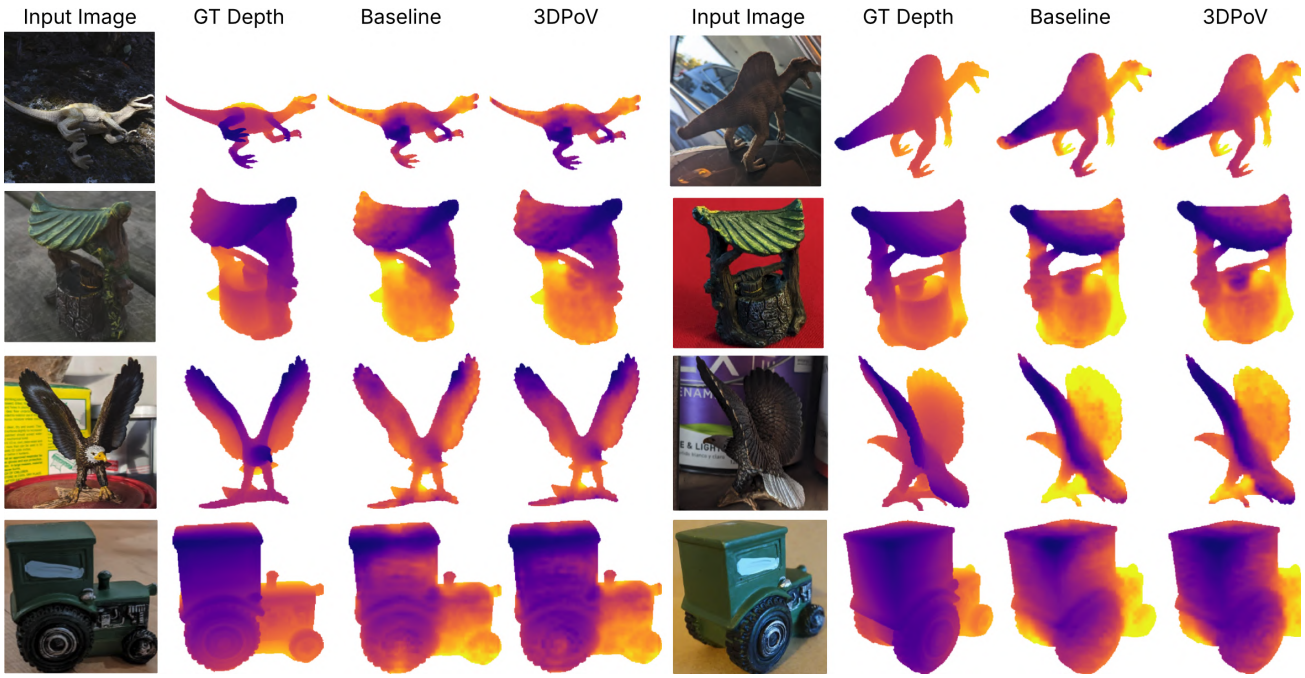

Figure 13. **More Depth Qualitative Examples.** Comparing predicted depth maps from Baseline (DinoV2-reg) and **3DPoV**. Ground truth (GT) depth is provided for reference

Figure 14. **More samples of CO3D movements and tracks**

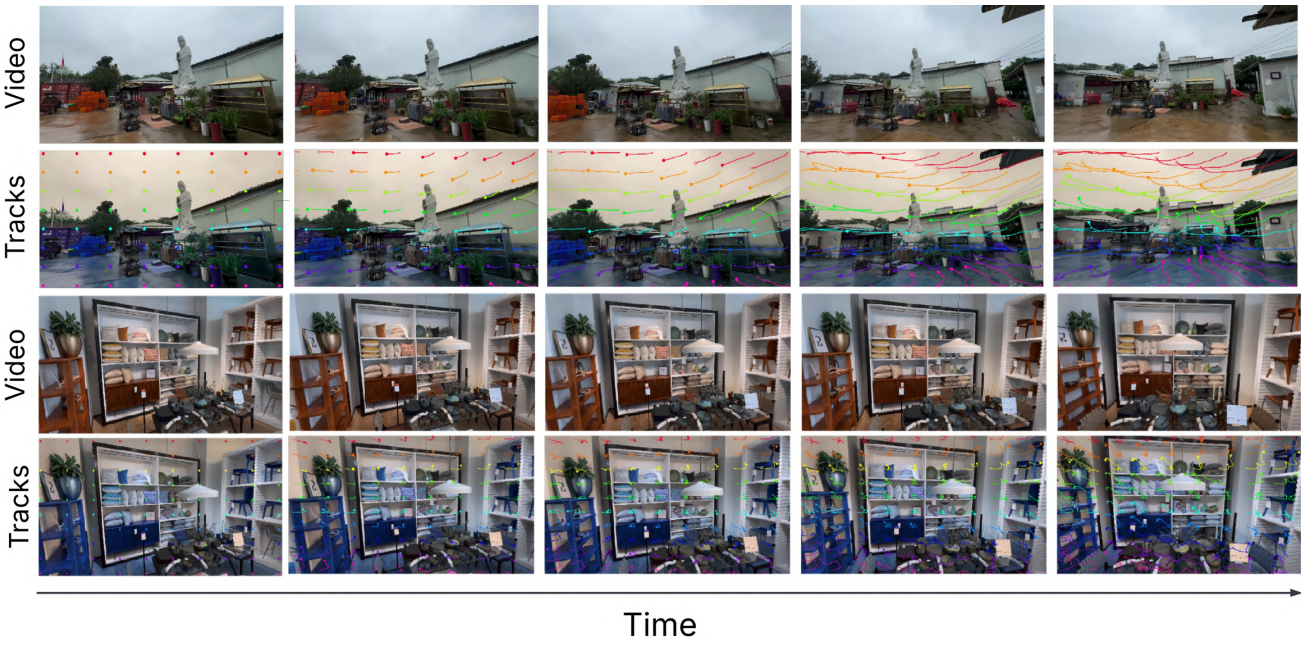

Figure 15. **Tracking behaviour acros DL3DV samples**