ProMo3D: Probing Motion Cues in Frozen 3D Foundation Models

Xiaoang Zhang[1*] **Mert Kiray**[1,2,3] **Yordanka Velikova**[1,2]
Manoj Biswanath[1,2] **Benjamin Busam**[1,2,3]

[1]Technical University of Munich (TUM), Germany
[2]Munich Center for Machine Learning (MCML), Germany
[3]Obsphera, Munich, Germany

Abstract

Feed-forward 3D foundation models have recently reshaped multi-view reconstruction, yet dynamic 4D understanding still often relies on large-scale motion supervision or hand-crafted, architecture-specific heuristics. We ask whether frozen 3D foundation model tokens already encode motion-relevant information that can be exposed by lightweight readout functions, rather than re-learning motion-specific features with high-capacity decoders. Through probing on three representative backbones, we find that static–dynamic separation is decodable from frozen token representations. Our rank-restricted and spectral analyses on π^3X reveal that the recovered motion cues are concentrated in a low-dimensional subspace. Building on this, we propose Probing-based Motion Distillation (PMD), which converts pairwise motion-state compatibility scores against a fixed anchor token into dense motion masks. Trained on only 160 YouTube-VOS videos using a single 16 GB GPU, PMD transfers zero-shot to DAVIS, SegTrackv2, and FBMS-59 without target-benchmark fine-tuning. With multi-view pretrained geometric backbones, PMD outperforms training-free heuristics and approaches substantially heavier supervised and self-supervised methods, suggesting that motion decoding from frozen 3D foundation models can be lightweight and data-efficient.

Project page: `https://xiaoang0211.github.io/ProMo3D/`

1 Introduction

Driven by large-scale geometry pretraining, feed-forward 3D foundation models have recently reshaped 3D vision. Methods such as DUSt3R [35], VGGT [34], π^3/π^3X [36], and Depth Anything 3 [21] bypass traditional iterative optimization and recover 3D geometry directly from unposed multi-view images.

Despite these advances, dynamic 4D understanding remains challenging because static geometry, camera-induced motion, and independently moving objects must be disentangled. Existing approaches typically follow two paradigms. Large-scale supervised methods [45, 11, 47] learn motion and object tracking from dense annotations but require substantial labeling effort and high computational costs. Training-free alternatives [9, 5] extract motion cues from frozen 3D foundation models using hand-crafted, architecture-specific heuristics such as attention layers or key–query Gram matrices. While efficient, these heuristics are often brittle and underperform learning-based motion decoders by a large margin, as shown by He et al. [8].

Several observations suggest a third paradigm: exploiting motion cues already encoded in frozen 3D models to bridge the gap between efficiency and performance. First, the increasing use of 4D

*This work was conducted while Xiaoang Zhang was a guest researcher at the Technical University of Munich.

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

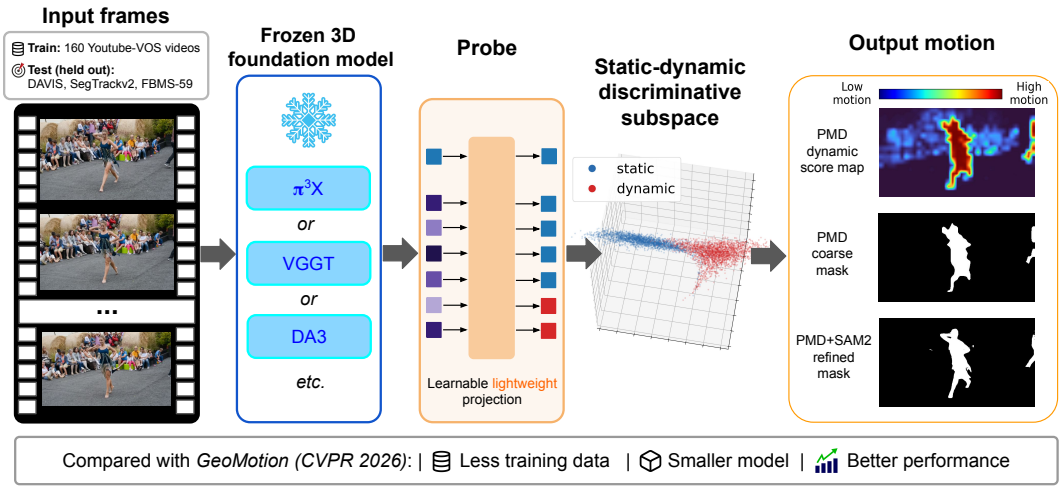


Figure 1: Overview of Probing-based Motion Distillation (PMD). Given an input frame group, a frozen feed-forward 3D foundation model such as π^3X [36], VGGT [34], or Depth Anything 3 (DA3) [21] produces token representations. A lightweight probe trained on only 160 videos maps these features into a probe-induced static–dynamic discriminative representation. PMD then converts anchor-based pairwise probe scores into dense dynamic score maps, which are thresholded into coarse masks and optionally refined with SAM2 [29]. It transfers zero-shot to DAVIS [26, 27], SegTrackv2 [17], and FBMS-59 [14], while requiring substantially less training data and trainable capacity than high-capacity motion decoders such as GeoMotion [8].

training data from multi-domain resources [32, 7, 49] suggests that large feed-forward 3D backbones may acquire motion-relevant cues during geometry pretraining. Second, their implicit multi-view geometric representations appear to remain robust in dynamic scenes: recent work [45, 9, 47] shows competitive camera pose estimation under scene motion, while He et al. [8] demonstrates that latent 4D geometry from π^3 [36] provides a strong prior for motion segmentation, albeit with a high-capacity neural network-based decoder. In addition, Bratulić et al. [4] shows that intermediate layers of VGGT encode geometric constraints such as correspondences and epipolar geometry. This suggests that regions violating multi-view geometric consistency may also be detectable in frozen token representations. Taken together, these findings motivate our central inquiry: can motion cues be recovered from frozen 3D foundation model tokens using low-capacity probes, without complex decoders or expensive fine-tuning?

We investigate this question through supervised representation probing [1, 3]. By training lightweight probes on frozen 3D foundation models, we find that motion cues within token representations can be mapped into a compact, probe-induced static–dynamic discriminative subspace that generalizes across diverse dynamic scenarios. Building on this finding, we introduce Probing-based Motion Distillation (PMD), which converts pairwise motion-state compatibility scores into dense motion masks.

Our contributions are summarized as follows:

- **Systematic localization of motion cues in 3D backbones.** We conduct an extensive empirical study on three representative feed-forward 3D foundation models: π^3X [36], Depth Anything 3 [21], and DUSt3R [35]. We identify the layers in which motion cues emerge in these frozen backbones, showing that they can be exposed by probing without relying on architecture-specific heuristics.
- **Characterization of a low-rank motion subspace.** Through rank-restricted probing and spectral analysis on π^3X, we find the probe-induced motion representation to be remarkably compact. We show that a low-rank quadratic probe is sufficient to extract dense motion masks, suggesting that motion understanding in 3D foundation models is largely a lightweight decoding problem rather than one requiring extensive feature re-learning.
- **Data-efficient Probing-based Motion Distillation (PMD).** We introduce PMD, a lightweight framework trained on 160 YouTube-VOS [41] videos using a single consumer-

grade GPU with 16 GB VRAM. PMD transfers zero-shot to standard motion segmentation benchmarks without target-benchmark fine-tuning. With multi-view pretrained geometric backbones, it outperforms training-free heuristics and remains competitive with supervised and self-supervised methods, while requiring substantially less training data.

2 Related Work

2.1 Feed-forward 3D Foundation Models

Feed-forward 3D foundation models such as DUSt3R [35], VGGT [34], and π^3 [36] have reshaped multi-view reconstruction by directly predicting 3D geometry and camera parameters from unposed images, bypassing iterative Structure-from-Motion pipelines [30]. Extensions such as Dens3R [6], AnySplat [12], and Depth Anything 3 [21] broaden this paradigm toward dense geometry, feed-forward 3D Gaussian reconstruction, and ray prediction. As recent backbones are increasingly trained on diverse geometric data, including dynamic sources [7, 32, 49], they provide a potential substrate for extracting motion cues.

2.2 Dynamic Scene Reconstruction and Motion Segmentation

Dynamic scene reconstruction remains an active research area due to challenges in disentangling geometry, camera motion, and object motion in diverse scenarios. NeRF-based approaches [28, 19] optimize scene-specific dynamic representations, often with auxiliary priors such as optical flow, depth, or masks. Recent feed-forward 4D methods [45, 11, 47] move toward direct dynamic-scene reconstruction by predicting geometry, camera pose, and motion-related quantities from unposed views. A parallel line of work [24, 37, 42, 38] focuses on motion segmentation, separating independently moving regions from static scene content using motion–appearance cues, optical flow, object-centric grouping, or appearance refinement.

Recent methods also attempt to extract motion cues from frozen 3D foundation models. Training-free approaches such as Easi3R [5] and VGGT4D [9] rely on hand-crafted heuristics, including attention patterns or key–query Gram matrices. While computationally efficient, these heuristics are typically brittle, architecture-specific, and show a substantial performance gap compared to learned motion segmentation approaches, as shown by He et al. [8]. Most closely related to our motivation, He et al. [8] show that latent 4D geometry from π^3 provides a strong prior for motion segmentation. However, their approach still relies on a high-capacity neural decoder and additional optical-flow cues. Our work targets the gap between these two regimes: training-free heuristics are often brittle, whereas learned decoders are substantially more complex and expensive to train.

2.3 Representation Probing and Latent Analysis

Representation probing [1, 3] tests whether specific information is decodable from neural representations. Recent studies show that pretrained vision and world models can contain structured latent information not explicit in the training objective: Maes et al. [23] probe physical quantities from JEPA-style latent representations, and Bratulić et al. [4] show that VGGT encodes geometric priors such as correspondences and epipolar constraints. Most closely related, Li et al. [18] show that object binding emerges in self-supervised ViT encoders and can be most effectively decoded using quadratic probes. We adopt this probing perspective for motion understanding, using low-capacity probes to test whether frozen 3D representations can be mapped into a static–dynamic discriminative subspace.

3 Method

Given an input frame group, a frozen feed-forward 3D foundation model produces target-frame token features at block ℓ:

$$\mathbf{H}^{(\ell)} = \{\mathbf{h}_0^{(\ell)}, \mathbf{h}_1^{(\ell)}, \ldots, \mathbf{h}_N^{(\ell)}\}, \qquad \mathbf{h}_i^{(\ell)} \in \mathbb{R}^d. \tag{1}$$

Here, N is the number of target-frame tokens and d is the feature dimension. Adjacent frames provide temporal or multi-view context, while token pairs are sampled only within the target frame.

Each token is assigned a binary motion label $y_i \in \{0, 1\}$ by projecting the dynamic mask onto the token grid, where $y_i = 1$ denotes a dynamic token and $y_i = 0$ denotes a static token. Following Li

et al. [18], we formulate motion probing as pairwise classification. For a token pair (i, j), we define

$$z_{ij} = \mathbb{I}[y_i = y_j], \tag{2}$$

where $\mathbb{I}[\cdot]$ is the indicator function, which equals 1 if the condition is true and 0 otherwise. Thus, $z_{ij} = 1$ indicates that two tokens share the same motion state, and $z_{ij} = 0$ indicates different motion states.

A probe predicts the pairwise probability $\hat{z}_{ij} \in [0, 1]$. All probes are trained on sampled token pairs $\mathcal{P}$ using binary cross-entropy while keeping the backbone parameters frozen:

$$\mathcal{L}_{\text{probe}} = -\frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \left[z_{ij} \log \hat{z}_{ij} + (1 - z_{ij}) \log(1 - \hat{z}_{ij}) \right]. \tag{3}$$

3.1 Pairwise Probe Functions

Let $\mathbf{h}_i, \mathbf{h}_j \in \mathbb{R}^d$ denote two token features. We evaluate three lightweight probe functions that differ in how they model pairwise motion-state compatibility: independent linear terms, channel-wise multiplicative interactions, and cross-channel bilinear interactions.

Linear Probe. The linear probe tests whether motion-state compatibility is directly decodable from individual token features:

$$\hat{z}_{ij}^{\text{lin}} = \sigma\big(\mathbf{w}_1^\top \mathbf{h}_i + \mathbf{w}_2^\top \mathbf{h}_j + b\big), \tag{4}$$

where $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are learnable parameters.

Diagonal Quadratic Probe. The diagonal quadratic probe captures channel-wise multiplicative interactions:

$$\hat{z}_{ij}^{\text{diag}} = \sigma\big(\mathbf{h}_i^\top \mathbf{W}_{\text{diag}} \mathbf{h}_j + b\big), \qquad \mathbf{W}_{\text{diag}} = \frac{1}{\sqrt{d}} \operatorname{diag}(\boldsymbol{\alpha}), \tag{5}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^d$.

Low-Rank Quadratic Probe. Our primary probe is a low-rank bilinear form:

$$\hat{z}_{ij}^{\text{lr}} = \sigma\big(\mathbf{h}_i^\top \mathbf{W}_{\text{lr}} \mathbf{h}_j + b\big), \qquad \mathbf{W}_{\text{lr}} = \frac{\mathbf{U}^\top \mathbf{V} + \mathbf{V}^\top \mathbf{U}}{2\sqrt{d}}, \tag{6}$$

where $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{k\times d}$ and $k \ll d$. Here, k is the factor-rank hyperparameter; symmetrization gives $\operatorname{rank}(\mathbf{W}_{\text{lr}}) \le 2k$. The symmetrized form matches the symmetric pair label $z_{ij} = z_{ji}$ and enforces order invariance. By varying the rank k, we test whether motion cues can be decoded through a compact, probe-induced low-rank bilinear representation.

3.2 Probing-based Motion Distillation

Probing-based Motion Distillation (PMD) converts the learned pairwise compatibility into a dense motion estimator. While the probe is trained to predict the motion-state compatibility between arbitrary token pairs, PMD additionally formulates a unary approach by designating an anchor token $\mathbf{h}_a$ with a fixed spatial index a across all frames.

Specifically, we use the first token at index $a = 0$ as the fixed anchor token $\mathbf{h}_a$ throughout all experiments. For padded backbones [34, 36], this token lies in the padded region and is therefore static by construction; this covers the strongest PMD variants in our experiments (see Table 2). For non-padded backbones [35, 21, 13], we keep the same fixed-index convention and analyze this choice empirically in Appendix C.1. Querying the trained probe $f(\mathbf{h}_i, \mathbf{h}_j) \triangleq \hat{z}_{ij} \in [0, 1]$ against this anchor yields a token-level static score:

$$\hat{p}_i^{\text{stat}} = f(\mathbf{h}_a, \mathbf{h}_i), \qquad a = 0. \tag{7}$$

The dynamic score is then defined as

$$\hat{p}_i^{\text{dyn}} = 1 - \hat{p}_i^{\text{stat}}. \tag{8}$$

The resulting token-level dynamic scores are reshaped to the token grid and bilinearly upsampled to the original image resolution. Since all anchor–target scores can be evaluated in parallel, PMD adds only lightweight computation on top of frozen backbone inference.

4 Experiments

4.1 Experimental Setup

All experiments, including backbone feature extraction, probe training, and evaluation, are conducted on a single consumer-grade NVIDIA RTX 5070 Ti GPU with 16GB VRAM.

Training Protocol. Unless otherwise specified, all probes are trained on 160 videos from the official YouTube-VOS 2018 training split [41] and validated on 80 videos from its validation split. No data from the evaluation benchmarks (DAVIS [26, 27], SegTrackv2 [17], or FBMS-59 [14]) is used for training and validation.

Datasets and Metrics. We evaluate one binary mask per frame on DAVIS [26, 27], SegTrackv2 [17], and FBMS-59 [14], reporting Jaccard mean (J-M) and recall (J-R), the fraction of evaluated frames with Jaccard score above 0.5. DAVIS16 maps valid foreground to dynamic and background to static. DAVIS17/DAVIS-all union all valid object IDs as dynamic, retain valid background as static, and exclude void pixels. SegTrackv2 unions separate foreground instance masks as dynamic and labels background as static. FBMS-59 labels valid pixels differing from the sequence-specific background label as dynamic and evaluates only annotated frames. These annotation-defined targets need not coincide with instantaneous physical motion. Token-wise score maps are bilinearly upsampled and binarized to obtain coarse PMD masks, with optional SAM2 [29] refinement. For token-level probing analysis, we additionally report dynamic-token accuracy ($\mathrm{Acc}_{\mathrm{dyn}}$) and Normalized Gain (NG):

$$\mathrm{Acc}_{\mathrm{dyn}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[\mathbf{1}(p_i \geq 0.5) = \mathbf{1}(r_i \geq \tau)], \qquad \mathrm{NG} = \frac{\mathrm{Acc}_{\mathrm{dyn}} - \mathrm{Baseline}}{1 - \mathrm{Baseline}}.$$

Here, N denotes the total number of evaluated tokens, p_i is the predicted dynamic score, and $r_i \in [0, 1]$ is the ground-truth dynamic pixel ratio within the image patch of token i. We use a threshold of $\tau = 0.35$ to label a token as dynamic. Because static regions dominate our evaluation benchmarks, the baseline predicts all tokens as static; detailed class-imbalance statistics are provided in Appendix C.2.

The selectivity control reports pairwise compatibility accuracy and balanced accuracy. Cross-backbone source-validation analysis reports pairwise normalized gain ($\mathrm{NG}_{\mathrm{pair}}$) against the always-same-state baseline (Appendix G.3). Implementation and thresholding details are in Appendix A.

4.2 Layer-wise Motion Cue Emergence

We perform layer-wise probing on three representative backbones: π^3X [36], Depth Anything 3 (DA3) [21], and DUSt3R [35]. Our probes operate on full block-output token representations. For π^3X and DA3, the selected block groups follow the VGGT-style alternating-attention design, consisting of frame-wise self-attention and global self-attention [34]. We refer to the corresponding frozen block-output token features as *intra-frame* and *inter-frame* features. As shown in Figure 2, motion decodability is widespread across the multi-view geometric modules. The low-rank quadratic probe consistently outperforms the linear and diagonal probes. Motion cues also propagate to intra-frame blocks and prediction heads, but their quality varies across backbones. For subsequent experiments, PMD averages all blocks in one fixed backbone-specific group, using the settings in Table 1 for every target sequence and dataset.

Table 1: Default PMD settings. All probes are trained on a subset of 160 YouTube-VOS [41] videos. For Depth Anything 3 (DA3), we use the DA3-GIANT-1.1 model.

Backbone	Selected Blocks	Probe	Rank k
π^3/π^3X [36]	18 × inter-frame blocks	low-rank quad.	8
DA3 [21]	14 × inter-frame blocks	low-rank quad.	8
DUSt3R [35]	12 × decoder-view-1 + 4 × head-in-view-1 blocks	low-rank quad.	8

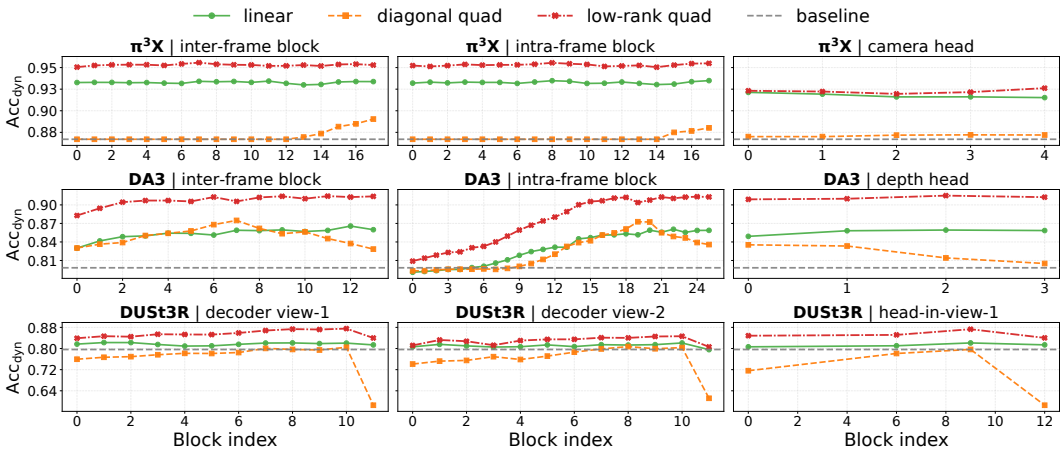


Figure 2: Layer-wise motion decodability on DAVIS-all [27]. Low-rank quadratic probes expose static–dynamic discriminative cues across the multi-view geometric modules of all three backbones, outperforming linear and diagonal probes.

4.3 Zero-shot Cross-Benchmark Motion Segmentation

Table 2: Motion segmentation benchmarking. Cells report J-M/J-R (↑). **First**, second, and third mark the top three distinct reported scores for each dataset and metric across all rows; ties share a color. Non-PMD scores are quoted from the cited papers; PMD scores are evaluated by us. Gray text marks target-benchmark fine-tuning; "–" denotes an unreported score.

Method	Motion-mask supervision; inference components	DAVIS17 J-M / J-R	SegTrackv2 J-M / J-R	FBMS-59 J-M / J-R
Baselines without motion-mask supervision				
VGGT4D [9]	None; VGGT geometry	56.5/65.6	–	–
Easi3R$_{\text{dust3r}}$ [5]	None; DUSt3R geometry	46.9/50.5	–	–
Easi3R$_{\text{dust3r}}$+SAM2	None; DUSt3R geometry, SAM2	60.1/65.3	–	–
Baselines with motion-mask supervision				
GeoMotion [8]	Five datasets; π^3, RAFT, DINOv2	75.4/87.2	–	–
GeoMotion+SAM2	Five datasets; π^3, RAFT, DINOv2, SAM2	81.1/**92.6**	77.3/–	72.5/–
Flow-SAM [39]	Synthetic masks, DAVIS fine-tuning; RAFT, SAM	71.6/–	84.2/–	**88.7**/–
SegAnyMo [10]	Three datasets; tracks, depth, DINOv2, SAM2	–	76.3/–	78.3/–
Ours: low-rank supervised probe				
DUSt3R-PMD	160 YouTube-VOS videos; DUSt3R geometry	62.2/70.0	58.6/77.3	64.0/74.5
DA3-PMD	160 YouTube-VOS videos; DA3 geometry	69.6/85.8	66.3/80.3	69.8/81.4
π^3-PMD	160 YouTube-VOS videos; π^3 geometry	72.7/89.9	72.9/92.9	72.6/83.4
π^3X-PMD	160 YouTube-VOS videos; π^3X geometry	73.7/91.0	72.9/94.0	73.5/88.8
π^3X-PMD+SAM2	160 YouTube-VOS videos; π^3X geometry, SAM2	**81.9**/92.3	**86.4**/**99.3**	83.2/**89.3**

As shown in Table 2, PMD generalizes zero-shot from YouTube-VOS to all three test benchmarks without target-domain fine-tuning. Its multi-view pretrained variants achieve stronger performance than DUSt3R-PMD. In particular, π^3- and π^3X-PMD surpass Easi3R [5] and VGGT4D [9] by a clear margin, while remaining competitive with supervised methods [8, 39, 10]. The following analyses focus on π^3X-PMD, which achieves the strongest overall PMD results. Additional benchmark results, including DAVIS16 and DAVIS-all, are provided in Appendix G.6.

Coarse Masks and Optional Refinement. On DAVIS17, coarse π^3X-PMD reaches 73.7 J-M and 91.0 J-R, compared with 75.4/87.2 for GeoMotion. SAM2 refinement improves π^3X-PMD to 81.9/92.3. The J-R gains are 1.2/1.3 points for π^3/π^3X-PMD, 5.4 for GeoMotion, and 9.9–14.8 for

Easi3R (Table 3). High coarse J-R indicates that most frames already exceed 0.5 Jaccard, while the J-M increase reflects improved average mask overlap after refinement.

Table 3: SAM2 [29] refinement on DAVIS2017 [27]. ΔJ-R is the percentage-point increase in the fraction of frames with Jaccard above 0.5. Additional backbones and methods are reported in Appendices G.2 and G.4.

Method	Flow	w/o SAM2		w/ SAM2		ΔJ-R ↓
		J-M ↑	J-R ↑	J-M ↑	J-R ↑	
Easi3R$_{\text{dust3r}}$ [5]	×	46.9	50.5	60.1	65.3	14.8
Easi3R$_{\text{monst3r}}$ [5]	×	54.8	66.2	67.9	76.1	9.9
GeoMotion [8]	✓	**75.4**	87.2	81.1	**92.6**	5.4
π^3-PMD (ours)	×	72.7	89.9	80.2	91.1	**1.2**
π^3X-PMD (ours)	×	73.7	**91.0**	**81.9**	92.3	1.3

Backbone-dependent Recoverability. Performance also varies among multi-view backbones: the π^3X–DA3 gap on DAVIS17 is 4.1 J-M for coarse masks and 4.2 points after SAM2 refinement. Source-validation pairwise decodability is positively associated with mean target coarse J-M across six backbones. Appendix G.3 reports the block-level statistics, correlations, and broad design regimes.

Generalization to Unseen Semantic Categories. Figure 3 shows selected examples from DAVIS and SegTrackv2 whose semantic categories are absent from the 160-video YouTube-VOS [41] training subset. PMD produces cleaner dynamic score maps and more coherent coarse masks than Easi3R [5] and VGGT4D [9], illustrating transfer to categories unseen during probe training.

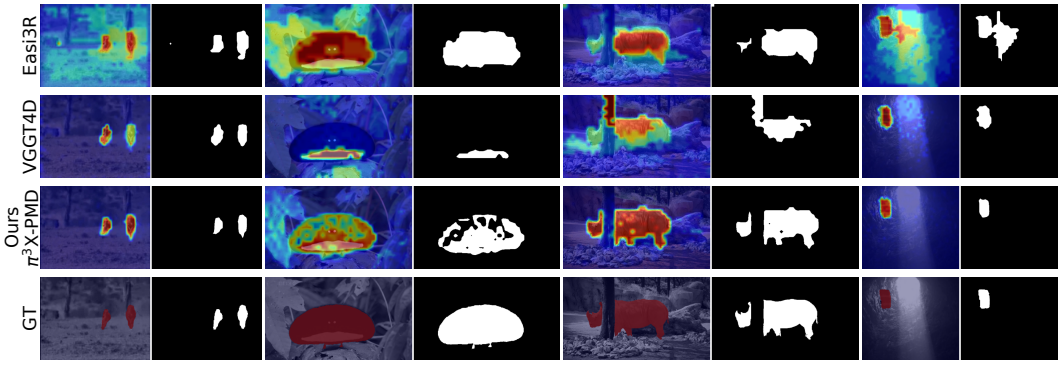


Figure 3: Qualitative comparison on DAVIS [26] and SegTrackv2 [17] against training-free methods based on feed-forward 3D foundation models. The semantic classes shown here are absent from the 160-video training subset of π^3X-PMD. All binary masks shown here are generated without SAM2 [29] refinement. PMD produces cleaner dynamic score maps and more coherent coarse masks.

Efficiency. Table 4 compares training data, trainable capacity, and inference overhead. Trained on only 160 videos, π^3X-PMD uses a 0.295M-parameter low-rank probe and adds 1.62G FLOPs per frame for scoring and aggregation. On a single RTX 5070 Ti, coarse PMD adds 4.2 ms/frame (0.87%) to frozen-backbone inference. End-to-end latency and peak VRAM, including the backbone and optional SAM2 refinement, are reported in Appendix E.

4.4 Compact Static–Dynamic Discriminative Subspace and Parsimonious Decoding

The strong zero-shot performance of π^3X-PMD, despite its small probe capacity and limited training data, suggests that motion cues in its frozen token representations are structured enough to be projected into a compact static–dynamic discriminative subspace.

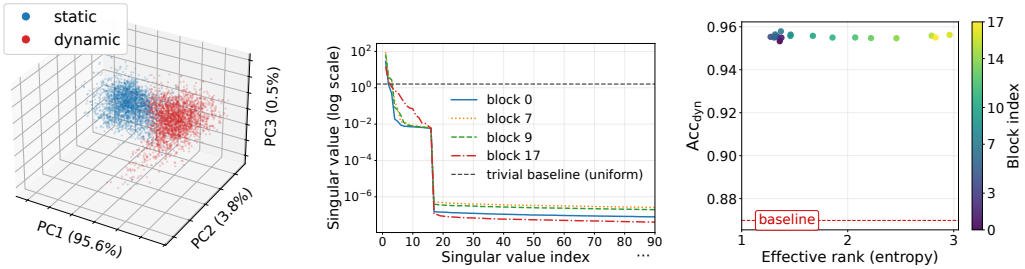
Figure 4 provides a qualitative and quantitative analysis of this probe-induced subspace. In Figure 4a, a 3D PCA visualization of the low-rank quadratic probe representation ($k = 8$) exhibits a clear

Table 4: Comparison of data efficiency, performance, and inference overhead. Mask scores use the complete pipelines, including SAM/SAM2 where applicable; PMD and GeoMotion scores include SAM2. Trainable Params count optimized parameters only. Added FLOPs exclude SAM/SAM2 and, for PMD, frozen-backbone inference; end-to-end latency and measurement details are reported in Appendix E.

Method	Training Data	DAVIS17 (J-M ↑)	SegTrackv2 (J-M ↑)	FBMS-59 (J-M ↑)	Trainable Params (M) ↓	Added FLOPs (G) ↓
Large-scale supervised						
SegAnyMo [10]	Kubric + Dynamic Replica + HOI4D + etc.	–	76.3	78.3	3.44	>2.08
Flow-SAM [39]	Synthetic pretraining + DAVIS fine tuning + etc.	71.6	84.2	**88.7**	9.83	$>3.63 \times 10^4$
GeoMotion [8]	GOT-Motion + HOI4D + Dynamic Replica + etc.	81.1	77.3	72.5	80.3	$>3.09 \times 10^2$
Ours (data-efficient probing)						
π^3-PMD	160 YouTube-VOS 2018 videos	80.2	84.5	81.1	**0.295**	**1.62**
π^3X-PMD	160 YouTube-VOS 2018 videos	**81.9**	**86.4**	83.2	**0.295**	**1.62**

separation between static and dynamic tokens. Figure 4b further shows rapid spectral decay, indicating that the probe-induced motion cues are concentrated in a small number of dominant directions. This spectral concentration is also reflected in Figure 4c, where strong performance is retained even when the probe-induced motion cues have low effective rank, particularly in shallow layers.

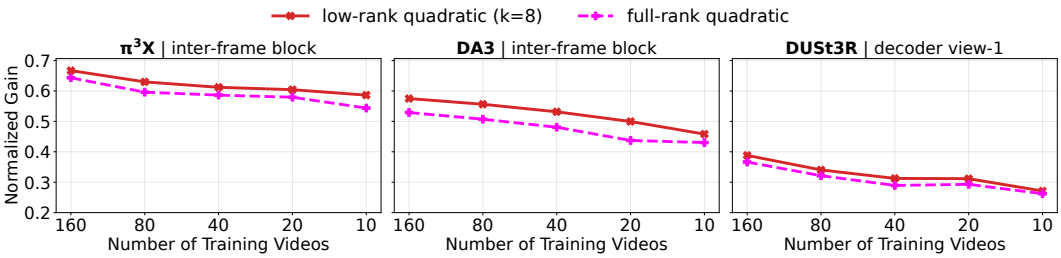
We next test whether the compact motion cues remain stable under reduced scene diversity. Keeping the number of sampled token pairs fixed, we progressively reduce the number of training videos. As shown in Figure 5a, normalized gain degrades gradually, while the low-rank probe ($k = 8$) consistently matches or exceeds the full-rank probe. This supports low rank as an effective inductive bias for the supervised readout. Figure 5b shows a similar trend at the mask level: downstream performance decreases gradually, both before and after SAM2 [29] refinement.



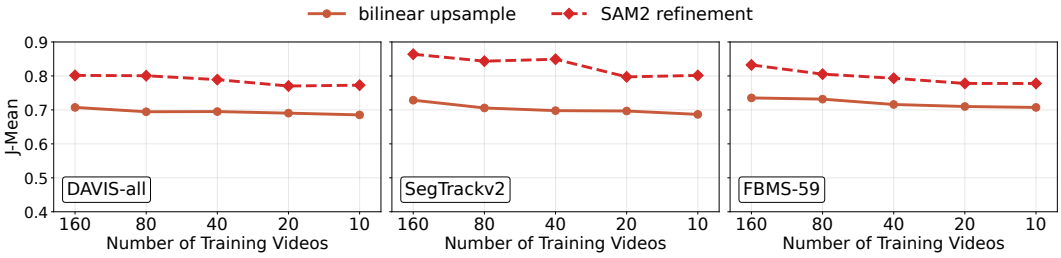
(a) (b) (c)

Figure 4: Low-rank structure of the probe-induced static–dynamic discriminative representation on π^3X-PMD, illustrated for DAVIS-all [27]. (a) Principal Component Analysis (PCA) illustrates clear separation between static and dynamic tokens. (b) The singular spectrum decays steeply in the leading components, especially in earlier layers, revealing strong rank concentration. (c) Performance saturates at a very low effective rank, indicating that the probe-induced motion cues are compact.

Probe Selectivity. To test whether the probe relies on token–motion-state correspondence, we shuffle labels within each frame group and retrain the same rank-8 π^3X probe, preserving frozen



(a) Token-level probing (Normalized Gain) on DAVIS-all [27].



(b) Downstream mask prediction performance (J-M) before and after SAM2 [29] refinement.

Figure 5: Scene-diversity starvation analysis. (a) Low-rank probing ($k = 8$) matches or exceeds full-rank probing under reduced training data, indicating a compact static–dynamic discriminative structure. (b) Downstream mask quality degrades gradually rather than collapsing.

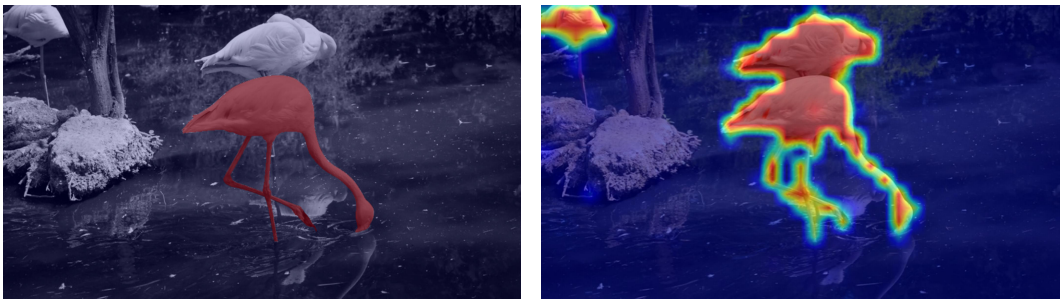
features and label counts. On YouTube-VOS validation, 84.47% of pairs share the same motion state, so always predicting "same state" already attains this accuracy. Balanced accuracy averages recall for same-state and different-state pairs equally; this constant predictor scores 50%. Averaged over 18 inter-frame blocks, shuffling lowers pair accuracy from 95.72% to 84.48% and balanced accuracy from 91.67% to 50.34% (Table 5). The return to baseline indicates that the gains rely on the correct token–label correspondence. Details are in Appendix D.3.

Table 5: Selectivity of low-rank quadratic probes on frozen π^3X tokens. Gain is the percentage-point improvement in pair accuracy over the always-same-state baseline.

Probe / labels	Params	Pair accuracy (%)	Gain (pp)	Balanced accuracy (%)
$k = 4$, correct	0.147M	95.70	+11.23	91.60
$k = 8$, correct	0.295M	95.72	+11.25	91.67
$k = 8$, shuffled	0.295M	84.48	+0.01	50.34
Always-same baseline	0	84.47	0.00	50.00

5 Limitations

While PMD shows strong cross-benchmark generalization and data efficiency, its spatial precision is limited by ViT token granularity. PMD makes binary static–dynamic decisions and does not distinguish fine-grained motion types such as in-place articulation, local deformation, or object displacement. In some cases, PMD also exhibits semantic leakage (Figure 6): static instances may receive elevated dynamic scores when another instance of the same category is moving, even for categories absent from the 160-video training subset. Object-centric YouTube-VOS supervision and frozen semantic features may contribute to this behavior. Examples are provided in Appendix H.2. Future work could probe additional physical quantities, such as displacement and velocity, from frozen 3D foundation models to distinguish different motion modes.



(a) Ground truth (b) PMD dynamic score map

Figure 6: Semantic leakage in π^3X-PMD. The two static background flamingos receive elevated dynamic scores alongside the moving foreground instance, illustrating imperfect separation of motion and semantic cues.

6 Conclusion

In this work, we show that frozen feed-forward 3D foundation models already contain motion cues that can be decoded without high-capacity motion decoders. Using π^3X [36], DA3 [21], and DUSt3R [35] as representative backbones, we find that lightweight low-rank quadratic probes can expose static–dynamic discriminative cues from frozen token representations. For π^3X [36], our rank-restricted and spectral analyses further show that these cues are concentrated in a compact low-dimensional discriminative subspace. When trained on only 160 YouTube-VOS [41] videos, our Probing-based Motion Distillation (PMD) transfers zero-shot to DAVIS [26, 27], SegTrackv2 [17], and FBMS-59 [14] while remaining competitive with substantially heavier methods.

These results suggest that motion understanding in multi-view pretrained 3D foundation models can be treated as a lightweight decoding problem of frozen tokens rather than re-learning motion-specific features through large-scale training. This perspective motivates probing emerging capabilities of 3D foundation models before resorting to large decoders, extensive supervision, or task-specific adaptation.

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers and the area chair for their constructive feedback.

Funding statement. This research received no external funding.

Competing interests. The authors declare no competing interests.

References

[1] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *International Conference on Learning Representations*, 2017.

[2] Nikita Araslanov, Anna Ribic, and Daniel Cremers. FlowFeat: Pixel-dense embedding of motion profiles. In *Advances in Neural Information Processing Systems*, 2025. NeurIPS 2025 Spotlight.

[3] Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

[4] Jelena Bratulić, Sudhanshu Mittal, Thomas Brox, and Christian Rupprecht. On geometric understanding and learned priors in feed-forward 3D reconstruction models. *arXiv preprint arXiv:2512.11508*, 2025. URL `https://arxiv.org/abs/2512.11508`. [v2] updated March 17, 2026.

[5] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Easi3R: Estimating disentangled motion from DUSt3R without training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9158–9168, 2025.

[6] Xianze Fang, Jingnan Gao, Zhe Wang, Zhuo Chen, Xingyu Ren, Jiangjing Lyu, Qiao-Mu Ren, Zhonglei Yang, Xiaokang Yang, Yichao Yan, and Chengfei Lv. Dens3R: A foundation model for 3D geometry prediction. In *International Conference on Learning Representations*, 2026. ICLR 2026.

[7] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J. Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3749–3761, 2022.

[8] Xiankang He, Peile Lin, Ying Cui, Dongyan Guo, Chunhua Shen, and Xiaoqin Zhang. GeoMotion: Rethinking motion segmentation via latent 4D geometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[9] Yu Hu, Chong Cheng, Sicheng Yu, Xiaoyang Guo, and Hao Wang. VGGT4D: Mining motion cues in visual geometry transformers for 4D scene reconstruction, 2025. URL `https://arxiv.org/abs/2511.19971`.

[10] Nan Huang, Wenzhao Zheng, Chenfeng Xu, Kurt Keutzer, Shanghang Zhang, Angjoo Kanazawa, and Qianqian Wang. Segment any motion in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3406–3416, 2025.

[11] Junhwa Hur, Charles Herrmann, Songyou Peng, Philipp Henzler, Zeyu Ma, Todd Zickler, and Deqing Sun. UFO-4D: Unposed feedforward 4D reconstruction from two images. In *International Conference on Learning Representations*, 2026. ICLR 2026.

[12] Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, et al. AnySplat: Feed-forward 3D gaussian splatting from unconstrained views. *ACM Transactions on Graphics*, 44(6):1–16, 2025.

[13] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal feed-forward metric 3D reconstruction. In *International Conference on 3D Vision*, 2026.

[14] Margret Keuper, Bjoern Andres, and Thomas Brox. Motion trajectory segmentation via minimum cost multicuts. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[15] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[16] Yeong Jun Koh and Chang-Su Kim. Primary object segmentation in videos based on region augmentation and reduction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2638–2647, 2017. URL `https://openaccess.thecvf.com/content_cvpr_2017/papers/Koh_Primary_Object_Segmentation_CVPR_2017_paper.pdf`.

[17] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M. Rehg. Video segmentation by tracking many figure-ground segments. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2192–2199, 2013. doi: 10.1109/ICCV.2013.273.

[18] Yihao Li, Saeed Salehi, Lyle Ungar, and Konrad Kording. Does object binding naturally emerge in large pretrained vision transformers? In *Advances in Neural Information Processing Systems*, 2025.

[19] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021.

[20] Long Lian, Zhirong Wu, and Stella X. Yu. Bootstrapping objectness from videos by relaxed common fate and visual grouping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14582–14591, 2023.

[21] Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. In *International Conference on Learning Representations*, 2026. ICLR 2026 Oral.

[22] Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3618–3627, 2019. doi: 10.1109/CVPR.2019.00374.

[23] Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorld-Model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

[24] Etienne Meunier, Anaïs Badoual, and Patrick Bouthemy. EM-driven unsupervised learning for efficient motion segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4462–4473, 2023. doi: 10.1109/TPAMI.2022.3198480.

[25] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, 2018.

[26] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. DAVIS: A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[27] 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.

[28] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021.

[29] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL `https://arxiv.org/abs/2408.00714`.

[30] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. doi: 10.1109/CVPR.2016.445.

[31] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[32] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurélien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020.

[33] Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press, 2019.

[34] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5294–5306, 2025.

[35] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jérôme Revaud. DUSt3R: Geometric 3D vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.

[36] 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. In *International Conference on Learning Representations*, 2026. ICLR 2026 Poster.

[37] Junyu Xie, Weidi Xie, and Andrew Zisserman. Segmenting moving objects via an object-centric layered representation. In *Advances in Neural Information Processing Systems*, volume 35, pages 28023–28036, 2022.

[38] Junyu Xie, Weidi Xie, and Andrew Zisserman. Appearance-based refinement for object-centric motion segmentation. In *European Conference on Computer Vision*, pages 238–256, 2024.

[39] Junyu Xie, Charig Yang, Weidi Xie, and Andrew Zisserman. Moving object segmentation: All you need is SAM (and flow). In *Proceedings of the Asian Conference on Computer Vision*, pages 162–178, 2024.

[40] Kai Xu, Tze Ho Elden Tse, Jizong Peng, and Angela Yao. DAS3R: Dynamics-aware gaussian splatting for static scene reconstruction. *arXiv preprint arXiv:2412.19584*, 2024.

[41] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. YouTube-VOS: Sequence-to-sequence video object segmentation. In *European Conference on Computer Vision*, pages 585–601, 2018.

[42] Shu Yang, Lu Zhang, Jinqing Qi, Huchuan Lu, Shuo Wang, and Xiaoxing Zhang. Learning motion-appearance co-attention for zero-shot video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1544–1553, 2021. doi: 10.1109/ICCV48922.2021.00159.

[43] Yanchao Yang, Antonio Loquercio, Davide Scaramuzza, and Stefano Soatto. Unsupervised moving object detection via contextual information separation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 879–888, 2019.

[44] Yanchao Yang, Brian Lai, and Stefano Soatto. DyStaB: Unsupervised object segmentation via dynamic-static bootstrapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2826–2836, 2021.

[45] Chuhan Zhang, Guillaume Le Moing, Skanda Koppula, Ignacio Rocco, Liliane Momeni, Junyu Xie, Shuyang Sun, Rahul Sukthankar, Joëlle K. Barral, Raia Hadsell, Zoubin Ghahramani, Andrew Zisserman, Junlin Zhang, and Mehdi S. M. Sajjadi. Efficiently reconstructing dynamic scenes one D4RT at a time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. Oral.

[46] 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. In *International Conference on Learning Representations*, 2025. ICLR 2025 Spotlight.

[47] Kaichen Zhou, Yuhan Wang, Grace Chen, Gaspard Beaudouin, Fangneng Zhan, Paul Pu Liang, and Mengyu Wang. PAGE-4D: Disentangled pose and geometry estimation for VGGT-4D perception. In *International Conference on Learning Representations*, 2026. ICLR 2026 Poster.

[48] Tianfei Zhou, Jianwu Li, Shunzhou Wang, Ran Tao, and Jianbing Shen. MATNet: Motion-attentive transition network for zero-shot video object segmentation. *IEEE Transactions on Image Processing*, 29:8326–8338, 2020. doi: 10.1109/TIP.2020.3013162.

[49] Yang Zhou, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Haoyu Guo, Zizun Li, Kaijing Ma, Xinyue Li, Yating Wang, Haoyi Zhu, Mingyu Liu, Dingning Liu, Jiange Yang, Zhoujie Fu, Junyi Chen, Chunhua Shen, Jiangmiao Pang, Kaipeng Zhang, and Tong He. OmniWorld: A multi-domain and multi-modal dataset for 4D world modeling. In *International Conference on Learning Representations*, 2026.

A Implementation Details

Hardware. All experiments were run on a single workstation with an NVIDIA GeForce RTX 5070 Ti GPU (16 GB memory), an AMD Ryzen 7 7700X CPU, and 64 GB of system memory. Probe training, motion-mask extraction, and SAM2 [29] refinement were performed on one GPU at a time; no distributed training was used.

Training and Validation Data. Unless otherwise specified, probes are trained on 160 videos sampled from the official YouTube-VOS 2018 [41] training split and validated on 80 videos sampled from its validation split. No videos or annotations from DAVIS [26, 27], SegTrackv2 [17], or FBMS-59 [14] are used to optimize probe training parameters.

Training Hyperparameters. We optimize the probe weights using AdamW with a constant learning rate of 1×10^{-3}. We do not use weight decay or learning-rate scheduling. All probes are trained for 16 epochs. For each epoch, we sample 200K token pairs from within the same exported token group, i.e., from tokens jointly sampled from the same target frame.

Inference Context. At inference time, PMD is applied to target-frame tokens extracted from a multi-frame input group. The surrounding frames provide temporal or multi-view context to the frozen backbone, while probing and mask decoding are performed on the target frame. In all the motion-mask extraction experiments, we use a 3-frame context with stride 2.

Mask Extraction. After obtaining token-wise dynamic score maps, we bilinearly upsample them to image resolution. Coarse PMD uses Otsu thresholding with an offset of 0.2. PMD+SAM2 constructs a prompt mask with the same threshold, erodes it for two iterations only during prompt construction, and filters connected components by their mean dynamic score using Otsu with an offset of 0.3 before SAM2 [29] refinement. These extra operations belong to the optional refinement pipeline.

Threshold Robustness. Table 6 evaluates nine binarization rules without SAM2. Each rule is applied unchanged across all sequences and benchmarks, without sequence- or dataset-specific selection. For fixed thresholds 0.40–0.60 and Otsu offsets 0.0–0.2, the across-rule standard deviation of mean J-M is 0.69 and 0.97, respectively. Otsu+0.1 attains a higher mean than the Otsu+0.2 default (73.0 versus 72.4); thus, Otsu+0.2 is not uniquely privileged. We retain it as the common default. Extreme thresholds reduce performance, and benchmark averages do not separately establish robustness to illumination or clutter.

Table 6: Threshold robustness of coarse π^3X-PMD without SAM2. Mean and Std. summarize J-M across the three benchmarks. The last two rows instead report across-rule standard deviations for each benchmark and for the three-benchmark mean.

Rule	DAVIS-all	SegTrackv2	FBMS-59	Mean	Std.
Fixed 0.30	64.5	69.4	66.6	66.8	1.98
Fixed 0.40	68.1	73.3	71.1	70.8	2.15
Fixed 0.50	69.9	73.3	73.9	72.4	1.75
Fixed 0.60	69.7	69.4	**74.2**	71.1	2.20
Fixed 0.70	66.5	61.6	71.2	66.4	3.93
Otsu	69.1	72.2	70.8	70.7	1.28
Otsu + 0.1	**71.2**	**74.5**	73.2	**73.0**	1.34
Otsu + 0.2 (default)	70.7	72.9	73.5	72.4	**1.17**
Otsu + 0.3	67.5	66.2	70.8	68.2	1.93
Across-rule std., Fixed 0.40–0.60	0.81	1.84	1.40	0.69	–
Across-rule std., Otsu + 0.0–0.2	0.90	0.96	1.21	0.97	–

B PMD Inference Algorithm

Algorithm 1 PMD Inference

Require: Frozen backbone F, input frame group I, selected blocks $\mathcal{L}$, trained pairwise probe $f(\cdot,\cdot)$, static anchor $a = 0$

Ensure: Dynamic score map $\hat{\mathbf{P}}^{\mathrm{dyn}}$, coarse mask $\hat{\mathbf{M}}_{\mathrm{coarse}}$, optional refined mask $\hat{\mathbf{M}}_{\mathrm{SAM2}}$

Extract target-frame token features $\mathbf{H}^{(\ell)} = \{\mathbf{h}_0^{(\ell)}, \ldots, \mathbf{h}_N^{(\ell)}\}$ for all $\ell \in \mathcal{L}$.
for each selected block $\ell \in \mathcal{L}$ **in parallel do**
    Compute $\hat{p}_{i,\ell}^{\mathrm{stat}} = f(\mathbf{h}_a^{(\ell)}, \mathbf{h}_i^{(\ell)})$ for all target-frame tokens i in parallel.
    Convert static probability to dynamic probability via $\hat{p}_{i,\ell}^{\mathrm{dyn}} = 1 - \hat{p}_{i,\ell}^{\mathrm{stat}}$.
end for
Average dynamic scores across blocks.
Reshape the averaged token scores to the token grid and bilinearly upsample them to obtain $\hat{\mathbf{P}}^{\mathrm{dyn}}$.
Obtain $\hat{\mathbf{M}}_{\mathrm{coarse}}$ using Otsu thresholding with offset 0.2.
if SAM2 refinement is enabled **then**
    Construct a coarse prompt mask using Otsu thresholding with offset 0.2.
    Erode the prompt mask for two iterations.
    Keep connected components whose mean dynamic score passes Otsu thresholding with offset 0.3.
    Use the surviving components as prompts for SAM2 to obtain $\hat{\mathbf{M}}_{\mathrm{SAM2}}$.
end if
return $\hat{\mathbf{P}}^{\mathrm{dyn}}$, $\hat{\mathbf{M}}_{\mathrm{coarse}}$, and optionally $\hat{\mathbf{M}}_{\mathrm{SAM2}}$

C Dataset Statistics and Fixed-Reference Diagnostics

C.1 Fixed Reference-Token Convention

PMD uses the first token $a = 0$ as a fixed reference token throughout all experiments. This choice is fixed a priori and is never selected or adapted per image, video, or benchmark. For padded backbones, including the strongest π^3/π^3X variants in our experiments, this token lies in the padded region and is therefore static by construction. For non-padded backbones such as DA3 [21] and DUSt3R [35], we keep the same fixed-index convention. Table 7 shows that this fixed upper-left reference location is static in the vast majority of frames on the training source. These statistics are used only as a post-hoc sanity check and are not used for anchor selection, threshold tuning, model selection, or benchmark-specific adaptation.

Figure 7 further stress-tests this fixed-reference convention qualitatively on the two non-padded backbones used in our main analysis. We compare a standard case where the fixed anchor patch is static with a rare violation where the same patch is fully dynamic under the ground-truth mask. In the violation case, the PMD score maps show a moderately elevated background response, but the moving foreground remains separable and is recovered by both the coarse and refined masks. This suggests that, for the non-padded backbones evaluated here, PMD is not determined solely by the instantaneous ground-truth state of the reference patch; the fixed spatial index may still provide a useful static-reference prior, likely supported by positional encoding and the highly static distribution of the upper-left patch. Since these two non-padded backbones are included in the main experiments and the stress-test violation does not qualitatively overturn their predictions, we use the same fixed $a = 0$ protocol for all reported results without benchmark-specific anchor correction.

Camera Motion and Uniform Motion States. In a fully static scene under camera motion, a uniform near-zero dynamic map is the intended output: image motion alone does not imply independently moving foreground. If all scene tokens share the same motion state, pairwise same-state labels do not distinguish all-static content from uniformly moving content without an independent reference. Under this formulation both admit a no-independent-foreground interpretation; the output is not an estimate of absolute scene motion. No dedicated evaluation on all-static camera-motion datasets is reported here.

Our implementation also provides an optional safeguard that pins $a = 0$ to the static label and explicitly includes anchor–token pairs involving $a = 0$ in each training batch. This safeguard is disabled for all results reported in this paper and is provided only for robustness checks or for applying PMD to datasets where the fixed-anchor convention may not hold.

Table 7: Fixed-reference staticness diagnostic on YouTube-VOS 2018 [41]. The last column reports the percentage of frames in which the upper-left 14×14 patch corresponding to the fixed reference token is static under the projected-mask labeling rule.

Split	# Seq.	# GT Frames	Static Pixels (%)	Static Upper-left 14×14 Patch (%)
Full official split				
Train	2,985	89,750	82.75	97.03
Valid	492	16,563	83.08	96.61
Our subset				
Train	160	4,933	87.03	97.73
Valid	80	2,893	84.79	96.58

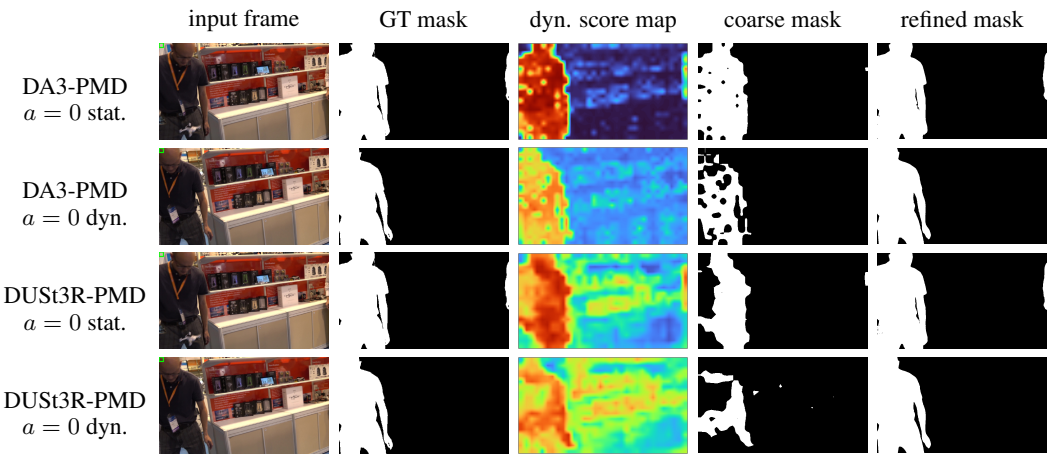


Figure 7: Stress test of the fixed anchor token $a = 0$ convention on a DAVIS-all [27] evaluation example for the two non-padded backbones in the main analysis. The upper-left reference patch is marked in green in the input frame. For each backbone, we compare a standard case where the fixed $a = 0$ patch is static with a rare violation where the same patch is fully dynamic according to the ground-truth mask. For both DA3-PMD and DUSt3R-PMD, the violation moderately increases the background response, but it does not invert the prediction: the moving foreground remains separable and is recovered by both the coarse and refined masks.

C.2 Token-Level Class Imbalance.

In addition to the fixed-anchor diagnostic above, we report token-level static–dynamic imbalance on the evaluation benchmarks in Table 8. These statistics are used only to contextualize the all-static baseline and the normalized-gain metric in the probing analysis; they are not used for anchor token selection, threshold tuning, or model training.

D Capacity Analysis of Low-Rank Quadratic Probes

While the main text empirically demonstrates the compactness of the decoded motion representation and PMD's data efficiency, this appendix provides a capacity-based interpretation of why PMD can generalize from a sparse subset of token pairs and only 160 training videos.

Table 8: Representative token-level class imbalance on the evaluation benchmarks. Static tokens dominate across all test sets. Ratios are computed from the π^3X [36] tokenization of the probed features; normalized-gain baselines are computed separately for each backbone using its own tokenization.

Dataset	# Valid Tokens	Static Tokens (%)	Dynamic Tokens (%)
DAVIS-all [27]	392,512	92.0	8.0
SegTrackv2 [17]	68,224	95.3	4.7
FBMS-59 [14]	46,080	88.4	11.6

D.1 Pairwise Low-Rank Probe Class

For a token pair (i, j), PMD predicts the pair label z_{ij} using the low-rank quadratic probe:

$$\hat{z}_{ij}^{(k)} = \sigma\left(s_{ij}^{(k)}\right), \tag{9}$$

with score

$$s_{ij}^{(k)} = \frac{1}{2\sqrt{d}}\, \mathbf{h}_i^{(\ell)\top} \left(\mathbf{U}^{(k)\top}\mathbf{V}^{(k)} + \mathbf{V}^{(k)\top}\mathbf{U}^{(k)}\right) \mathbf{h}_j^{(\ell)} + b, \tag{10}$$

where $\mathbf{U}^{(k)}, \mathbf{V}^{(k)} \in \mathbb{R}^{k\times d}$, $b \in \mathbb{R}$, and k is the rank hyperparameter. Due to the symmetrization, the resulting bilinear matrix has rank at most $2k$, but its degrees of freedom still scale linearly with kd.

D.2 Capacity Scaling

The default π^3X-PMD configuration averages probe outputs from $L_{\text{agg}} = 18$ selected inter-frame blocks, with one low-rank probe per block. Therefore, the learnable factor matrices contribute

$$P_{\text{PMD}}(k) = L_{\text{agg}}(2kd + 1) = 18(2kd + 1) \tag{11}$$

parameters. An unconstrained full-rank quadratic probe over the same aggregated blocks requires

$$P_{\text{full}} = L_{\text{agg}}(d^2 + 1) = 18(d^2 + 1). \tag{12}$$

For $d = 1024$, this yields

$$P_{\text{full}} = 18(1024^2 + 1) = 18{,}874{,}386. \tag{13}$$

Table 9: Validation-based rank selection for PMD. Parameter counts are reported across candidate rank hyperparameters k, compared with the implemented full-rank quadratic probe at $d = 1024$ and $L_{\text{agg}} = 18$ aggregated blocks. For π^3X-PMD, we report the best token-level Accuracy and Normalized Gain on the shared YouTube-VOS validation subset (80 sequences) over all inter-frame blocks. This validation subset, not any target evaluation benchmark, is used to select the default rank $k = 8$.

Rank k	$P_{\text{PMD}}(k)$	$P_{\text{PMD}}(k)/P_{\text{full}}$	Accuracy (%) ↑	NG $\times 10^2$ ↑
4	147,474	0.78%	96.10	60.60
8	294,930	1.56%	96.12	60.77
16	589,842	3.12%	96.10	60.55
32	1,179,666	6.25%	96.11	60.63
64	2,359,314	12.50%	96.08	60.38
128	4,718,610	25.00%	96.08	60.38

Table 9 reports the validation-based rank-selection analysis on the shared YouTube-VOS validation subset. We use this validation-only analysis to choose the default rank, avoiding any target-benchmark-specific rank selection. In this validation setting, $k = 4$ already recovers most of the attainable performance, while $k = 8$ gives the best Accuracy and Normalized Gain among the tested ranks. We therefore adopt $k = 8$ as the default rank for all main experiments.

Table 10: Target-benchmark diagnostic rank sweep on DAVIS-all [27]. Parameter counts are reported across candidate rank hyperparameters k, compared with the implemented full-rank quadratic probe at $d = 1024$ and $L_{\text{agg}} = 18$ aggregated blocks. For π^3X-PMD, we report the best token-level Accuracy and Normalized Gain over all inter-frame blocks. This analysis is used only to diagnose whether the compact-rank behavior transfers to DAVIS-all, not for selecting k.

Rank k	$P_{\text{PMD}}(k)$	$P_{\text{PMD}}(k)/P_{\text{full}}$	Accuracy (%) ↑	NG $\times 10^2$ ↑
4	147,474	0.78%	95.54	66.48
8	294,930	1.56%	95.57	66.70
16	589,842	3.12%	95.54	66.49
32	1,179,666	6.25%	95.60	66.95
64	2,359,314	12.50%	95.48	66.01
128	4,718,610	25.00%	95.36	65.09
256	9,437,202	50.00%	95.30	64.66

D.3 Shuffled-label Selectivity Control

To test selectivity to token–label correspondence, we permute motion labels within each frame group while preserving its label counts and retrain the same rank-8 π^3X probe on identical frozen features. Table 5 reports pairwise compatibility accuracy and balanced accuracy on YouTube-VOS validation, averaged over the 18 inter-frame blocks. Balanced accuracy averages recall for the same-state and different-state pair classes. Parameter counts include all block-specific probes.

The shuffled-label control is close to the always-same baseline and chance balanced accuracy, whereas the correctly labeled probes retain gains above 11 percentage points at both capacities. This supports recoverability that depends on the correct token–motion-state correspondence under the restricted probe. It does not separate physical motion from correlated semantic or objectness information, nor establish an intrinsic low-dimensional subspace of raw tokens. The compactness analysis concerns the supervised probe-induced representation.

D.4 Sample-Complexity Intuition

The parameter counts above should be interpreted as a degree-of-freedom proxy rather than a formal generalization bound, especially because our video-derived token pairs are highly correlated. Under standard bounded-feature assumptions, capacity-based generalization bounds often scale with the effective degrees of freedom, up to constants, logarithmic factors, and assumptions on the data distribution [31, 25, 33]. A full-rank quadratic probe has capacity on the order of

$$P_{\text{full}} = \mathcal{O}(L_{\text{agg}} d^2),$$

whereas the low-rank bilinear probe has capacity on the order of

$$P_{\text{PMD}}(k) = \mathcal{O}(L_{\text{agg}} k d).$$

To put these complexity classes into perspective, let N denote the total number of sampled pairwise training examples. Our default pipeline samples 2×10^5 token pairs per epoch for 16 epochs (Appendix A), yielding $N \approx 3.2 \times 10^6$ highly correlated pairs derived from 160 videos. For the full-rank probe, $P_{\text{full}} \approx 1.88 \times 10^7$, which is substantially larger than N. This places the unconstrained quadratic probe in a high-dimensional regime that can be prone to overfitting, especially given the correlation among video-derived token pairs.

By contrast, at the default rank $k = 8$, PMD has $P_{\text{PMD}}(8) \approx 2.95 \times 10^5$ trainable parameters, well below the number of sampled pairs. This reduction from the $\mathcal{O}(d^2)$ regime to the $\mathcal{O}(kd)$ regime provides a capacity-based explanation for why low-rank quadratic PMD can generalize efficiently from limited video diversity. This interpretation is consistent with the empirical rank ablations in Table 10.

D.5 Comparison to GeoMotion Decoder Capacity

To complement the degree-of-freedom analysis above, we compare PMD with GeoMotion [8], a representative learned motion decoder built on top of a feed-forward 3D reconstruction model.

This comparison is intended as a capacity and parameter-count comparison, not as an architecture-independent sample-complexity bound.

For transformer-style neural decoders with depth L and width d_{dec}, the dominant parameter term scales on the order of Ld_{dec}^2. GeoMotion uses a motion decoder with $d_{\mathrm{dec}} = 1024$ and $L = 5$, whereas PMD uses one low-rank bilinear probe per aggregated block:

$$P_{\mathrm{PMD}}(k) = L_{\mathrm{agg}}(2kd + 1) = 18(2kd + 1), \tag{14}$$

with $L_{\mathrm{agg}} = 18$ and $d = 1024$.

We further compare against the actual trainable motion branch used by GeoMotion. This branch includes the confidence decoder, confidence head, and motion-aware decoder, with total parameter count

$$P_{\mathrm{GeoMotion}} = 80{,}320{,}196. \tag{15}$$

Table 11: Comparison between PMD and the trainable GeoMotion [8] motion head. The second column reports the concrete parameter ratio. The third column reports a simple degree-of-freedom proxy $L_{\mathrm{agg}}kd/(Ld_{\mathrm{dec}}^2)$, ignoring constant factors, with $L_{\mathrm{agg}} = 18$, $d = d_{\mathrm{dec}} = 1024$, and $L = 5$; it is not a formal sample-complexity bound.

PMD rank k	$P_{\mathrm{GeoMotion}}/P_{\mathrm{PMD}}(k)$	$L_{\mathrm{agg}}kd/(Ld_{\mathrm{dec}}^2)\downarrow$
4	5.45×10^{2}	1.41×10^{-2}
8	2.72×10^{2}	2.81×10^{-2}
16	1.36×10^{2}	5.63×10^{-2}
32	6.81×10^{1}	1.13×10^{-1}
64	3.40×10^{1}	2.25×10^{-1}
128	1.70×10^{1}	4.50×10^{-1}
256	8.51	9.00×10^{-1}

Thus, even at the largest candidate rank $k = 256$, PMD remains nearly one order of magnitude smaller than the trainable GeoMotion motion head, while in the practically relevant low-rank regime ($k \leq 16$) the concrete parameter gap remains above two orders of magnitude.

E Computational Efficiency and Inference Cost

To further highlight the lightweight and plug-and-play nature of PMD, we analyze its inference-time overhead. A major bottleneck in existing dynamic scene understanding pipelines is reliance on heavy neural decoders, iterative optimization, or auxiliary motion-estimation modules, which substantially increase memory footprint and computational cost. Table 12 summarizes the resulting trainable-parameter and inference-cost comparison.

Measurement Protocol. In Tables 4 and 12, all added FLOPs are normalized per original video frame. For PMD, the reported Added FLOPs are computed directly from the current implementation and count only PMD-specific scoring and averaging beyond the frozen backbone. Concretely, they include probe scoring, pair averaging, and block averaging, while excluding backbone inference. For external methods, when full end-to-end inference costs cannot be cleanly isolated from the public implementation, we report conservative lower bounds under the official default inference configuration, denoted by $>$. Trainable Params denote only the parameters optimized by the released training pipeline.

Why Some External Pipelines Are Expensive at Inference. A small trainable decoder does not necessarily imply low inference cost. In particular, Flow-SAM [39] combines optical-flow preprocessing with repeated prompt-conditioned decoding over a dense uniform grid of point prompts. Under the official default inference configuration, this leads to substantially higher per-frame FLOPs despite a relatively small trainable decoder. PMD, by contrast, evaluates lightweight pairwise probe scores against a fixed anchor token and then performs simple averaging, resulting in a much smaller added inference cost.

Table 12: Inference efficiency comparison.

Method	Backbone	Trainable Module	Trainable Params ↓	Added FLOPs / Frame ↓
Large-scale supervised				
SegAnyMo [10]	DINOv2 + Depth Anything v2 + TAPIR	Trajectory-based motion segmentation network	3.44 M	>2.08 G
Flow-SAM [39]	SAM + RAFT	FlowP-SAM decoder + FlowI-SAM decoder	9.83 M	$>3.63 \times 10^4$ G
GeoMotion [8]	π^3 + RAFT	Confidence decoder + confidence head + motion-aware decoder	80.3 M	$>3.09 \times 10^2$ G
Ours (PMD, $k = 8$***)***				
PMD	π^3	Low-rank quadratic probe	**0.295 M**	**1.62 G**
PMD	π^3X	Low-rank quadratic probe	**0.295 M**	**1.62 G**

Discussion. As shown in Table 12, PMD remains substantially smaller than competing learned motion-decoding pipelines under this protocol. Its core rank-8 probe uses only 0.295M trainable parameters and adds 1.62G FLOPs per original video frame, whereas heavier motion pipelines require either much larger trainable decoders or significantly more expensive inference-time procedures. This gap is especially pronounced relative to GeoMotion [8], which also builds on π^3 as its feed-forward 3D backbone but still optimizes an 80.3M-parameter motion branch in its released frozen-backbone setting. It is also pronounced relative to Flow-SAM [39], whose repeated grid-prompt decoding makes inference particularly expensive.

End-to-end Latency and Peak Memory. Table 13 reports batch-1 wall-clock latency and peak GPU memory on one RTX 5070 Ti with 16 GB VRAM. Measurements include required preprocessing and backbone inference, use 10 warm-up frames and 100 measurement frames, and exclude model loading and result writes. Sequence-level runtimes are divided by 100 output frames. Each released pipeline otherwise retains its native settings, so these measurements compare complete implementations rather than equalized resolutions or module costs. Unlike the literature-quoted segmentation scores, these runtime measurements are obtained from the released pipelines on the same GPU.

Table 13: End-to-end inference measurements. SAM2 is included only in the corresponding refined pipelines; native SAM/SAM2 components are included for Flow-SAM and SegAnyMo.

Pipeline	Included inference scope	Latency (ms/frame)	Peak VRAM (GiB)
π^3X backbone	Preprocessing and frozen-backbone forward	480.6	4.47
π^3X+coarse PMD	Above, plus probe, aggregation, upsampling, thresholding	484.8	4.47
π^3X+PMD+SAM2	Above, plus sequence refinement	952.8	4.47
GeoMotion coarse [8]	Released π^3 + RAFT + motion-decoder pipeline	371.3	8.83
GeoMotion+SAM2	Above, plus released sequence refinement	526.7	9.80
Flow-SAM [39]	Released RAFT + FlowP/FlowI + SAM pipeline	3,104.5	12.33
SegAnyMo [10]	Released depth + appearance + tracking + SAM2 pipeline	2,921.1	3.02

Coarse PMD adds 4.2 ms/frame (0.87%) to π^3X inference, with no change in peak VRAM at the reported precision. Optional SAM2 adds 468.0 ms/frame and dominates the additional latency.

GeoMotion has lower total latency in both coarse and refined configurations despite its larger trainable motion branch, while SegAnyMo has lower measured peak VRAM. The efficiency claim therefore concerns PMD's small added readout cost and trainable capacity, not universal superiority in full-pipeline latency or memory.

F Broader Impacts

Our experiments show that motion-relevant information can be recovered from frozen 3D foundation models with low-capacity supervised probes and limited source-domain annotations.

Positive impacts. By reducing the computational and data requirements for dynamic scene understanding, our approach can improve the accessibility and efficiency of research on 4D perception. This is particularly relevant for resource-constrained settings, where the lightweight extraction of motion cues from pretrained models may support applications in robotics, autonomous driving, and AR/VR systems.

Potential negative impacts. Like any robust motion segmentation and tracking technology, our method could be repurposed for unauthorized surveillance. The ability to track moving entities without requiring labeled target data could theoretically be applied to unconstrained CCTV footage for privacy-invading monitoring. We advocate that the deployment of such systems be accompanied by appropriate privacy-preserving safeguards.

G Additional Quantitative Results

G.1 Additional Layer-wise Probing Results

Figure 8 complements the main layer-wise analysis by reporting results on VGGT [34], π^3 [36], and MapAnything [13]. The same trend holds beyond the three main backbones: low-rank quadratic probes consistently expose stronger motion decodability than linear or diagonal probes. VGGT's camera head collapses close to the all-static baseline, whereas π^3 shows a similar layer-wise pattern but retains decodable motion cues in its camera head, consistent with its highly similar yet more dynamic-scene-oriented training setup with datasets such as OmniWorld [49]. However, for π^3, the motion decodability is substantially stronger in the geometric backbone blocks, including inter-frame and intra-frame blocks, than in the camera head. MapAnything similarly shows stronger motion decodability in geometric feature blocks than in its pose head.

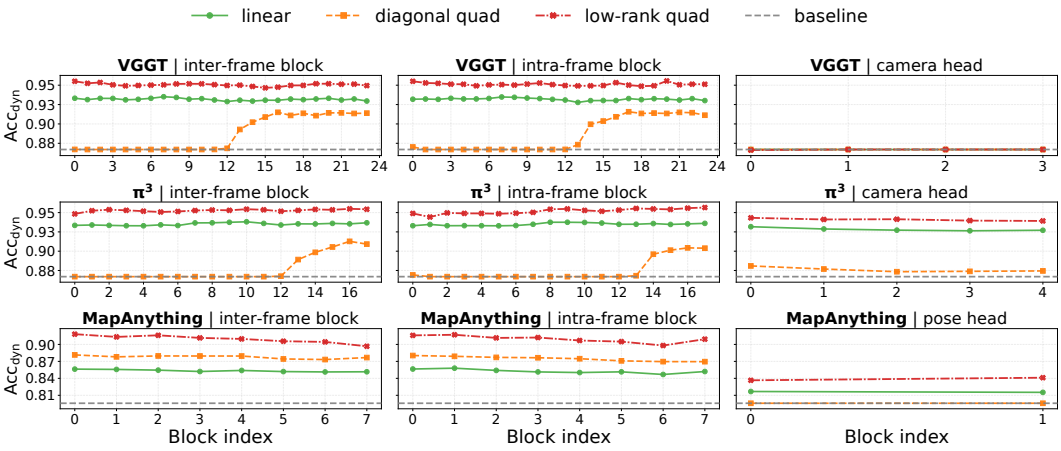


Figure 8: Additional layer-wise motion decodability on DAVIS-all [27]: VGGT [34], π^3 [36], and MapAnything [13].

G.2 Additional Backbone Benchmarking

Table 14 reports detailed PMD results across additional feed-forward 3D foundation model backbones, both with and without SAM2 [29] refinement. These results demonstrate that the probing-based

decoding framework is not specific to a single architecture. In these comparisons, π^3 and π^3X [36] combine strong coarse masks with small J-R gains from SAM2. Their larger J-M improvements show that refinement still affects mask quality substantially.

Table 14: Additional PMD benchmarking of different feed-forward 3D foundation model backbones. Each benchmark cell reports J-Mean / J-Recall. ΔJ-R denotes the J-Recall gain from SAM2 [29] refinement. Avg. ΔJ-R and Std. ΔJ-R are computed over DAVIS2017 [27], SegTrackv2 [17], and FBMS-59 [14]; smaller gains indicate less change in the fraction of frames exceeding 0.5 Jaccard, not necessarily smaller changes in mask quality.

Model	Refine	DAVIS17 (J-M / J-R) ↑	SegTrackv2 (J-M / J-R) ↑	FBMS-59 (J-M / J-R) ↑	Avg. ΔJ-R ↓	Std. ΔJ-R ↓
π^3-PMD	w/o SAM2	72.7 / 89.9	72.9 / 92.9	72.6 / 83.4	2.59	1.45
	w/ SAM2	80.2 / 91.1	84.5 / 97.5	81.1 / 85.4		
π^3X-PMD	w/o SAM2	73.7 / 91.0	72.9 / 94.0	73.5 / 88.8	2.37	2.11
	w/ SAM2	81.9 / 92.3	86.4 / 99.3	83.2 / 89.3		
DA3-PMD	w/o SAM2	69.6 / 85.8	66.3 / 80.3	69.8 / 81.4	6.36	2.96
	w/ SAM2	77.7 / 88.2	75.7 / 89.8	82.3 / 88.6		
DUSt3R-PMD	w/o SAM2	62.2 / 70.0	58.6 / 77.3	64.0 / 74.5	3.74	2.50
	w/ SAM2	66.5 / 70.6	66.0 / 84.0	74.0 / 78.3		
VGGT-PMD	w/o SAM2	72.6 / 89.9	70.8 / 84.7	73.5 / 85.9	6.35	3.92
	w/ SAM2	80.8 / 92.3	83.6 / 96.4	82.9 / 90.8		
MapA-PMD	w/o SAM2	69.2 / 86.0	62.3 / 75.7	69.1 / 79.9	3.95	6.11
	w/ SAM2	77.3 / 86.6	75.5 / 88.2	75.8 / 78.5		

G.3 Backbone Disparity and Source-validation Decodability

To compare recoverability across backbones before mask extraction, we evaluate rank-8 quadratic probes on the same 80-video YouTube-VOS validation split [41] and average over all blocks in each fixed PMD-selected group. We distinguish pairwise normalized gain from the unary dynamic-token NG defined in Section 4.1:

$$\mathrm{NG}_{\mathrm{pair}} = \frac{\mathrm{Acc}_{\mathrm{pair}} - \mathrm{Baseline}_{\mathrm{same}}}{1 - \mathrm{Baseline}_{\mathrm{same}}}, \qquad \mathrm{Baseline}_{\mathrm{same}} = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \mathbf{1}[z_{ij} = 1].$$

Here, $\mathrm{Acc}_{\mathrm{pair}}$ is the accuracy of thresholding pairwise compatibility at 0.5, and $\mathrm{Baseline}_{\mathrm{same}}$ always predicts that a pair shares the same motion state. These measurements use frozen tokens before thresholding score maps or applying SAM2. Target J-M is the mean coarse-mask J-M across DAVIS17, SegTrackv2, and FBMS-59.

Relative to DA3, π^3X has 10.59 points higher mean source-validation NG, 4.54 points lower across-block standard deviation, and 4.77 points higher mean target J-M (Table 15). The same restricted probe therefore recovers stronger and more uniform motion-state information from the selected π^3X tokens. On DAVIS17, the coarse π^3X–DA3 gap is 4.1 J-M, close to the 4.2-point gap after SAM2; the disparity is already present before refinement. Excluding DUSt3R reduces the cross-backbone standard deviation of three-benchmark mean J-M from 4.14 to 2.57, computed from unrounded benchmark means. This identifies DUSt3R as a relative outlier while leaving variation among the multi-view backbones.

Table 16 groups broad design regimes rather than strict architecture families. In particular, DA3 and MapAnything differ in token flow, supported input modalities, and prediction heads despite sharing a ViT-G-scale local/global geometry-decoding pattern. Source-validation pairwise NG is positively associated with coarse target J-M, including within the five multi-view backbones (Table 17).

High mean source-validation NG and low across-block dispersion may help identify promising candidates across broad design regimes without target-benchmark measurements. They do not

Table 15: Source-validation pairwise decodability and target coarse-mask performance. Source NG is $\mathrm{NG}_{\mathrm{pair}} \times 10^2$ (mean $\pm$ standard deviation across blocks in each selected group); target J-M is averaged over DAVIS17, SegTrackv2, and FBMS-59.

Backbone	**Architecture**	**Source-val. NG** mean $\pm$ std.	**Target J-M** ($\uparrow$)
Multi-view backbones (probe group: inter-frame)			
π^3X [36]	DINOv2-L + 36 alternating frame/global blocks	72.46 ± 2.06	73.35
π^3 [36]	DINOv2-L + 36 alternating frame/global blocks	76.76 ± 0.78	72.77
VGGT [34]	DINOv2-L + 24 paired frame/global stages	73.60 ± 4.18	72.31
DA3 [21]	40-block ViT-G; local/global alternation from block 13	61.87 ± 6.60	68.58
MapAnything [13]	DINOv2-G + separate 16-block UniCeption stack	67.20 ± 2.10	66.86
Pairwise backbone (probe group: decoder/head view 1)			
DUSt3R [35]	24-block CroCo encoder + two 12-block reciprocal decoders	33.73 ± 5.89	61.59

Table 16: Descriptive grouping by broad architectural design regime. Source-validation NG and target J-M are averaged over the backbones in each row.

Broad design regime	Backbones	Mean source NG $\times 10^2$	Mean target J-M
Separate ViT-L encoder + alternating frame/global multi-view stack	π^3X, π^3, VGGT	74.27	72.81
ViT-G-scale local/global multi-view stack + dense geometry heads	DA3, MapAnything	64.54	67.72
Pairwise asymmetric encoder–decoder with reciprocal cross-attention	DUSt3R	33.73	61.59

Table 17: Descriptive associations between source-validation pairwise NG and mean target coarse-mask J-M.

Comparison	Spearman ρ	Pearson r
All six backbones	0.77	0.93
Five multi-view backbones	0.60	0.80
Three design-regime means	1.00	0.97

determine the exact ordering of closely related backbones: π^3 has higher source-validation NG than π^3X, whereas π^3X has 0.58 points higher target J-M. These small-sample associations are descriptive; architecture, pretraining data, token flow, and prediction heads vary together, so the analysis does not isolate a causal architectural driver of motion recoverability.

G.4 Additional Feed-forward Geometry-based Comparisons

Table 18 complements the main benchmark by isolating feed-forward geometry-based motion segmentation methods and reporting their performance with and without SAM2 [29] refinement. This comparison is useful because these methods operate in a similar regime: they derive motion cues from feed-forward 3D foundation models. We therefore separate the quality of the coarse motion cues from the effect of SAM2 refinement by reporting both J-Mean and J-Recall before and after refinement, as well as the J-Recall gain ΔJ-R. For methods with high coarse J-R, a small ΔJ-R means that refinement moves relatively few additional frames above the 0.5 Jaccard threshold. It does not measure pixel recall or imply negligible improvement in mask extent or boundaries.

G.5 Additional Scene-Diversity Starvation Results

Figure 9 provides additional token-level scene-diversity starvation results for the default low-rank quadratic probe with $k = 8$. Across π^3X [36], DA3 [21], and DUSt3R [35], normalized gain decreases gradually as the number of training videos is reduced, further supporting that the decoded

Table 18: Feed-forward geometry-based motion segmentation comparison on DAVIS2017 [27]. We report J-Mean (J-M), J-Recall (J-R), and the J-Recall gain from SAM2 refinement. ΔJ-R measures changes in the fraction of frames above 0.5 Jaccard; J-M separately captures changes in average mask overlap. Note that PMD introduces substantially fewer additional parameters and lower FLOP overhead than GeoMotion; see Table 12.

Method	Flow	w/o SAM2		w/ SAM2		ΔJ-R ↓
		J-M ↑	J-R ↑	J-M ↑	J-R ↑	
Easi3R$_{\text{dust3r}}$ [5]	×	46.9	50.5	60.1	65.3	14.8
Easi3R$_{\text{monst3r}}$ [5]	×	54.8	66.2	67.9	76.1	9.9
MonST3R [46]	✓	38.1	36.1	56.4	59.6	23.5
DAS3R [40]	×	44.5	44.0	57.4	61.3	17.3
VGGT4D [9]	×	56.5	65.6	–	–	–
GeoMotion [8]	✓	**75.4**	87.2	81.1	**92.6**	5.4
Ours						
π^3-PMD (ours)	×	72.7	89.9	80.2	91.1	**1.2**
π^3X-PMD (ours)	×	73.7	**91.0**	**81.9**	92.3	1.3

motion cues are compact and can be recovered with limited scene diversity. Among the three backbones, π^3X achieves the highest normalized gain and degrades most slowly under reduced training data. This observation is consistent with the quantitative evaluation in Table 2, where the π^3X-PMD variant achieves the strongest downstream motion segmentation performance. To assess sensitivity to subset choice, we additionally report the mean and one-standard-deviation band over three random subset seeds whenever multiple seeds are available. The uncertainty remains modest overall and increases mainly in the most data-starved settings, indicating graceful degradation rather than unstable collapse.

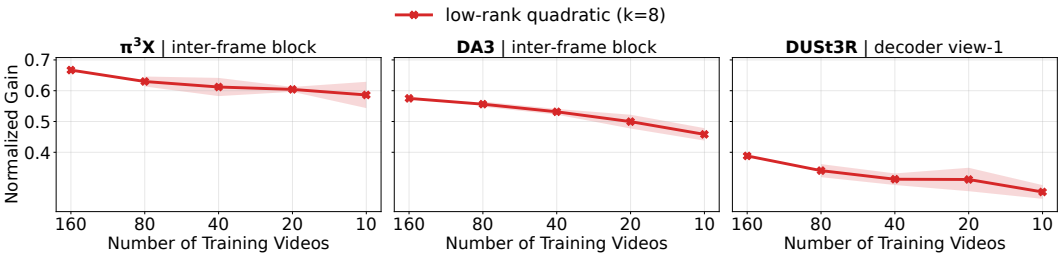


Figure 9: Additional scene-diversity starvation analysis on DAVIS-all [27]. We report token-level Normalized Gain for the default low-rank quadratic probe ($k = 8$) across π^3X [36], DA3 [21], and DUSt3R [35] while reducing the number of YouTube-VOS training videos. Solid lines denote the mean over available random subset seeds, and shaded bands denote one standard deviation when multiple seeds are available. Performance degrades gradually as training diversity is reduced, while the uncertainty bands remain moderate overall and widen mainly in the most data-starved regimes.

G.6 Extended Benchmark Results

The following table retains the full set of benchmark scores, including DAVIS16 and DAVIS-all. Refinement use and target-domain training are marked explicitly; the grouped comparison in the main text provides additional context on supervision and inference-time priors.

H Additional Qualitative Results

In this section, we provide extended qualitative visualizations to complement the findings in the main paper. We focus on: (1) the effect of network depth and scene-diversity starvation on motion–semantic entanglement, and (2) an analysis of failure cases and current limitations.

Table 19: Extended motion segmentation benchmarking. All non-PMD scores are quoted from the cited papers; only PMD rows are evaluated with our evaluator. These complete-pipeline results have differing supervision and inference priors, separated in Table 2. Trained on only 160 YouTube-VOS 2018 [41] videos, PMD transfers across DAVIS [26, 27], SegTrackv2 [17], and FBMS-59 [14]; its strongest multi-view pretrained variants approach supervised and self-supervised methods. Gray entries denote in-domain training, fine-tuning, or test-time optimization on the target benchmark. Methods using SAM/SAM2 [15, 29], either for external refinement or as a native pipeline component, are marked with †.

Method	DAVIS16 (J-M ↑)	DAVIS17 (J-M ↑)	DAVIS-all (J-M ↑)	SegTrackv2 (J-M ↑)	FBMS-59 (J-M ↑)
Unsupervised / Self-supervised					
CIS [43]	71.5	–	–	62.0	63.5
OCLR [37]	80.9	–	–	72.3	69.8
DyStaB [44]	82.8	–	–	72.7	68.6
RCF [20]	80.9	–	–	76.7	69.9
OCMOS [38]	81.1	–	–	76.6	81.9
EM-OFS [24]	75.2	–	–	65.5	71.3
FlowFeat [2]	–	62.3	–	–	–
Training-free / Heuristic					
ARP [16]	76.2	–	–	57.2	59.8
VGGT4D [9]	62.1	56.5	50.75	–	–
Easi3R$_{\text{dust3r}}$† [5]	67.9	60.1	54.7	–	–
Easi3R$_{\text{monst3r}}$† [5]	70.7	67.9	63.1	–	–
Supervised					
MATNet [48]	82.4	–	–	50.4	76.1
COSNet [22]	80.5	–	–	49.7	75.6
AMC-Net [42]	84.5	–	–	–	76.5
Flow-SAM† [39]	86.2	71.6	–	84.2	**88.7**
SegAnyMo† [10]	**90.6**	–	–	76.3	78.3
GeoMotion† [8]	84.5	81.1	74.8	77.3	72.5
Ours (data-efficient probing)					
π^3X-PMD†	81.4	**81.9**	**80.2**	**86.4**	83.2
π^3-PMD†	79.5	80.2	79.6	84.5	81.1
DA3-PMD†	75.9	77.7	75.7	75.7	82.3
DUSt3R-PMD†	61.5	66.5	63.8	66.0	74.0

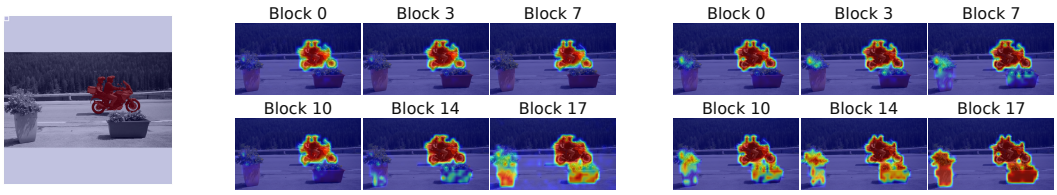
H.1 Visualization of the Effect of Network Depth and Scene-Diversity Starvation

Figure 10 provides visual evidence of how the decoded motion cues and motion–semantic entanglement are affected by both network depth and training-set size, using the π^3X backbone [36] as a representative example.

We compare the ground-truth motion masks (shown in Figure 10a) against π^3X-PMD dynamic score maps extracted across different network layers, contrasting our default 160-video training subset with a severely starved 10-video subset. As shown, the 160-video model highlights the moving target in its shallow and intermediate layers. However, deeper layers begin to entangle motion with semantic category information, consistent with the increased effective rank required for decoding in deeper blocks (Figure 4c). Under extreme data scarcity (10 videos), this entanglement becomes more pronounced across layers: the starved probe assigns high dynamic scores to static foreground instances rather than strictly isolating actual moving objects.

H.2 Failure Cases and Limitation Analysis

Figure 6 presents a semantic-leakage example discussed in Section 5. Here, Figure 11 illustrates the complementary limitation of binary motion detection. These cases reflect the binary static–dynamic



(a) Ground Truth (b) Trained with 160 videos (c) Trained with 10 videos

Figure 10: Scene-diversity starvation and layer-wise semantic leakage on π^3X [36]. Visualization of PMD dynamic score maps across network depth under varying amounts of training data. While deeper representations entangle motion with semantics, reducing the training subset to only 10 videos exacerbates this leakage across all layers. The starved probe incorrectly highlights static foreground objects, whereas the 160-video model maintains a focus on actual motion, particularly in its shallow and intermediate layers.

formulation, possible motion–semantic entanglement in frozen representations, and object-centric source supervision.

Coarse Binary Detection vs. Fine-Grained Motion. PMD is designed to identify and segment dynamically moving regions. As illustrated in Figure 11, it does not currently distinguish fine-grained motion patterns, such as in-place articulation, localized deformation, or different motion magnitudes, or decode explicit physical kinematics, such as 3D displacement or velocity vectors.



(a) Input frame (b) PMD dynamic score map (c) Binary mask

Figure 11: **Limitation: Fine-grained Motion vs. Binary Detection.** PMD highlights the breakdancer with the most prominent motion and assigns moderately elevated scores to nearby people with subtler in-place motion relative to the static background. However, the pipeline ultimately produces a binary static–dynamic mask rather than distinguishing different motion types or motion magnitudes.

Semantic Leakage. The probe sometimes struggles to fully decouple motion from semantics. As shown in Figure 6, in scenes containing multiple instances of the same object category, static instances may receive erroneously elevated dynamic scores when another instance of that category is moving. Notably, this semantic leakage can occur even for categories absent from the 160-video training subset, consistent with entanglement of motion and category-level cues. Object-centric YouTube-VOS supervision may also contribute; the present experiments do not isolate these sources of bias or establish pure geometric dynamicity.

Evaluation Scope. The present evaluation covers frozen feed-forward 3D reconstruction backbones. Controlled comparisons with video segmentation and tracking foundation models under matched supervision and prompting would address differences in pretraining objectives, architectures, and decoding pipelines. Whether the observed recoverability is specific to 3D pretraining remains open. Training beyond 160 YouTube-VOS videos has not been evaluated, leaving performance saturation, an optimal data budget, and the benefits of additional source diversity unresolved.

I Existing Assets, Licenses, and Terms of Use

In this section, we provide detailed information regarding the existing assets used in our research to document compliance with the NeurIPS guidelines on data and model usage. Table 20 outlines the licenses and specific terms of use for all pretrained models and tools integrated into our pipeline.

Similarly, Table 21 summarizes the licensing details for the datasets and benchmarks used for training and evaluation. We obtained these assets from their official sources and used them for non-commercial research evaluation, following the corresponding licenses and terms of use to the best of our understanding.

Table 20: Existing pretrained models and tools used in this work. We obtain all models and tools from their official sources and follow their respective licenses and terms of use.

Asset	Use in this work	License / terms
SAM2 [29]	Optional mask refinement	SAM2 code and model weights are released under Apache-2.0; the SAM2 evaluation code is released under BSD-3-Clause. We use SAM2 only as an optional refinement module and follow the official license terms.
π^3/π^3X [36]	Frozen backbone	Code is released under BSD-3-Clause; π^3 and π^3X model weights are released under CC BY-NC 4.0 and are restricted to non-commercial research and educational use.
Depth Anything 3 [21]	Frozen backbone	Code is released under Apache-2.0. The DA3-GIANT-1.1 model weights used in this work are released under CC BY-NC 4.0 and are restricted to non-commercial use.
DUSt3R [35]	Frozen backbone	Code and released materials are distributed under CC BY-NC-SA 4.0, with non-commercial use only. We use DUSt3R only for research evaluation.
VGGT [34]	Frozen backbone	The VGGT repository is distributed under the custom VGGT License. According to the official repository, only the newer VGGT-1B-Commercial checkpoint is licensed for commercial usage, whereas the original VGGT-1B checkpoint remains non-commercial. Our experiments use the original VGGT-1B checkpoint only for non-commercial research evaluation and follow the corresponding license terms.
MapAnything [13]	Frozen backbone	Code is released under Apache-2.0. The official pretrained model variants have separate licenses: `facebook/map-anything` is CC BY-NC 4.0, while `facebook/map-anything-apache` is Apache-2.0. We follow the license of the specific checkpoint used in each experiment.

Table 21: Existing datasets and benchmarks used in this work. We obtain all datasets from their official sources and follow their respective licenses and terms of use.

Asset	Use in this work	License / terms
YouTube-VOS 2018 [41]	Probe training and validation source	Annotations are licensed under CC BY 4.0; data are released for non-commercial research purposes only. We use the dataset only for research and obtain it from the official source.
DAVIS 2016/2017 [26, 27]	Evaluation benchmark	DAVIS 2017 challenge annotations are licensed under CC BY 4.0; the official DAVIS 2017 evaluation code is released under BSD-3-Clause. We use the dataset only for benchmark evaluation and cite the official DAVIS papers.
SegTrackv2 [17]	Evaluation benchmark	The official dataset page does not explicitly state a standard license. We use SegTrackv2 only for non-commercial research evaluation, obtain it from the official project page, and cite the original dataset paper.

Continued on next page

Table 21 (continued)

Asset	Use in this work	License / terms
FBMS-59 [14]	Evaluation benchmark	Custom research-only terms: the dataset is provided only for research purposes and without warranty; commercial use is prohibited. We use FBMS-59 only for research evaluation and cite the corresponding dataset paper.

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 introduction state the main claims: motion-relevant information can be decoded from frozen feed-forward 3D foundation models using low-capacity probes, and PMD achieves data-efficient zero-shot transfer to standard motion segmentation benchmarks. These claims are supported by the layer-wise probing, benchmark comparisons, data-starvation analysis, and compactness analysis in Sections 4.2–4.4; limitations are discussed in the Limitations section.

   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: The paper includes a dedicated Limitations section discussing token-level spatial granularity, the binary static–dynamic formulation, lack of fine-grained motion quantities such as displacement or velocity, and semantic leakage failure cases. Additional qualitative examples are provided in Appendix H, while SAM2 refinement dependence is quantified separately in Tables 3 and 14.

   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 present formal theoretical results or theorems. Appendix D provides a capacity-based explanatory analysis using standard learning-theoretic scaling arguments, but it is not used as a formal theorem or proof.

   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: The paper describes the probe formulation, PMD inference procedure, selected blocks, training and validation data, hyperparameters, mask extraction, SAM2 prompt construction, and evaluation protocols. Further details are provided in Section 4.1 and Appendix A.

   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 anonymized code and detailed reproduction instructions in the supplementary material. The supplementary package explains how to obtain the original public datasets from official sources, includes the sampled split files for the YouTube-VOS subsets used in our experiments, and provides scripts or exact commands for reproducing the main training, inference, and evaluation pipelines covered by the release. For results that rely on additional downloaded artifacts or pretrained checkpoints, we explicitly document the required files, expected directory layout, and reproduction steps.

   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: The paper specifies the training and validation splits, evaluation benchmarks, optimizer, learning rate, number of epochs, sampled token pairs, selected blocks, mask

thresholding, SAM2 prompt construction, inference context, and evaluation metrics. These details are provided in Section 4.1 and Appendix A.

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: The main benchmark tables do not report confidence intervals, statistical tests, or error bars for all methods and backbones. Appendix G reports mean and one-standard-deviation bands for the data-starvation experiment when multiple subset seeds are available, but this uncertainty analysis does not cover all main benchmark results.

   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: The paper reports the hardware setup, including GPU, CPU, memory configuration, trainable parameter counts, and inference FLOP overhead in Section 4.1 and Appendices A and E. All experiments are conducted on a single consumer-grade GPU workstation without distributed training.

   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: The research uses public datasets and pretrained models, does not involve private data collection or human-subject experiments, and discusses potential misuse risks. Broader impacts and potential negative uses are addressed in Appendix F.

   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 F discusses positive impacts such as improving accessibility and computational efficiency for dynamic scene understanding, as well as potential negative impacts such as misuse for unauthorized surveillance.

    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: The paper does not release a new high-risk pretrained model, image generator, scraped dataset, or other asset requiring controlled-release safeguards. The work uses public datasets and existing pretrained models, and potential misuse of motion segmentation is discussed in Appendix F.

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: The paper credits the existing datasets, pretrained models, and tools through citations, including YouTube-VOS, DAVIS, SegTrackv2, FBMS-59, SAM2, and feed-forward 3D foundation models. Appendix I lists these assets and their license or terms-of-use information where publicly available; for SegTrackv2, whose official page does not explicitly state a standard license, we state this limitation and use the dataset only for research evaluation.

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: The supplementary material includes anonymized code and sampled split files used for reproduction. These assets are documented in the accompanying README files, which describe the expected directory structure, required external dependencies, dataset preparation, and the commands needed to reproduce the released experiments. We do not release a new dataset or benchmark; the released assets are intended to support reproducibility of the method and experiments in the paper.

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 experiments or research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human-subject 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 were used only for writing refinement, language editing, formatting assistance, and non-core utility code such as visualization or file-I/O helpers. LLMs were not used as part of the core method, experimental design, data generation, analysis, or scientific claims; therefore, declaration is not required under the checklist guidance.

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.