SELF-SUPERVISED LEARNING OF STRUCTURED MOTION FROM VIDEOS

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding motion in video is a fundamental challenge for visual learning, as frame-to-frame change entangles two sources of motion: camera and object. This decomposition has remained underexplored in representation learning, partly because these factors are tightly coupled in natural videos and difficult to supervise separately. Yet recovering it is important for learning robust motion representations that separate meaningful object movement from camera-induced variation. We study whether such structured motion representations can be recovered from frozen features of a pretrained image vision transformer. We propose the Structured Motion Model (SMM), which explicitly separates the dominant source of temporal change from residual motion through feature prediction, rather than representing video change with a single entangled latent or with unstructured, spatially dense transition tokens. Training combines self-supervised learning on real video with weak supervision of scene motion on synthetic Kubric data. We evaluate SMM on ProbeMotion, a new evaluation suite spanning synthetic and real videos with camera, object, and combined motion. SMM outperforms backbone baselines using global `CLS` or average-pooled features, and compares favorably to strongly supervised representations such as VGGT on several probes, despite using *substantially* weaker supervision. These results suggest that pretrained image models can be readily repurposed into structured motion representations, providing a useful inductive bias for learning and analyzing latent video motion.[1]

1 INTRODUCTION

Recent progress in computer vision has produced strong self-supervised image backbones that capture rich semantic and geometric structure from large-scale image data (Oquab et al., 2023; Siméoni et al., 2025; Weinzaepfel et al., 2022; Tschannen et al., 2025; Cao et al., 2026; Yang et al., 2025; Fini et al., 2025). These representations are increasingly used as foundations for 3D perception and reconstruction systems. However, despite building on self-supervised backbones, many successful approaches still rely on substantial supervision during adaptation, for example through 3D annotations, pseudo-ground truth from structure-from-motion pipelines, or other geometric labels (Wang et al., 2025b; 2024; 2025d; Lin et al., 2025a; Luo et al., 2026). VGGT, for instance, trains on 17 annotated datasets (Wang et al., 2025b). This limits scalability and leaves open the question of how much scene motion can be recovered directly from generic pretrained visual features without relying on dense supervision.

Video offers a natural source of supervision for this problem. However, frame-to-frame change entangles the two sources of motion, camera and object. Standard self-supervised prediction objectives can therefore model aggregate change without separating its underlying causes. Recovering such structure is important for learning motion representations organized around interpretable physical factors, but remains underexplored in representation learning. In particular, many latent action or world models summarize scene change with a single latent token (Garrido et al., 2026; Kerssies et al., 2026), mixing dominant global change with residual object-centric movement.

Rather than training a video model end-to-end, we ask whether a pretrained image backbone already contains enough information to support an explicit structuring of scene motion. To this end, we build a *Structured Motion* Model (SMM) on top of frozen backbone features and train it through feature prediction. We find that, when combined with weak supervision on synthetic data indicating only whether the camera or the scene is static, SMM organizes temporal change into primary and residual motion components that adapt to the dominant sources of change in each video.

[1] Code will be made available.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

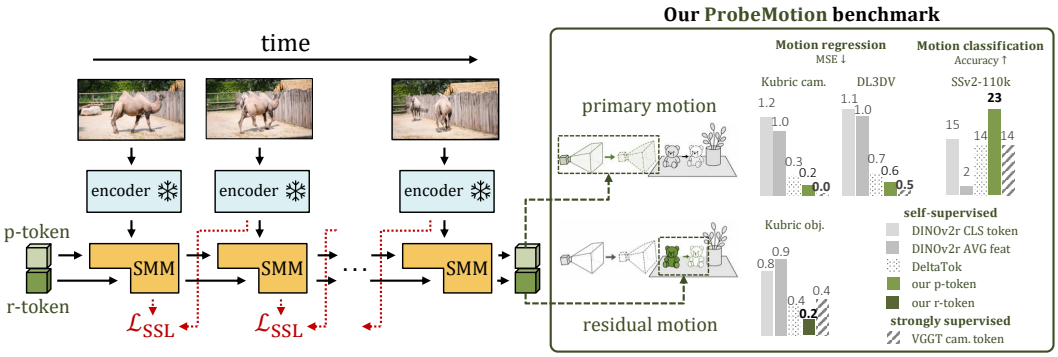


Figure 1: **The Structured Motion Model (SMM) remodels frozen image features into structured motion tokens.** Left: A frozen image encoder extracts frame features, and a recurrent SMM is trained by self-supervised feature prediction. SMM structures temporal change into a primary motion token p for the dominant source of change and a residual motion token r for the remaining differences. Right: Linear probes on ProbeMotion show that these structured motion tokens outperform direct frozen-backbone descriptors such as `CLS` and average-pooled features, and even surpass VGGT probing performance on 3/7 tasks (see Table 2), while showing stronger generalization to motion-adjacent semantic action classification.

We evaluate this setting on an evaluation suite of benchmarks spanning synthetic and real videos with camera, object, and combined motion. We refer to this collective evaluation as ProbeMotion. Across these settings, SMM consistently outperforms naïve baselines based on global `CLS` or average-pooled features and even surpasses VGGT-based probes on 3 out of 7 benchmarks. These results suggest that pretrained image representations can be remodeled into strong structured motion representations, with only weak synthetic supervision and self-supervised learning on unlabeled real videos.

Our contributions are:

- We study whether structured motion representations can be extracted from frozen pretrained image features.
- We introduce the Structured Motion Model (SMM), a simple architecture on top of self-supervised image backbones that learns this structure through feature prediction, outperforms naïve global-feature baselines, and is competitive with strongly supervised 3D representations on several motion-probing tasks.
- We develop ProbeMotion, an evaluation suite spanning synthetic and real videos covering camera, object, and combined motion.

2 Method

We learn structured motion representations by predicting how frozen visual features change over time. Rather than training a video encoder, we ask whether a simple model, SMM, on top of a pretrained image backbone can organize temporal change into primary and residual motion components, which explain the dominant source of scene change and remaining motion. An overview is shown in Figure 2.

2.1 Frozen visual features

Let $\mathcal{E}$ denote a frozen vision transformer. For each frame x_t, we extract a spatial feature map:

$$f_t = \mathcal{E}(x_t) \in \mathbb{R}^{H \times W \times D}, \tag{1}$$

where H, W are spatial dimensions and D is the feature dimension. SMM predicts the target feature map f_t from f_{t-1} in frozen feature space. The model is autoregressive in its recurrent motion tokens but non-causal, f_t is available for extracting the transition token used to predict f_t from f_{t-1}. SMM is therefore an *observed-transition* encoder.

Conditional feature prediction requires information about the target transition. CroCo and SiamMAE use cross-view conditioning (Weinzaepfel et al., 2022; Gupta et al., 2023), while latent-action models use compact transition tokens for future prediction (Garrido et al., 2026; Kerssies et al., 2026). A single transition token must explain all sources of change, from appearance-dependent feature variation to low-dimensional motion factors such as camera translation or object direction. We instead follow the intuition that much of the temporal change in videos is governed by lower-dimensional movements acting on high-dimensional visual features. This motivates a *structured* prediction model that organizes temporal change into primary and residual motion tokens.

2.2 STRUCTURED MOTION MODEL

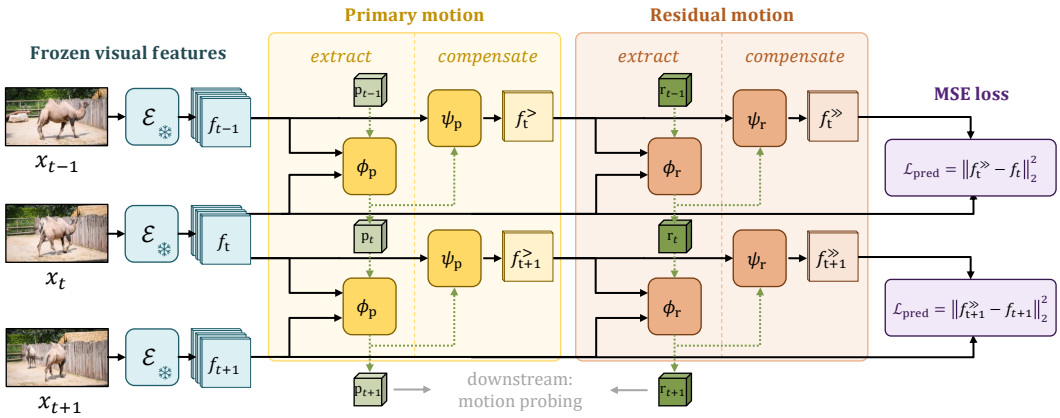


Figure 2: **Overview of the Structured Motion Model shown for T=3.** For each transition, SMM predicts the target frozen feature map f_t from the previous one f_{t-1} through two sequential compensation stages. The *primary* stage extracts a recurrent motion token p_t from the feature pair (f_{t-1}, f_t) and uses it to produce a primary-compensated feature map $f_t^{>}$, explaining the dominant change. The *residual* stage extracts r_t from the mismatch between $f_t^{>}$ and f_t, and refines $f_t^{>}$ into the final prediction $f_t^{\gg}$. The motion tokens are autoregressive but non-causal across time, allowing temporal information to accumulate.

Figure 2 gives an overview of SMM. Given consecutive feature maps (f_{t-1}, f_t), SMM predicts f_t through two sequential stages. The *primary stage* consists of a motion extractor ϕ_p and a feature predictor ψ_p. The extractor reads the current feature pair (f_{t-1}, f_t) and updates the previous primary token p_{t-1}, giving p_t. The predictor then uses p_t to compensate the source feature map f_{t-1}, producing an intermediate prediction $f_t^{>}$, pronounced 'f modified once'. The *residual stage* has the same structure, with modules ϕ_r and ψ_r. It estimates a residual token r_t from the remaining mismatch between $f_t^{>}$ and f_t, and uses this token to refine $f_t^{>}$ into the final prediction $f_t^{\gg}$, pronounced 'f modified twice'.

Primary motion extraction. The primary extractor updates a recurrent motion token $p_t \in \mathbb{R}^D$ by attending to the current frame pair:

$$p_t = \phi_p(p_{t-1}; f_{t-1}, f_t), \tag{2}$$

where p_0 is learnable. The recurrence allows the model to integrate motion information over time while each update only compares adjacent frames. The resulting token is then used to compensate the source feature map.

Primary motion compensation. The predictor ψ_p uses p_t to compensate the source feature map:

$$f_t^{>} = \psi_p(f_{t-1}; p_t). \tag{3}$$

We interpret $f_t^{>}$ as a primary-motion-compensated intermediate representation that serves as input to the residual stage. It is trained to explain the part of the transition from f_{t-1} to f_t that can be captured by the primary source of scene change.

Residual motion extraction. Given $f_t^{>}$, the remaining discrepancy to f_t corresponds to motion not explained by the primary component. We capture this with a second latent token $r_t \in \mathbb{R}^D$:

$$r_t = \phi_r(r_{t-1}; f_t^{>}, f_t), \tag{4}$$

where r_0 is learnable. Conditioning it on $f_t^{>}$ encourages r_t to represent only the residual motion that remains after the primary component has been accounted for.

Residual motion compensation. The residual predictor refines the intermediate representation:

$$f_t^{\gg} = \psi_r(f_t^{>}; r_t). \tag{5}$$

The final prediction $f_t^{\gg}$ is therefore obtained by first compensating for primary and then residual motion. The pair (p_t, r_t) forms the structured motion representation used throughout the paper.

Extension beyond pairs of frames. SMM naturally extends to longer clips. We apply the same updates at each timestep and carry p_{t-1} and r_{t-1} forward. Thus, temporal context is accumulated in the recurrent motion tokens.

2.3 TRAINING OBJECTIVE

SMM is trained through feature prediction in the frozen encoder space. To align the architectural decomposition with the intended semantics of primary and residual motion, we use two weak scene-level annotations, static scene and static camera, and selectively apply the loss.

Unlabeled or dynamic videos. For unlabeled videos and samples with both camera and scene motion, we supervise only the final prediction: $\mathcal{L} = \mathcal{L}_{\text{MSE}}(f_t^{\gg}, f_t)$, where $\mathcal{L}_{\text{MSE}}(x,y) = \frac{1}{HWD}\|x-y\|_2^2$.

Static scenes. For static-scene samples, the primary component should explain the full transition, so we bypass the residual stage and supervise the output of the primary compensation directly: $\mathcal{L} = \mathcal{L}_{\text{MSE}}(f_t^{>}, f_t)$.

Static cameras. If a sample is annotated as having a static camera, there is no camera-induced global change to compensate for. Hence, the primary stage is regularized to leave the source representation unchanged while the residual stage captures scene motion: $\mathcal{L} = \mathcal{L}_{\text{MSE}}(f_t^{\gg}, f_t) + \lambda_{\text{reg}}\mathcal{L}_{\text{MSE}}(f_t^{>}, f_{t-1})$.

2.4 IMPLEMENTATION DETAILS

Visual features. We use frozen DINOv2-B/14 with registers (Darcet et al., 2024) as $\mathcal{E}$ and extract patch features from all 12 blocks. Following Mur-Labadia et al. (2026), we concatenate these intermediate features channel-wise and project them with a two-layer MLP to $D{=}768$, matching the hidden dimension of one backbone block, before applying SMM. The model's predictions are then linearly projected back to the concatenated multi-layer feature space before applying the loss.

Motion extractor ϕ. Both ϕ_p and ϕ_r are 4-block transformer decoders with $n_{\text{reg}} = 4$ learnable registers (Darcet et al., 2024). The motion token and registers self-attend and cross-attend to frame features, using a 3D extension of RoPE (Su et al., 2024; Assran et al., 2025) for spatial and temporal positions with a base of 100 (Heo et al., 2024). The initial primary and residual motion tokens are learnable parameters initialized from $\mathcal{N}(0, 0.02^2)$.

Predictor ψ. We parameterize ψ_p and ψ_r as shallow 2-block decoders that cross-attend to the corresponding motion tokens. We use 2D RoPE for spatial positions (Heo et al., 2024).

3 EXPERIMENTS

3.1 DATASETS

We train on a mixture of synthetic and real videos: Kubric (Greff et al., 2022), SSv2 (Goyal et al., 2017), and DL3DV (Ling et al., 2024). Kubric provides weak scene-level labels indicating whether the camera or scene is static, while SSv2 and DL3DV are used without labels. We generate 180k Kubric sequences, equally split between static-scene and dynamic-scene videos. Static-scene videos use a moving camera, while dynamic-scene videos are further equally split between moving-camera and static-camera settings (see Section A.1 for details). Real-video training uses approximately 170k SSv2 clips and 4k DL3DV clips.

For evaluation, we introduce ProbeMotion, a diverse benchmark suite for probing structured motion representations across Kubric, DL3DV, CameraBench (Lin et al., 2025b), static-camera subsets of DAVIS2017 (Pont-Tuset et al., 2017) and YouTubeVOS (Xu et al., 2018b;a), and an SSv2 subset denoted SSv2-110k. Table 1 summarizes the datasets. Split and sampling details are in Section A.3.

Kubric evaluates controlled synthetic camera and object motion with disjoint training/test objects and backgrounds as well as randomly sampled motions. DL3DV and CameraBench evaluate real camera motion, while static DAVIS2017 and static YouTubeVOS evaluate 2D object motion under approximately static cameras. We construct the static-camera subsets by filtering clips using VGGT-estimated camera motion and labeling 2D object motion from mask-centroid displacement (see Section A.2 for details). SSv2-110k evaluates motion-heavy action semantics.

3.2 TRAINING AND EVALUATION DETAILS

Training details. Mini-batches mix Kubric, SSv2, and DL3DV with fixed ratios of $0.5/0.25/0.25$. Kubric samples are drawn from the three controlled settings described above: moving-camera/static-scene, static-camera/dynamic-scene, and moving-camera/dynamic-scene. Consequently, $1/4$ of each mini-batch receives static-scene supervision, $1/8$ receives static-camera supervision, and the remaining $5/8$ is trained only with the standard final-prediction objective. This last group consists of moving-camera/dynamic-scene Kubric samples and all unlabeled real-video samples, for which we do not impose additional structure

Table 1: **ProbeMotion evaluation datasets.** Summary of the linear-probe datasets. Numbers given for source videos refer to the samples taken from the corresponding public or generated dataset, where applicable after clip-length filtering. Dashes indicate that the source dataset does not provide a split suitable for ProbeMotion. For these datasets, we create our own probing splits. Parentheses indicate partial motion for SSv2's camera (see Section B.1). Further details, including split creation for these datasets, are provided in Section A.3.

	Kubric	DL3DV	Static DAVIS2017	Static YouTubeVOS	Camera Bench	SSv2-110k
Source train/test videos	15k/5k	3.5k/–	60/30	3.5k/–	1.4k/1.1k	110k/20k
Dynamic camera/scene	✓/✓	✓/✗	✗/✓	✗/✓	✓/✓	(✗)/✓
Num. frames per clip	4	5	4	4	5	7
Probing target	cam.+obj. motion	cam. motion	obj. displacement	obj. displacement	15 binary cam. motion tasks	action class

on the decomposition. We sample 5 frames at 2 fps and resize inputs to 224×224 without additional data augmentation. We train all components with AdamW for 200k iterations, batch size 128, learning rate 6.25×10^{-5} with 1k warmup, weight decay 0.05, and set the regularization weight to $\lambda_{\text{reg}}=0.5$. One run takes approximately 11h on 4 H100 GPUs.

Evaluation protocol. We evaluate ProbeMotion with linear probes on frozen SMM motion tokens and report final-timestep performance over 3 seeds. We probe p for camera motion on Kubric, DL3DV, and CameraBench, and for object motion on approximately static-camera DAVIS2017/YouTubeVOS. For SSv2-110k, where camera motion is limited (see Table 1 and Section B.1), we probe p for action recognition. For Kubric object motion under camera motion, we probe r and study token specialization and composition in Section 3.4. Regression targets are 6-DoF camera motion for Kubric/DL3DV, 3D object translation for Kubric, and 2D object displacement for DAVIS2017/YouTubeVOS. CameraBench uses macro mAP and SSv2-110k top-1 accuracy. Full task, split, balancing, and probe-training details are in Section A.

3.3 Comparison to baselines

Baselines. Since image backbones produce frame-level representations, we convert consecutive-frame features into motion descriptors using either concatenation or feature differences. We compare against direct frozen-backbone descriptors from the `CLS` token and average-pooled patch features, using concatenation for `CLS` and feature differences for AVG-pooled features based on the ablations in Section B.2. Multi-frame variants average these frame-pair descriptors over time to match SMM's temporal context. We additionally evaluate the self-supervised DeltaTok (Kerssies et al., 2026), which represents frame-to-frame change with a single delta token. DeltaTok is trained for $8\times$ more iterations than SMM and operates at higher resolution, making it a strong baseline (see Section A.7). We also compare to strongly supervised 3D and geometry models: VGGT-1B, Depth Anything 3 (DA3) (Lin et al., 2025a), and Pi3X (Wang et al., 2025d). For each, we evaluate average-pooled feature maps and camera tokens, selecting the better of concatenation and feature differences per benchmark based on Section B.2. For Pi3X, which has no camera token, we use averaged register tokens instead. These models are substantially larger than SMM (roughly 1B parameters each *vs*. 215M, including an 87M frozen backbone) and use explicit geometric supervision such as camera pose, depth, or point clouds, contextualizing performance under much stronger supervision.

SMM outperforms direct frozen-backbone descriptors. Across ProbeMotion, SMM improves over `CLS` and AVG-pool baselines on all tasks except static DAVIS2017. This exception might reflect the small, curated nature of static DAVIS2017 and the relative simplicity of image-plane displacement prediction. The direct baselines remain substantially weaker on 3D motion probes such as Kubric and DL3DV, suggesting that gains come from structured feature prediction rather than strong frozen features alone. This remains true when the direct baselines are given access to multi-layer DINOv2 features from all 12 blocks (see Table 9). We also test whether DINOv2's representations contain complementary task-relevant information, and find that concatenating `CLS` or AVG-pooled features with the task-appropriate SMM token does not improve linear-probing performance, suggesting that frozen-backbone descriptors do not provide complementary task-relevant information under linear probing (see Table 10).

SMM outperforms DeltaTok across most probes despite a smaller training budget. SMM is stronger on five of seven ProbeMotion tasks, with the largest gains on Kubric object motion, supporting the value of structured latents across diverse motion and action probes.

SMM is competitive with supervised 3D representations on ProbeMotion. Despite using weaker supervision and fewer parameters than the supervised 3D and geometry baselines, SMM approaches

Table 2: **Linear probing results on ProbeMotion.** We compare SMM with direct DINOv2 with registers descriptors, self-supervised DeltaTok, and strongly supervised 3D/geometry models. DINOv2 uses `CLS` concatenation or AVG-pooled feature differences, supervised descriptors use the better of concatenation and feature differences per benchmark (Section B.2), with averaged registers replacing camera tokens for Pi3X. DeltaTok is probed via its delta token. SMM outperforms direct DINOv2 baselines on most tasks and DeltaTok on five of seven, while remaining competitive with supervised representations. We report standardized-target MSE, CameraBench macro mAP (%), and SSv2-110k accuracy (%). Best and second-best results among methods without explicit geometric supervision are shown in **bold** and underlined.

Method	Regression (MSE ↓)					Classification (mAP/acc. ↑)	
	Kubric cam	Kubric obj	DL3DV	Static DAVIS2017	Static YouTubeVOS	Camera Bench	SSv2-110k (action pred.)
strongly supervised							
VGGT avg. feat.	0.09±0.0	0.32±0.0	0.56±0.0	0.78±0.1	0.79±0.0	48.6±0.4	15.5±0.1
VGGT cam. token	0.03±0.0	0.43±0.0	0.53±0.0	0.71±0.0	0.81±0.0	57.9±0.4	13.5±0.1
DA3 avg. feat.	0.11±0.0	0.40±0.0	0.55±0.0	0.87±0.1	0.89±0.0	45.9±1.3	14.0±0.1
DA3 cam. token	0.04±0.0	0.58±0.0	0.53±0.0	0.87±0.0	0.91±0.0	40.9±1.4	12.2±0.4
Pi3X avg. feat.	0.07±0.0	0.38±0.0	0.52±0.0	0.52±0.0	0.75±0.0	38.5±1.5	19.8±0.1
Pi3X avg. regs.	0.08±0.0	0.45±0.0	0.53±0.0	0.61±0.0	0.77±0.0	42.6±0.6	12.7±0.3
baselines							
DeltaTok	0.29±0.0	0.35±0.0	0.65±0.0	**0.69**±0.0	0.75±0.0	**57.8**±0.7	13.5±0.2
AVG-pool	0.96±0.0	0.93±0.0	1.04±0.0	0.78±0.0	0.91±0.0	15.3±0.5	2.3±0.0
`CLS`	1.16±0.0	0.78±0.0	1.09±0.0	0.90±0.1	1.05±0.0	19.0±0.2	14.9±0.2
SMM	**0.16**±0.0	**0.19**±0.0	**0.59**±0.0	0.87±0.1	**0.71**±0.0	52.7±1.2	**23.1**±1.3

supervised-model performance on DL3DV probing and remains competitive on CameraBench. Furthermore, SMM performs best on two out of three object-motion probes, Kubric object motion and static YouTubeVOS, while being on par with DA3 on static DAVIS2017, suggesting that structured feature prediction recovers motion-relevant information not fully captured by supervised descriptors alone.

Primary tokens generalize to object-centric action prediction. On SSv2-110k, SMM improves over `CLS`, AVG-pool as well as DeltaTok by 8.2 to 20.8 p.p., and remains above the best supervised descriptor, Pi3X average-feature concatenation, even though concatenation doubles the probe's input dimension.

SMM supports task-specific nonlinear adaptation. To evaluate SMM beyond frozen linear probing, we fine-tune its primary motion extractor together with a CameraBench classification head, while keeping the rest frozen. This increases CameraBench performance from 52.7 to 60.7 mAP, outperforming fine-tuned DeltaTok at the same 224×224 resolution (57.6 mAP) and performing on par with the 60.0 mAP reported for full supervised fine-tuning of Qwen2.5-VL-7B (Lin et al., 2025b). Full details are given in Sections A.8 and B.3.

3.4 Analysis of SMM

Utilizing temporal context. Table 3 shows that SMM benefits from longer context on temporally coherent benchmarks: Kubric camera/object MSE improves from 0.26/0.22 at $T=2$ to 0.16/0.19 at $T=4$, while SSv2-110k increases monotonically to 23.1% at $T=7$, beyond the training range. The multi-frame baseline also benefits from additional frames but remains below SMM even when its maximum context is compared with SMM at $T=2$. Thus, the gains of SMM cannot be explained by access to additional frames alone. Improvements are less monotonic for more complex motion (see Section B.4).

Structured motion. A key goal of SMM is to produce *structured* motion latents whose specialization follows the dominant motion regime. We test this by swapping which token is used for each probing task, and find that the learned roles support the primary/residual interpretation. Results are given in Table 5. On dynamic-camera datasets, p is substantially better than r for camera-motion prediction, while r performs best for Kubric object motion, capturing motion left unexplained by the primary component. Under approximately static cameras, object motion becomes dominant, and p performs best on static YouTubeVOS. Similarly, on SSv2-110k, where our camera-motion analysis indicates low inter-frame camera motion, p performs best for action recognition. Static DAVIS2017 is an exception with high variability across seeds for the residual token (0.8 ± 0.15, see Section B.5). Concatenating the two tokens leads to similar performance, indicating that task-relevant information is primarily encoded in one of the two tokens. A notable exception is Kubric object motion, where predicting object translation in world coordinates can benefit from both the residual object-motion signal and the primary camera-motion component.

Table 3: **SMM utilizes temporal context to aggregate motion information.** We compare a single frame pair ($T=2$) against the maximum context for Kubric and CameraBench, and evaluate on sequences longer than the training range of $T=5$ on SSv2-110k. The baseline is the AVG-pooled feature-difference descriptor averaged over frame pairs. We report MSE on standardized frame-to-frame pose deltas for regression on Kubric, mAP in % for CameraBench and accuracy in % for SSv2-110k.

method	Kubric cam. ↓		Kubric obj. ↓		CameraBench ↑	
	T=2	T=4	T=2	T=4	T=2	T=5
baseline	0.96	0.87 -0.09	0.93	0.72 -0.21	15.1	15.9 +0.8
SMM	0.26	0.16 -0.10	0.22	0.19 -0.03	46.6	52.7 +6.1

method	SSv2-110k ↑					
	T=2	T=3	T=4	T=5	T=6	T=7
baseline	2.4	5.9 +3.5	8.4 +2.5	10.9 +2.5	11.7 +0.8	13.3 +1.6
SMM	16.7	18.6 +1.9	20.4 +1.8	21.6 +1.2	22.5 +0.9	23.1 +0.6

Table 4: **SMM works across backbones.** We report DL3DV MSE on standardized frame-to-frame pose deltas and CameraBench mAP / SSv2-110k accuracy in %. MAE (He et al., 2022), SigLIP (Zhai et al., 2023), and DINOv3 (Siméoni et al., 2025) use the base variant with patch size 16. DINOv2 with registers (Darcet et al., 2024) (DINOv2r) uses patch size 14.

backbone	DL3DV↓	Camera Bench ↑	SSv2 110k↑
MAE	0.61	52.3	14.1
SigLIP	0.64	45.5	11.6
DINOv3	0.58	54.4	27.0
DINOv2r	0.59	52.7	23.1

Table 5: **SMM leads to structured motion representations where p and r specialize** in primary and residual motion under linear motion/action probing. We report standardized-target MSE, macro mAP in % for CameraBench and accuracy in % for SSv2-110k. Best single-token results are in **bold**, matched representation/target pairs are in green, and the last row is an upper-bound probe using concatenated tokens.

token	Regression (MSE ↓)					Classification (mAP/acc. ↑)	
	Kubric camera	Kubric object	DL3DV	Static DAVIS2017	Static YouTubeVOS	Camera Bench	SSv2-110k
primary p	**0.16**	0.21	**0.59**	0.87	**0.71**	**52.7**	**23.1**
residual r	0.28	**0.19**	0.62	**0.80**	0.79	31.5	12.0
concat. $[p,r]$	0.16	0.14	0.56	0.81	0.71	54.1	22.1

Counterfactual composition. We directly test whether SMM's tokens support counterfactual control of predicted motion, rather than merely motion-label decoding. We extract p and r from different camera/object motion variants of the same Kubric scene, transfer the tokens to a different scene, and retrieve the predicted future among 36 counterfactual camera-object motion combinations. SMM retrieves the correct joint future in 53.1% of cases, compared with 14.9% for the capacity-matched unstructured two-token model and 2.8% chance. Camera motion is recovered similarly by both models (90.0% vs. 89.1%), while object recovery differs substantially (58.5% vs. 16.7%), the latter matching random performance. This shows that SMM's sequential residual structure preserves independently transferable residual motion (see Section B.7 for further details).

Motion magnitude resolution. We further test how finely the residual token resolves object-motion magnitude. In controlled Kubric scenes, cross-scene nearest-neighbor retrieval over 17 object-motion magnitudes yields 2.51 px MAE, with half of predictions within 1.55 px of the target (see Section B.7).

3.5 Ablation Studies

We summarize the main ablation findings below, with full component ablations in Appendix Table 16.

Weak anchors and sequential extraction promote specialization. Removing either weak scene-level constraint weakens the primary/residual structure. Joint extraction or a single stage with two tokens remain competitive on dominant-motion probes but fail to produce a complementary organization.

Non-causal prediction improves specialization. A causal variant using only past features to extract motion tokens weakens the intended p/r specialization, supporting SMM's observed-transition design.

Intermediate features and real data improve performance. Intermediate encoder features outperform final-layer features alone and unlabeled real videos improve generalization beyond synthetic training. MSE improves Kubric object-motion probing significantly while $\mathcal{L}_1$ is slightly stronger on some real-video probes.

Static-camera regularization balances probes. Varying λ_{reg} reveals a trade-off between Kubric object-motion probing and SSv2-110k action recognition. We use $\lambda_{\text{reg}}=0.5$ as a balanced default.

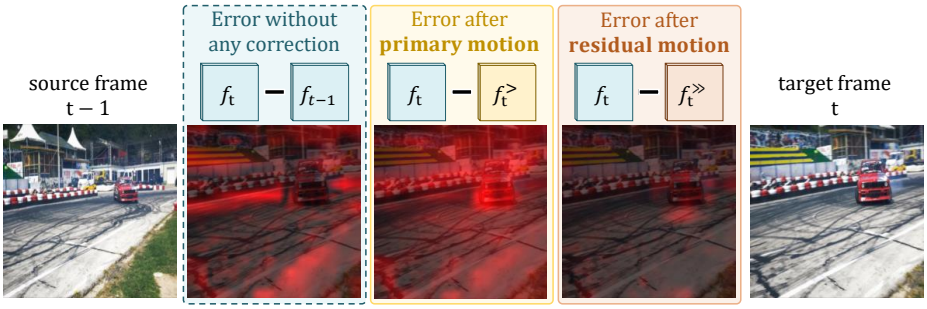


Figure 3: **Stage-wise structured feature prediction.** We visualize feature-map error to the target after primary and residual compensation on a sample from DAVIS2017. Errors are highlighted in red. The primary stage corrects the dominant global change, including the camera-induced movement of the edges of the track. Notably, primary motion compensation increases the error around the independently moving car. The residual stage reduces this localized error.

SMM works across pretrained backbones. Table 4 shows that SMM can be applied to diverse image encoders. DINOv2/3-based backbones perform best overall.

3.6 Qualitative evaluation

We complement probing with qualitative analyses of whether the two prediction stages explain complementary changes and whether motion tokens can be reused beyond the observed transition.

Stage-wise feature compensation. Figure 3 shows that the primary stage explains the dominant global transformation, while residual compensation reduces localized errors around the independently moving object.

Motion extrapolation. Repeatedly applying the last observed motion token yields plausible short-horizon feature extrapolations (Figure 4), but predictions drift over longer horizons. This suggests that the token captures local temporal motion but does not support reliable long-range extrapolation without new observations.

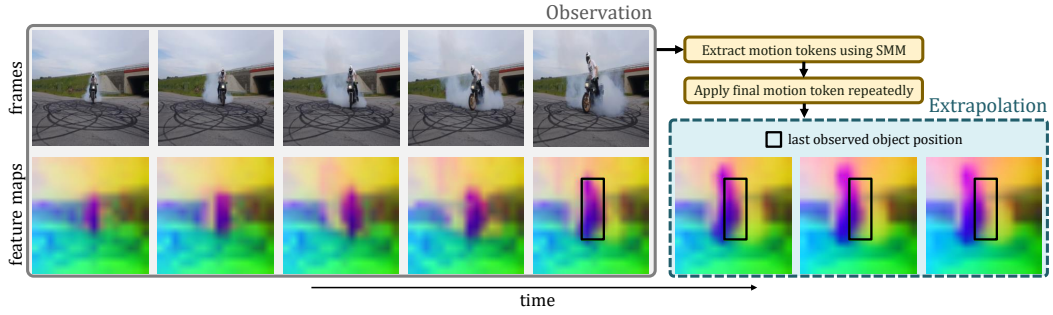


Figure 4: **Motion extrapolation.** We repeatedly apply the last motion token from the observed frames to predict future feature maps on a sample from the DAVIS2017 test split. Visualization is done by PCA projection to RGB after normalization. Black boxes mark the last observed ground-truth object position, which is repeated for the extrapolated feature maps. The extrapolated object features follow the object's motion beyond its last observed position over short horizons and drift at longer horizons.

Latent motion swapping. Applying motion tokens from a source clip to a target clip with a different object induces consistent short-horizon feature-space motion (Figure 5). This indicates that the learned tokens can parameterize local and generalizable transformations across videos, although quality decreases over longer horizons. We quantify this compositional behavior under controlled interventions in Section 3.4.

4 Related work

Self-supervised image encoders. Self-supervised image encoders such as DINOv2 (Oquab et al., 2023), AIM (El-Nouby et al., 2024), and MAE (He et al., 2022) provide strong frozen visual features. These features

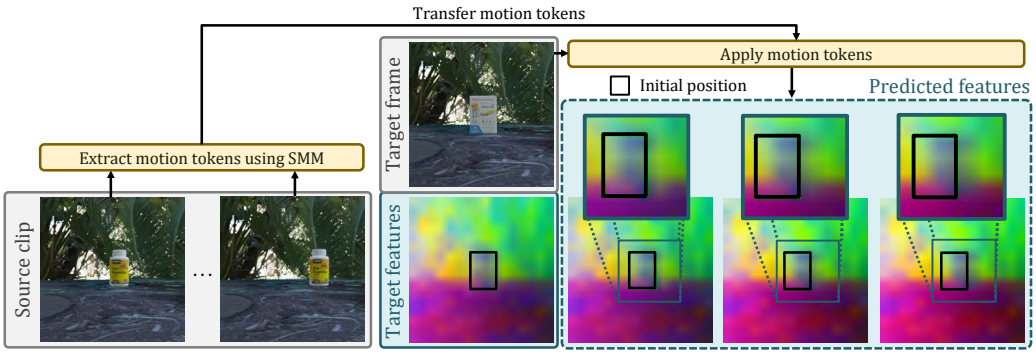


Figure 5: **Latent motion swapping.** We apply motion tokens from a source clip to the feature map of a target clip with a different object, both from the Kubric test set, and visualize the resulting features by PCA projection to RGB after normalization. Black boxes mark the initial object position. The target feature map follows the transferred motion over short horizons and drifts at longer horizons.

contain useful 3D cues (El Banani et al., 2024) and support recent reconstruction systems (Wang et al., 2025b; Lin et al., 2025a; Wang et al., 2025d; Luo et al., 2026), but do not directly expose temporal or motion structure.

Self-supervised video encoders. Self-supervised video encoders learn general, spatio-temporal representations from unlabeled videos often through masked video modeling (Tong et al., 2022; Wang et al., 2023; Carreira et al., 2024; Zoran et al., 2026) or latent predictive objectives (Bardes et al., 2024; Assran et al., 2025; Mur-Labadia et al., 2026; Wang et al., 2025a). In contrast, SMM keeps an image backbone frozen and learns explicit primary/residual motion tokens on top.

Training image encoders on videos. Several works adapt image-pretrained models using temporal consistency (Salehi et al., 2025; 2023) or correspondence learning (Gupta et al., 2023). Others train strong image encoders directly from long, uncurated video (Venkataramanan et al., 2024). We instead keep the image backbone frozen and learn structured motion tokens on top.

3D reconstruction models. Models such as DUSt3R (Wang et al., 2024), VGGT (Wang et al., 2025b), and CUT3R (Wang et al., 2025c) estimate camera pose, depth, point maps, and related geometry using strong 3D supervision, with recent extensions targeting dynamic scenes and 4D reconstruction (Zhang et al., 2025; Hu et al., 2025). In contrast, SMM uses much weaker supervision to remodel frozen image features into primary/residual motion tokens.

Latent action and world models. Latent action and world models predict future frames or pretrained features from video without dense annotations (Kerssies et al., 2026; Baldassarre et al., 2025; Garrido et al., 2026). Prior work represents temporal change with spatial future tokens or compact latent actions (Baldassarre et al., 2025; Garrido et al., 2026; Kerssies et al., 2026; Boduljak et al., 2025; Zhou et al., 2025). SMM instead explicitly structures feature-space motion into primary and residual components.

5 Conclusion

We studied whether structured motion representations could be extracted from frozen pretrained image backbones and proposed the Structured Motion Model (SMM). SMM learns structured primary/residual motion latents through feature prediction, using weak scene-level supervision on synthetic videos together with unlabeled real data. Our results suggest that frozen image backbones can be remodeled into structured motion representations, enabling interpretable temporal abstractions to emerge without full video pretraining while remaining competitive with substantially larger supervised geometry models and delivering strong task-specific fine-tuning performance.

The learned tokens specialize according to scene motion and support counterfactual recombination across scenes. The primary token captures camera motion in dynamic-camera videos and object motion in static-camera videos, while the residual token captures remaining object-centric movement when primary motion explains camera-induced change. This mirrors iterative coarse-to-fine refinement in other domains, from optical flow estimation to diffusion models, and highlights residual compensation as a useful inductive bias.

6 AI DISCLOSURE

We used generative AI tools to assist with drafting and refining parts of the manuscript and with method implementation and writing or editing software code. Generative AI was also used to create part of the illustrative scene visualization in Figure 1 (the moving-camera scene containing the teddy bear and plant). We did not use generative AI for any other research tasks requiring disclosure under the ICLR policy. Tasks concerning mathematical claims and proofs are not applicable to this work. All AI-assisted text, code, and visual content were reviewed and verified by the authors, who take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

7 REPRODUCIBILITY STATEMENT

We provide architecture and training details in Sections 2 and 3.2, and detailed dataset generation, evaluation splits, probe training, baseline evaluation, and visualization protocols in Appendix A. Additional ablations and controlled evaluations are provided in Appendix B, information on used assets and licenses in Appendix D, and computational resources in Appendix E. Code will be made available.

REFERENCES

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Federico Baldassarre, Marc Szaffraniec, Basile Terver, Vasil Khalidov, Francisco Massa, Yann LeCun, Patrick Labatut, Maximilian Seitzer, and Piotr Bojanowski. Back to the features: Dino as a foundation for video world models. *arXiv preprint arXiv:2507.19468*, 2025.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024.

Gabrijel Boduljak, Yushi Lan, Christian Rupprecht, and Andrea Vedaldi. VFMF: World modeling by forecasting vision foundation model features. *arXiv preprint arXiv:2512.11225*, 2025.

Bingyi Cao, Koert Chen, Kevis-Kokitsi Maninis, Kaifeng Chen, Arjun Karpur, Ye Xia, Sahil Dua, Tanmaya Dabral, Guangxing Han, Bohyung Han, et al. TIPSv2: Advancing vision-language pretraining with enhanced patch-text alignment. *arXiv preprint arXiv:2604.12012*, 2026.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019.

João Carreira, Dilara Gokay, Michael King, Chuhan Zhang, Ignacio Rocco, Aravindh Mahendran, Thomas Albert Keck, Joseph Heyward, Skanda Koppula, Etienne Pot, et al. Scaling 4d representations. *arXiv preprint arXiv:2412.15212*, 2024.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*, 2024.

A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. URL `http://lmb.informatik.uni-freiburg.de/Publications/2015/DFIB15`.

Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. Probing the 3d awareness of visual foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21795–21806, 2024.

Alaaeldin El-Nouby, Michal Klein, Shuangfei Zhai, Miguel Ángel Bautista, Vaishaal Shankar, Alexander T Toshev, Joshua M Susskind, and Armand Joulin. Scalable pre-training of large autoregressive image models. In *International Conference on Machine Learning*, pp. 12371–12384, 2024.

Enrico Fini, Mustafa Shukor, Xiujun Li, Philipp Dufter, Michal Klein, David Haldimann, Sai Aitharaju, Victor G Turrisi da Costa, Louis Béthune, Zhe Gan, et al. Multimodal autoregressive pre-training of large vision encoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9641–9654, 2025.

Quentin Garrido, Tushar Nagarajan, Basile Terver, Nicolas Ballas, Yann LeCun, and Michael Rabbat. Learning latent action world models in the wild. *arXiv preprint arXiv:2601.05230*, 2026.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pp. 5842–5850, 2017.

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

Agrim Gupta, Jiajun Wu, Jia Deng, and Fei-Fei Li. Siamese masked autoencoders. *Advances in Neural Information Processing Systems*, 36:40676–40693, 2023.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision*, pp. 289–305. Springer, 2024.

Yu Hu, Chong Cheng, Sicheng Yu, Xiaoyang Guo, and Hao Wang. VGGT4D: Mining motion cues in visual geometry transformers for 4d scene reconstruction. *arXiv preprint arXiv:2511.19971*, 2025.

E. Ilg, T. Saikia, M. Keuper, and T. Brox. Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In *European Conference on Computer Vision (ECCV)*, 2018. URL `http://lmb.informatik.uni-freiburg.de/Publications/2018/ISKB18`.

Tommie Kerssies, Gabriele Berton, Ju He, Qihang Yu, Wufei Ma, Daan de Geus, Gijs Dubbelman, and Liang-Chieh Chen. A frame is worth one token: Efficient generative world modeling with delta tokens. *arXiv preprint arXiv:2604.04913*, 2026.

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

Zhiqiu Lin, Siyuan Cen, Daniel Jiang, Jay Karhade, Hewei Wang, Chancharik Mitra, Tiffany Ling, Yuhan Huang, Sifan Liu, Mingyu Chen, et al. Towards understanding camera motions in any video. *arXiv preprint arXiv:2504.15376*, 2025b.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. DL3DV-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22160–22169, 2024.

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

Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2023.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

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:1704.00675*, 2017.

Mohammadreza Salehi, Efstratios Gavves, Cees GM Snoek, and Yuki M Asano. Time does tell: Self-supervised time-tuning of dense image representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16536–16547, 2023.

Mohammadreza Salehi, Shashanka Venkataramanan, Ioana Simion, Efstratios Gavves, Cees GM Snoek, and Yuki M Asano. MoSiC: Optimal-transport motion trajectory for dense self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6541–6551, 2025.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pp. 402–419. Springer, 2020.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35: 10078–10093, 2022.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Shashanka Venkataramanan, Mamshad Nayeem Rizve, Joao Carreira, Yuki M Asano, and Yannis Avrithis. Is imagenet worth 1 video? learning strong image encoders from 1 long unlabelled video. In *The Twelfth International Conference on Learning Representations*, 2024.

Alex N Wang, Christopher Hoang, Yuwen Xiong, Yann LeCun, and Mengye Ren. PooDLe: Pooled and dense self-supervised learning from naturalistic videos. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025b.

Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. VideoMAE v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14549–14560, 2023.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 10510–10522, 2025c.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20697–20709, 2024.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. pi3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025d.

Philippe Weinzaepfel, Vincent Leroy, Thomas Lucas, Romain Brégier, Yohann Cabon, Vaibhav Arora, Leonid Antsfeld, Boris Chidlovskii, Gabriela Csurka, and Jérôme Revaud. CroCo: Self-supervised pre-training for 3d vision tasks by cross-view completion. *Advances in Neural Information Processing Systems*, 35: 3502–3516, 2022.

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 *Proceedings of the European conference on computer vision (ECCV)*, pp. 585–601, 2018a.

Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. YouTube-VOS: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018b.

Lihe Yang, Shang-Wen Li, Yang Li, Xinjie Lei, Dong Wang, Abdelrahman Mohamed, Hengshuang Zhao, and Hu Xu. In pursuit of pixel supervision for visual pre-training. *arXiv preprint arXiv:2512.15715*, 2025.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

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 *The Thirteenth International Conference on Learning Representations*, 2025.

Gaoyue Zhou, Hengkai Pan, Yann Lecun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *International Conference on Machine Learning*, pp. 79115–79135. PMLR, 2025.

Daniel Zoran, Nikhil Parthasarathy, Yi Yang, Drew A Hudson, João Carreira, and Andrew Zisserman. Recurrent video masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17744–17755, 2026.

A Dataset and Evaluation Details

For regression tasks, motion-magnitude balancing denotes retaining high-motion samples and subsampling low-motion samples as described in Section A.3.

A.1 Kubric generation

We generate our synthetic training and evaluation data with Kubric (Greff et al., 2022), starting from the MOVi-E setup[2]. Dedicated controlled Kubric evaluations for counterfactual composition and motion-magnitude resolution use the same rendering framework and are described in Section B.7. Each sample is generated from a unique random seed. Additionally, training and evaluation samples use disjoint subsets of objects and backgrounds. We render short clips of 10 frames at 2 fps and resolution 224×224. We use three scene configurations: static scene with moving camera, dynamic scene with static camera, and dynamic scene with moving camera. Static scenes contain no foreground objects, whereas dynamic scenes contain a single moving object.

For dynamic scenes, we disable gravity and assign the object a randomly sampled initial velocity. This produces approximately unbiased object motions that are not dominated by falling trajectories. For moving-camera scenes, the camera follows a linear trajectory while continuously looking at a fixed point above the world origin. This yields controlled ego-motion while keeping the scene center approximately in view.

We apply several pre-rendering validity checks to ensure that dynamic objects are sufficiently visible. First, we require the selected object to have a minimum size of one simulator unit, measured as the shortest side of its 3D bounding box. If the sampled object is too small, we resample the object. If no valid object is found after 100 attempts, the sample is rejected. Second, for all frames, we require at least 50% of the projected 2D bounding box of the object to lie inside the image. Third, we reject samples in which the object lies

[2] https://github.com/google-research/kubric/tree/main/challenges/movi

Table 6: **Evaluation tasks.** Summary of the probing tasks used for each evaluation dataset.

	Kubric	DL3DV	CameraBench
Task			
target	camera / object motion	camera motion	15 camera-motion tasks
target format	6-DoF pose / 3D translation	6-DoF pose	binary labels
probe type	regression	regression	classification
metric	norm. MSE	norm. MSE	macro mAP
Representation			
probed token	p / r	p	p
temporal target	frame-pair motion	frame-pair motion	clip-level motion
Split and sampling			
split	generated train/test	video-level 70/30	train 80/20, official test
train sequences / video	1	3	1
test sequences / video	1	3	1
balancing	motion-mag. subsampling	motion-mag. subsampling	-
	Static DAVIS2017	Static YouTubeVOS	SSv2-110k
Task			
target	object displacement	object displacement	action class
target format	2D displacement	2D displacement	action label
probe type	regression	regression	classification
metric	norm. MSE	norm. MSE	top-1 acc.
Representation			
probed token	p	p	p
temporal target	frame-pair motion	frame-pair motion	clip label
Split and sampling			
split	official train/test	video-level 70/30	official train/val
train sequences / video	3	3	1
test sequences / video	3	3	1
balancing	motion-mag. subsampling	motion-mag. subsampling	class downsampling

behind the camera, i.e., when its maximum depth satisfies $\texttt{zmax} < 0$. These checks are performed before rendering for efficiency. Invalid samples are discarded and regenerated with a new seed.

Because of the low resolution, generation can be performed without a GPU and takes approximately 12 seconds per valid sample. Invalid samples take only 0.3 seconds due to the pre-rendering checks. Approximately 90% of samples with dynamic objects pass the validity checks.

A.2 Static-camera object-motion benchmarks

Inputs. We build a real-world object-motion benchmark with approximately static camera from DAVIS2017 and YouTubeVOS. We start from videos that contain both RGB frames and object annotations, and keep only timestamps for which both modalities are available. For each scene, we summarize the annotation masks by recording per-frame visible object area and image-plane centroid.

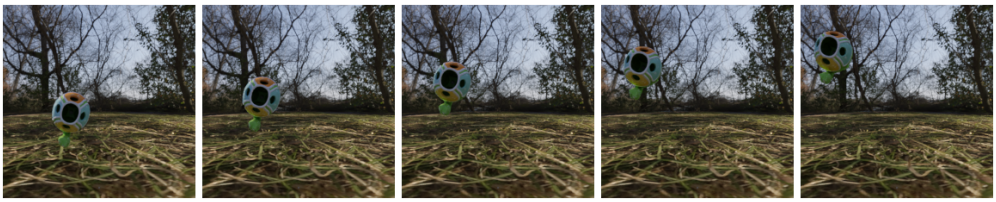
Camera-motion filtering. To identify near-static-camera clips, we estimate camera motion from sampled video frames using VGGT-based camera-pose prediction, which provides per-interval camera rotation and translation estimates. We then construct candidate windows of length $T=4$ at 2 fps and retain only those windows whose maximum local camera motion across consecutive frame pairs stays below fixed translation and rotation thresholds. Specifically, we require the maximum relative rotation between consecutive frames to be below 2° and the maximum translation between camera centers to be below 0.05.

Object filtering. Among the near-static windows, we further keep only samples containing sufficiently large objects, requiring objects to span more than 15 pixels and to move by at least 0.05 in normalized centroid displacement.

Motion targets. For each retained window, we compute continuous object motion from the selected object's mask sequence. We extract the object centroid in each frame and define motion as the frame-to-frame centroid displacement in image coordinates. These displacements are normalized by image width and height, yielding continuous horizontal and vertical motion components in relative image-plane units.



(a) Moving camera, static scene.



(b) Static camera, dynamic scene.



(c) Moving camera, dynamic scene.

Figure 6: **Kubric examples.** Example sequences from our generated Kubric training data. The three subsets provide weak scene-level supervision for disentangling primary and residual motion: moving-camera/static-scene sequences contain only camera-induced motion, static-camera/dynamic-scene sequences contain only object motion, and moving-camera/dynamic-scene sequences combine both sources of motion. Examples are shown at 1 fps.

A.3 Dataset splits and sampling

CameraBench training annotations. The released CameraBench training set is QA-formatted rather than organized into the 15 binary evaluation tasks. We retain only questions that exactly match the atomic Yes/No question templates corresponding to the official tasks, map their answers to binary targets, and discard compound or unrelated questions. Duplicate labels for the same video and task are collapsed.

Dataset-specific probing splits. SSv2 uses the official train/validation split filtered by videos that are too short for the required clip sampling. For CameraBench, we use the released training split for probe fitting and the official test split for evaluation. We deterministically split the training release 80/20 into probe-training and probe-validation sets and evaluate the selected probes once on the complete official test split. We extract deterministic centered five-frame clips at 0.5s spacing, using uniform sampling for videos too short for the full window. For DL3DV, we use the `1K` to `7K` subsets and form splits at the video level to avoid leakage, assigning 70% of videos to training and 30% to testing. We do not use the official test split because it is too small for our probing setting. To further increase the number of probing samples, we sample three sequences per video for both training and testing. Videos in the probing test split are excluded from unlabeled SMM training. For static YouTubeVOS, we also use a video-level pseudo split because the official test split does not provide object segmentations at sufficient temporal resolution. Similar to DL3DV, we sample three sequences per training and test video. For static DAVIS2017, we use the official train/test split and also sample three sequences per video. For the static-camera object-motion probes, we train on the combined static DAVIS2017 and static YouTubeVOS training sets to increase probe-training data, while reporting evaluation results separately on each benchmark. Kubric uses predefined train and test sets for the moving-camera/dynamic-scene setting. The actual probing splits are further balanced by regression bins, with bin edges computed on the training split and reused on the test split.

Balancing. SSv2-110k is class-balanced by downsampling each class to the smallest class count. CameraBench retains the official binary-label distribution. For regression probing, we downsample low-motion samples

by constructing a tail-aware subset. We retain all samples above the 75th percentile of motion magnitude and randomly subsample the remaining lower-motion samples to the same count. This yields equal numbers of high- and lower-motion samples and prevents regression probes from being dominated by near-static examples.

A.4 PROBE TARGETS

Regression tasks. For regression, we train one linear probe per probing setting, where each probe jointly predicts the full target vector for that setting. Camera-motion regression predicts the 6D vector $[v_x, v_y, v_z, \text{yaw}, \text{pitch}, \text{roll}]$. This is used for DL3DV and Kubric camera-motion evaluation. Kubric object-motion regression predicts the 3D translation vector $[v_x, v_y, v_z]$, since objects do not rotate in our Kubric setup. The relative pose is computed between frame $t{+}1$ and frame t, expressed in the coordinate system of frame $t{+}1$. Static DAVIS2017 and static YouTubeVOS object-motion regression predict the 2D image-plane centroid displacement $[\Delta x, \Delta y]$ in normalized image coordinates.

Classification tasks. For classification, we train one linear probe for each label set. SSv2-110k uses a single full-class action-recognition probe. CameraBench uses 15 independent binary probes, one for each official camera-motion task: move down, move up, move in, move out, move left, move right, pan left, pan right, roll clockwise, roll counterclockwise, tilt down, tilt up, zoom in, zoom out, and static.

A.5 PROBE TRAINING

Our evaluation uses linear probes trained on frozen motion representations. Each probe consists of a single linear layer on top of the corresponding motion token and is trained for 100 epochs with SGD, momentum 0.9, cosine learning-rate decay, and evaluation after every epoch. For all datasets except CameraBench, following prior work (Oquab et al., 2023), we train probes with multiple learning rates in parallel. Specifically we use:

$$\text{lr} \in \{1,2,5\} \times 10^{\{-5,-4,-3,-2,-1\}} \cup \{3 \times 10^{-1}\}, \tag{6}$$

which corresponds to the union of the learning rates used in the DINOv2 paper and its official implementation. We linearly scale these learning rates from a reference batch size of 256 to batch size 64. For CameraBench, we train 15 independent binary linear probes for 100 epochs with SGD, momentum 0.9, weight decay 0.05, batch size 16, and a reference learning rate of 10^{-3} linearly scaled from batch size 256. We use cosine learning-rate decay and select each class head independently by its best validation AP. For regression probes, we additionally apply gradient clipping with norm 5 to improve optimization stability. Depending on the task, we probe either the primary motion token p or the residual motion token r. The choice is stated in the corresponding result table.

As we train many probes, we improve the efficiency of probe training by using no input augmentations and precomputing the motion features before training the linear probes.

A.6 METRICS AND PROBE SELECTION

For SSv2-110k, we report top-1 accuracy. For CameraBench, we report per-class AP and macro mAP across the 15 binary tasks. For regression, we report mean squared error on targets normalized with the training-set mean and standard deviation. For non-CameraBench probes, we report the best held-out metric over probe training and across the parallel learning-rate sweep. For CameraBench, each binary head is selected independently by validation AP and the selected heads are evaluated once on the official test set. This reduces sensitivity to the exact probe optimization hyperparameters when comparing many probes across datasets, targets, and representations. All methods and baselines use the same protocol.

A.7 DELTATOK EVALUATION

We evaluate DeltaTok (Kerssies et al., 2026) as a strong self-supervised baseline using the released Kinetics-700 (Carreira et al., 2019) ViT-B model and the same evaluation protocol as for SMM. We follow its inference protocol by, unless stated otherwise, resizing frames to $512{\times}512$ and extracting the final delta token for each clip as its motion descriptor.

DeltaTok and SMM are not trained under matched compute or input-resolution budgets. DeltaTok uses $512{\times}512$ inputs, DINOv3 (Siméoni et al., 2025) features, and is trained for 1.6M iterations with an effective batch size of 1024. In comparison, SMM uses $224{\times}224$ inputs, DINOv2-with-registers (Darcet et al., 2024) features by default, and is trained for 200k iterations with batch size 128. Thus, DeltaTok uses $8\times$ more optimization iterations and an $8\times$ larger effective batch size, in addition to operating at substantially higher spatial resolution.

A.8 CameraBench supervised fine-tuning

To evaluate task-specific nonlinear adaptation, we additionally fine-tune representations on the 15 CameraBench binary tasks using a shared 15-output linear classification head. For SMM, we fine-tune the primary motion extractor ϕ_p together with the classification head, while keeping the image encoder, primary predictor, and complete residual pathway frozen. For DeltaTok, we analogously fine-tune its motion extractor and classification head while keeping its DINOv3 backbone frozen. To match the fine-tuning budget, both SMM and DeltaTok use 224×224 inputs.

We train on all CameraBench training videos carrying at least one of the 15 atomic binary-motion labels for 10 epochs using masked binary cross-entropy. We use AdamW with batch size 8, weight decay 0.01, a motion-extractor learning rate of 10^{-5}, and a classification-head learning rate of 10^{-3}. We use one warmup epoch followed by cosine decay. Training uses random valid five-frame windows at 0.5 s spacing, while evaluation uses the deterministic centered window described in Section A.3. We evaluate the final epoch once on the complete official test set and report per-class AP and macro mAP.

A.9 Supervised model evaluation

We evaluate supervised 3D and geometry-oriented models as frozen feature extractors. For each model, we use its default inference preprocessing, run inference on the same frame sequences as in our probing datasets, and train linear probes using the same protocol as for SMM. We derive motion descriptors from consecutive frames using the same temporal concatenation or difference baselines used for frozen image features.

For VGGT, we use `VGGT-1B`. Frames are converted to RGB, resized to a target width 518 while preserving aspect ratio, rounded to a height divisible by 14, center-cropped to 518×518, and scaled to $[0,1]$. We use either the camera token or mean-pooled patch tokens from the final aggregated token map. For Depth Anything 3, we use `DA3-GIANT-1.1` with the official inference pipeline and extract either camera-token or mean-pooled patch-token features from the final backbone output. For Pi3X, we resize RGB inputs with preserved aspect ratio to the largest resolution with area at most $255{,}000$ pixels and dimensions divisible by 14, and extract the output features of the decoder. The final descriptor is obtained either by averaging the features of the patch tokens or, as Pi3X does not have a dedicated camera token, by averaging the 5 registers.

A.10 Sample visualization details

For qualitative sample visualization, we project dense feature maps to RGB using a sample-specific PCA. For a given sample, we collect the feature tokens from all frames of the sequences being compared, flatten them into a single feature matrix, and normalize them. The PCA is fit jointly on all extracted feature maps.

We retain the first three principal components and map them to the three RGB channels. After projection, each component is scaled independently to $[0,1]$ using the 1st and 99th percentiles computed on the PCA fit set, and the resulting token grid is reshaped to the spatial feature-map layout and bilinearly upsampled to the image resolution for display.

Pixel-wise error maps are the L2 distance between predicted and ground-truth feature maps, computed in the projected space. These distances are normalized with a shared 99th-percentile scale across all compared maps, converted to a heatmap, and overlaid on the corresponding RGB video frames.

B Additional results

B.1 SSv2 camera motion analysis

To motivate the choice of token to use for SSv2-110k action recognition probing, we analyze the amount of camera motion in SSv2 clips. We randomly sample 5k SSv2 clips, estimate frame-to-frame camera motion using VGGT (Wang et al., 2025b) and compute the average per-step camera rotation and translation. To contextualize these values, we apply the same procedure to the CameraBench dataset and aggregate the camera motion separately for samples annotated as `slow-speed`, `regular-speed`, or `fast-speed`. The `slow-speed` subset includes not only clips with little camera motion but also all samples with a static camera.

SSv2 camera motion lies between the slow- and regular-speed CameraBench subsets and is closer to the slow-speed subset in terms of rotation. This suggests that many SSv2 clips contain limited inter-frame camera motion. We therefore follow a similar reasoning as for the static-camera DAVIS2017 and YouTubeVOS object-motion benchmarks (see Section 3.2). When camera-induced change is small, action-related object motion is expected to be a dominant source of temporal change, and we probe the primary token p for SSv2 action recognition.

Table 7: **Camera-motion statistics for SSv2 and CameraBench.** We report average frame-to-frame camera motion estimated with VGGT. Rotation is reported in degrees per second while translation is in VGGT's normalized scene-scale units per second. SSv2 exhibits lower camera motion than regular- and fast-speed CameraBench clips and is closer to the slow-speed subset in rotation.

	SSv2	CameraBench		
		`slow-speed`	`regular-speed`	`fast-speed`
rotation	3.89	1.57	7.77	32.89
translation	0.08	0.03	0.13	0.62

Table 8: **Baseline descriptor ablations.** We compare descriptor choices for frozen DINOv2-with-registers representations and strongly supervised representations. For DINOv2 with registers, we evaluate `CLS` tokens and average-pooled patch features using either concatenation or feature differences across consecutive frames. Multi-frame variants (m.f.) average frame-pair descriptors over time. For VGGT and Depth Anything 3, we compare average-pooled feature maps and camera tokens using either concatenation or differences across consecutive frames. Since Pi3X does not have a camera token, we use the average of its registers as an alternative and otherwise follow the same setup. We report MSE on standardized regression targets, CameraBench mAP in % and SSv2-110k accuracy in %. The best results for each model and baseline type are highlighted in **bold**.

Method	Regression (MSE ↓)					Classification (mAP/acc. ↑)	
	Kubric cam	Kubric obj	DL3DV	Static DAVIS2017	Static YouTubeVOS	Camera Bench	SSv2-110k
DINOv2 with registers baselines							
AVG-pool concat	1.05±0.0	0.73±0.0	1.06±0.0	0.96±0.03	0.97±0.0	**19.2**±0.8	**14.1**±0.1
AVG-pool concat, m.f.	0.97±0.0	**0.69**±0.0	1.07±0.0	0.96±0.02	0.95±0.01	18.8±0.7	14.0±0.1
AVG-pool diff	0.96±0.0	0.93±0.0	**1.04**±0.0	**0.78**±0.02	**0.91**±0.0	15.3±0.5	2.3±0.0
AVG-pool diff, m.f.	**0.87**±0.0	0.72±0.0	1.05±0.0	1.02±0.00	0.92±0.0	15.9±1.1	13.3±0.0
`CLS` concat	1.16±0.0	0.78±0.0	1.09±0.0	**0.90**±0.10	1.05±0.01	**19.0**±0.2	14.9±0.2
`CLS` concat, m.f.	1.05±0.0	**0.77**±0.0	1.09±0.0	0.92±0.02	0.98±0.0	18.3±0.6	**15.5**±0.1
`CLS` diff	1.09±0.0	0.96±0.0	1.09±0.0	**0.90**±0.01	**0.93**±0.0	14.5±0.8	3.0±0.0
`CLS` diff, m.f.	**0.96**±0.0	0.80±0.0	**1.07**±0.0	1.01±0.01	**0.93**±0.0	14.8±0.7	12.5±0.1
strongly supervised							
VGGT AVG feat. concat	0.18±0.0	0.42±0.0	0.59±0.0	0.99±0.06	0.95±0.0	48.6±0.4	**15.5**±0.1
VGGT AVG feat. diff	0.09±0.0	**0.32**±0.0	0.56±0.0	0.78±0.10	**0.79**±0.0	24.4±0.5	5.6±0.0
VGGT cam. token concat	0.05±0.0	0.43±0.0	**0.53**±0.0	0.97±0.07	0.98±0.0	**57.9**±0.4	13.5±0.1
VGGT cam. token diff	**0.03**±0.0	0.44±0.0	**0.53**±0.0	**0.71**±0.04	0.81±0.0	38.2±1.9	4.8±0.0
DA3 AVG feat. concat	0.28±0.0	0.52±0.01	0.66±0.01	1.89±0.20	1.65±0.02	**45.9**±1.3	**14.0**±0.1
DA3 AVG feat. diff	0.11±0.0	**0.40**±0.0	0.55±0.0	**0.87**±0.11	**0.89**±0.01	36.3±1.9	5.2±0.1
DA3 cam. token concat	0.18±0.0	0.72±0.01	0.62±0.0	3.53±1.04	1.66±0.06	39.1±3.6	12.2±0.4
DA3 cam. token diff	**0.04**±0.0	0.58±0.0	**0.53**±0.0	**0.87**±0.04	0.91±0.02	40.9±1.4	4.6±0.1
Pi3X AVG feat. concat	0.10±0.0	0.44±0.0	0.54±0.0	0.71±0.01	0.86±0.0	34.7±1.2	**19.8**±0.1
Pi3X AVG feat. diff	**0.07**±0.0	**0.38**±0.00	**0.52**±0.0	**0.52**±0.00	**0.75**±0.0	38.5±1.5	6.2±0.1
Pi3X avg. registers concat	0.33±0.0	0.77±0.0	0.65±0.0	0.95±0.03	0.94±0.0	35.8±2.9	12.7±0.3
Pi3X avg. registers diff	0.08±0.0	0.45±0.0	0.53±0.0	0.61±0.01	0.77±0.0	**42.6**±0.6	5.7±0.1
SMM	0.16±0.0	0.19±0.0	0.59±0.0	0.87±0.1	0.71±0.0	52.7±1.2	23.1±1.3

B.2 Baseline results

We compare different ways to derive motion descriptors from strongly supervised and baseline models in Tables 8 and 9. We use these ablations to choose between concatenation and feature differences for the direct-feature baselines and supervised-model descriptors reported in the main paper. For single-layer DINOv2 features, the best descriptor depends on the pooling strategy and task. Feature differences of AVG-pooled features perform best overall, especially on regression tasks, while concatenating `CLS` tokens of consecutive frames works best for classification tasks. For the supervised models, feature differences generally remain strong on regression tasks, while CameraBench shows no uniform preference and SSv2-110k strongly favors concatenation. On this action recognition benchmark, all three supervised models show the same pattern. Feature differences perform poorly, with the best accuracy reaching only 6.2%, whereas concatenation-based descriptors achieve between 12.2% and 19.8%. A possible explanation is that, unlike the other benchmarks,

Table 9: **Multi-layer direct-feature baselines.** To control for the use of intermediate DINOv2 features, we evaluate direct `CLS` and average-pooled patch descriptors extracted from all 12 backbone blocks. Descriptors are averaged across layers and combined across adjacent frames either by concatenation or feature difference. m.f. indicates multi-frame variants. Even with access to multi-layer features, direct descriptors remain substantially below SMM on all datasets except for static DAVIS2017, which is consistent with the results obtained without multi-layer features (see Table 2). This indicates that the gains are not explained by intermediate-feature aggregation alone. We report MSE on standardized regression targets, mAP in % on CameraBench, and accuracy in % for SSv2-110k. Best results are highlighted in **bold**.

Method	Regression (MSE ↓)					Classification (mAP/acc. ↑)	
	Kubric cam	Kubric obj	DL3DV	Static DAVIS2017	Static YouTubeVOS	Camera Bench	SSv2-110k
DINOv2 with registers, multi-layer baselines							
AVG-pool concat	0.94±0.0	0.66±0.0	1.02±0.0	0.81±0.01	0.94±0.0	16.0±0.9	10.6±0.2
AVG-pool concat, m.f.	0.91±0.0	0.70±0.0	1.04±0.0	0.89±0.02	0.95±0.0	16.0±0.6	12.6±0.1
AVG-pool diff	0.88±0.0	0.83±0.0	1.04±0.0	**0.65**±0.02	0.83±0.0	15.3±0.8	1.9±0.0
AVG-pool diff, m.f.	0.82±0.0	0.60±0.0	1.06±0.0	1.01±0.01	0.91±0.0	16.4±0.5	6.9±0.0
`CLS` concat	1.05±0.0	0.72±0.0	1.05±0.0	0.93±0.02	0.94±0.01	17.5±0.5	13.6±0.3
`CLS` concat, m.f.	1.01±0.0	0.77±0.0	1.06±0.0	0.94±0.0	0.94±0.0	17.2±0.8	14.5±0.4
`CLS` diff	0.98±0.0	0.91±0.0	1.04±0.0	1.00±0.01	0.92±0.0	14.1±0.4	2.0±0.1
`CLS` diff, m.f.	0.88±0.0	0.68±0.0	1.05±0.0	1.02±0.0	0.93±0.0	15.3±0.8	7.5±0.0
SMM	**0.16**±0.0	**0.19**±0.0	**0.59**±0.0	0.87±0.1	**0.71**±0.0	**52.7**±1.2	**23.1**±1.3

Table 10: **Complementarity with frozen DINOv2 features.** We concatenate the task-appropriate SMM token with either the source-frame DINOv2 `CLS` token or AVG-pooled patch features and apply the same linear-probing protocol. We report MSE on standardized regression targets, CameraBench macro mAP in %, and SSv2-110k accuracy in %. Mean and standard deviation are reported over three runs. Best results are highlighted in **bold**.

method	Regression (MSE ↓)					Classification (mAP/acc. ↑)	
	Kubric cam.	Kubric obj.	DL3DV	Static DAVIS2017	Static YouTubeVOS	Camera Bench	SSv2-110k
SMM	**0.16**±0.01	**0.19**±0.01	**0.59**±0.00	**0.87**±0.05	**0.71**±0.02	**52.7**±1.2	**23.1**±1.3
SMM + DINO `CLS`	0.17±0.01	0.21±0.01	0.60±0.00	1.12±0.03	0.78±0.01	50.1±1.9	18.1±0.5
SMM + DINO AVG	**0.16**±0.01	0.20±0.01	**0.59**±0.00	1.12±0.03	0.74±0.02	52.2±1.6	17.7±0.4

SSv2-110k uses action labels rather than motion targets, and these labels may depend on semantic information that is reduced when subtracting features. Finally, Table 9 shows that giving direct `CLS` and AVG-pool baselines access to all 12 DINOv2 blocks improves these baselines but does not close the gap to SMM. Thus, the improvements of SMM are not explained by access to intermediate features alone, but by using structured recurrent prediction to turn these features into motion representations.

We additionally test whether the original frozen-backbone representation provides complementary information to the learned motion tokens. We concatenate the task-appropriate SMM token with either the source-frame DINOv2 `CLS` token or AVG-pooled patch features and apply the same linear-probing protocol. As shown in Table 10, neither backbone descriptor improves performance on any of the seven ProbeMotion tasks, suggesting that these DINOv2 representations do not provide complementary task-relevant information under linear probing.

B.3 CameraBench results

Table 11 reports per-class AP for the principal CameraBench representations compared in Table 2. SMM substantially improves over direct frozen DINOv2 descriptors across most motion classes. DeltaTok and the strongly supervised VGGT camera-token representation achieve higher overall mAP. For additional context, SMM's 52.7 mAP also exceeds all methods reported in CameraBench without CameraBench-specific supervised fine-tuning, including models like MegaSAM (50.1 mAP), Qwen2.5-VL-72B (37.7 mAP), and GPT-4o (36.4 mAP) (Lin et al., 2025b).

Table 11: **Per-class CameraBench linear-probing results.** We report AP in % for all 15 official binary camera-motion tasks and macro mAP. We show SMM together with the direct DINOv2 descriptors used in Table 2, DeltaTok, and the strongest supervised representation on CameraBench from Table 2. Mean and standard deviation are reported over three runs.

Task	SMM p	DINO AVG diff	DINO CLS concat	DeltaTok	VGGT cam. concat
Move Down	28.3±8.0	13.0±6.2	10.4±2.8	22.9±1.9	41.0±5.6
Move Up	13.1±1.1	10.2±4.0	13.1±2.1	23.8±5.1	42.9±2.2
Move In	83.5±2.9	35.2±1.4	61.1±1.4	84.0±0.6	86.4±0.4
Move Out	36.5±0.9	11.4±1.0	14.8±2.6	45.5±0.5	46.5±14.4
Move Left	33.9±1.4	14.0±1.7	14.6±0.9	42.2±3.2	54.3±2.5
Move Right	48.6±2.3	19.3±1.4	24.1±2.7	55.2±6.3	69.9±2.9
Pan Left	83.5±2.4	22.3±2.3	19.8±3.0	80.1±1.1	87.3±0.1
Pan Right	80.6±2.7	17.3±3.0	24.4±0.7	80.9±2.0	82.0±0.5
Roll Clockwise	62.2±1.7	13.1±2.9	11.6±2.5	67.4±2.0	68.6±15.1
Roll Counterclockwise	65.5±4.8	13.5±1.7	16.2±4.6	76.1±5.8	72.1±0.9
Tilt Down	72.7±2.4	9.1±0.4	14.4±4.4	78.0±2.5	67.2±2.5
Tilt Up	74.2±4.2	19.6±2.0	15.9±1.0	70.5±1.1	73.3±4.0
Zoom In	32.2±10.1	12.2±1.1	14.7±2.0	40.2±8.2	19.8±2.1
Zoom Out	56.8±2.0	10.9±1.8	17.9±5.8	76.4±1.2	35.6±2.7
Static	19.3±4.8	9.1±0.2	11.7±0.8	23.8±5.5	21.7±1.7
mAP	52.7±1.2	15.3±0.5	19.0±0.2	57.8±0.7	57.9±0.4

Table 12: **CameraBench supervised fine-tuning.** We report macro mAP in % on the official CameraBench test set. SMM and DeltaTok-224 use 224×224 inputs. Mean and standard deviation are reported over three runs for our evaluations. The Qwen2.5-VL-7B result is reported by CameraBench (Lin et al., 2025b). Best results are highlighted in **bold**.

method	linear probe	supervised fine-tuning
SMM	**52.7**±1.2	**60.7**±1.0
DeltaTok-224	47.6±0.6	57.6±0.8
Qwen2.5-VL-7B (Lin et al., 2025b)	–	60.0

We additionally evaluate task-specific adaptation using the supervised fine-tuning protocol in Section A.8. Table 12 shows that SMM outperforms resolution-matched DeltaTok and performs on par with the full supervised fine-tuning result reported for Qwen2.5-VL-7B in CameraBench (Lin et al., 2025b).

B.4 Temporal context

Tables 13 and 14 provide the full temporal-context results with standard deviations. SMM benefits most from longer context on temporally coherent benchmarks, such as linear Kubric motion and SSv2-110k, where performance improves monotonically even beyond 5 frames, the longest context used during training. The same trend holds for the multi-frame baseline based on average-pooled feature differences, which nevertheless remains below SMM even when using longer context. In contrast, SMM remains stable on datasets with less constrained motion, such as DL3DV, static YouTubeVOS, and static DAVIS2017, where the baseline's performance even deteriorates significantly with increasing temporal context. Overall, additional context is useful when motion is coherent, but not uniformly beneficial for unconstrained real-world videos.

B.5 Structured motion

Table 15 reports the full structured-motion results with standard deviations. The primary and residual tokens specialize according to the dominant motion source. The primary token performs best for camera motion in dynamic-camera videos, for object motion in approximately static-camera videos, and for SSv2-110k action prediction, while the residual token is stronger for Kubric object motion. One exception is static DAVIS2017, where the residual token outperforms the primary token. This result is highly variable across seeds and therefore less reliable. The performance of the upper-bound probes based on the concatenated p and r tokens remains close to the best performance achieved with a single token on most benchmarks. This indicates that most of the task-relevant information is encoded in a single token. A large gain appears

Table 13: **Additional temporal-context results.** SMM and the baseline generally benefit from additional frames on the temporally coherent Kubric and CameraBench benchmarks. On benchmarks with less temporal coherence, our method remains stable with increasing temporal context, while the frame-pair baseline can degrade. We report MSE on standardized regression targets and CameraBench mAP in %.

method	Kubric camera ↓		Kubric object ↓		DL3DV ↓	
	T=2	T=4	T=2	T=4	T=2	T=5
baseline	0.96±0.0	0.87±0.0 -0.09	0.93±0.0	0.72±0.0 -0.21	1.04±0.0	1.05±0.0 +0.01
SMM	0.26±0.03	0.16±0.01 -0.10	0.22±0.01	0.19±0.01 -0.03	0.59±0.0	0.59±0.0 ±0.0

method	Static DAVIS2017 ↓		Static YouTubeVOS ↓		CameraBench ↑	
	T=2	T=4	T=2	T=4	T=2	T=5
baseline	0.78±0.03	1.02±0.0 +0.24	0.91±0.0	0.92±0.0 +0.01	15.1±0.4	15.9±1.1 +0.8
SMM	0.87±0.07	0.87±0.05 ±0.0	0.70±0.01	0.71±0.02 +0.01	46.6±2.5	52.7±1.2 +6.1

Table 14: **Additional temporal-context results on SSv2-110k.** SMM improves monotonically as the number of input frames increases, including beyond the $T = 5$ sequence length used during training. While the frame-pair baseline also improves with increasing temporal context, even with the maximum number of $T = 7$ frames it performs worse than SMM on a single pair of frames. Accuracy is reported in %.

method	SSv2-110k					
	T=2	T=3	T=4	T=5	T=6	T=7
baseline	2.4±0.2	5.9±0.0 +3.5	8.4±0.0 +2.5	10.9±0.2 +2.5	11.7±0.0 +0.8	13.3±0.0 +1.6
SMM	16.7±0.7	18.6±0.8 +1.9	20.4±1.1 +1.8	21.6±1.2 +1.2	22.5±1.7 +0.9	23.1±1.3 +0.6

on Kubric object motion, where predicting object translation in world coordinates can benefit from both the residual object-motion signal and the primary camera-motion component.

Table 15: **Additional structured-motion results.** Full results with standard deviations showing that SMM learns specialized primary and residual motion tokens. The primary token captures the dominant motion source, while the residual token captures remaining motion. The only exception is static DAVIS2017, where the residual token outperforms the primary token. However, the high variability of this result makes it less reliable. We report MSE on standardized regression targets, mAP in % for CameraBench and accuracy in % for SSv2-110k. Among individual tokens, the best result is shown in **bold**. Representations and targets that match are highlighted in green. The last row reports an upper-bound probe using the concatenated tokens $[p,r]$.

token	Regression (MSE ↓)					Classification (mAP/acc. ↑)	
	Kubric camera	Kubric object	DL3DV	Static DAVIS2017	Static YouTubeVOS	Camera Bench	SSv2-110k
primary	**0.16**±0.01	0.21±0.03	**0.59**±0.0	0.87±0.05	**0.71**±0.02	**52.7**±1.2	**23.1**±1.3
residual	0.28±0.04	**0.19**±0.01	0.62±0.0	**0.80**±0.15	0.79±0.01	31.5±0.3	12.0±0.9
concat. $[p,r]$	0.16±0.01	0.14±0.01	0.56±0.0	0.81±0.05	0.71±0.00	54.1±1.4	22.1±1.2

B.6 Ablation

Table 16 provides the detailed component ablations summarized in Section 3.5.

Weak supervision and sequential extraction lead to specialization. On Kubric, SMM yields the expected specialization: p is better for camera motion (0.16 vs. 0.28), while r is better for object motion (0.19 vs. 0.21). Removing static-scene supervision weakens this structure, making the residual token nearly as predictive of camera motion as the primary token (0.21 vs. 0.18). Removing static-camera supervision has a stronger effect on object motion. p becomes more predictive than r (0.14 vs. 0.38), indicating that the static-camera constraint is important for preventing object motion from being absorbed into the primary token. We additionally train three capacity-matched alternatives to analyse the importance of SMM's sequential architecture. Jointly extracting p and r and matching SMM's total capacity by doubling ϕ_p's depth also weakens specialization, with p becoming more predictive for object and camera motion than r. The same

Table 16: **Detailed component ablations.** We ablate weak scene-level supervision, sequential primary/residual extraction, intermediate encoder features, unlabeled real-video training, causal prediction, the regularization strength, and the feature-prediction loss. We report MSE on standardized regression targets, CameraBench macro mAP in % and SSv2-110k accuracy in %. Blue indicates settings used by SMM. Best performance for each metric is highlighted in **bold**.

(a) **Weak supervision and sequential extraction improve token specialization.** Matched token-target pairs are highlighted in green.

token	SMM		no static scene sup.		no static camera sup.	
	Kubric camera ↓	Kubric object ↓	Kubric camera ↓	Kubric object ↓	Kubric camera ↓	Kubric object ↓
primary	**0.16**	0.21	**0.18**	0.22	**0.20**	**0.14**
residual	0.28	**0.19**	0.21	**0.20**	0.44	0.38

token	joint motion extraction		token	unstructured two-token		token	single-token	
	Kubric camera ↓	Kubric object ↓		Kubric camera ↓	Kubric object ↓		Kubric camera ↓	Kubric object ↓
primary	**0.20**	**0.16**	slot 1	**0.19**	**0.17**	slot 1	**0.19**	**0.16**
residual	0.25	0.19	slot 2	0.25	0.29			

(b) **Intermediate features improve performance.**

	Kubric↓ camera	Kubric↓ object	DL3DV↓	Camera Bench↑	SSv2-110k↑
only final feat.	0.50	0.22	0.89	31.7	13.6
multiple feat.	**0.16**	**0.19**	**0.59**	**52.7**	**23.1**

(c) **Unlabeled real data facilitates real-video generalization.**

	Kubric↓ camera	Kubric↓ object	DL3DV↓	Camera Bench↑	SSv2-110k↑
no real data	0.17	**0.11**	0.91	36.6	10.7
syn&real data	**0.16**	0.19	**0.59**	**52.7**	**23.1**

(d) **Causal prediction leads to r outperforming p on most benchmarks.**

	Kubric↓ camera	Kubric↓ object	DL3DV↓	Camera Bench↑	SSv2-110k↑
causal p	0.25	0.67	0.96	25.7	17.5
causal r	0.41	**0.17**	0.81	29.8	**26.2**
non-causal p	**0.16**	0.21	**0.59**	**52.7**	23.1
non-causal r	0.28	0.19	0.62	31.5	12.0

(e) **Medium static-camera regularization balances object motion and action prediction.**

	Kubric↓ camera	Kubric↓ object	DL3DV↓	Camera Bench↑	SSv2-110k↑
$\lambda_{\text{reg}} = 0.1$	**0.16**	0.41	**0.56**	51.6	**25.1**
$\lambda_{\text{reg}} = 0.25$	0.17	0.35	**0.56**	51.8	24.3
$\lambda_{\text{reg}} = 1.0$	0.20	**0.14**	0.59	51.4	17.7
$\lambda_{\text{reg}} = 0.5$	**0.16**	0.19	0.59	**52.7**	23.1

(f) **MSE loss improves Kubric object probing.**

	Kubric↓ camera	Kubric↓ object	DL3DV↓	Camera Bench↑	SSv2-110k↑
L_1	0.18	0.30	**0.57**	**53.6**	**24.4**
MSE	**0.16**	**0.19**	0.59	52.7	23.1

holds for a single-stage two-token model, where one slot dominates both tasks. A single-token control achieves similar absolute performance but cannot specialize by design.

Intermediate input features help across benchmarks. Using all encoder layers instead of only final-layer features improves performance across all benchmarks, including Kubric camera error from 0.50 to 0.16, DL3DV error from 0.89 to 0.59, and SSv2-110k accuracy from 13.6% to 23.1%. Despite increasing the raw feature dimensionality by 12×, the fused prediction space keeps training efficient, increasing training time by only 1.2×.

Unlabeled real data improves generalization. Adding unlabeled real videos improves over synthetic-only training on all evaluated real benchmarks: DL3DV error decreases from 0.91 to 0.59, CameraBench mAP increases from 36.6 to 52.7, and SSv2-110k accuracy increases from 10.7% to 23.1%.

A causal shifted-token variant deteriorates specialization. We modify SMM and use tokens inferred from the previous transition to predict the next. This fully causal variant leads to an r token that outperforms p across all benchmarks except Kubric camera motion probing, severely reducing token specialization.

Static-camera regularization balances object motion and action prediction. We vary λ_{reg}, the weight of the static-camera regularization term. Smaller values yield stronger SSv2-110k performance, while larger values improve Kubric object-motion probing. We use $\lambda_{\text{reg}} = 0.5$ as a compromise, preserving strong Kubric object-motion performance (0.19) and SSv2-110k accuracy (23.1%).

MSE improves Kubric object probing. Compared to $\mathcal{L}_1$, *MSE* improves Kubric object-motion error from 0.30 to 0.19 and Kubric camera-motion error from 0.18 to 0.16. In contrast, $\mathcal{L}_1$ performs slightly better on DL3DV (0.57 vs. 0.59), CameraBench (53.6 vs. 52.7 mAP), and SSv2-110k (24.4% vs. 23.1%). We use *MSE* because it significantly improves Kubric object-motion performance.

B.7 Controlled Kubric evaluations

We complement linear probing with two controlled evaluations that directly test compositionality and motion resolution. Both use the Kubric/MOVi-E rendering framework described in Section A.1, but construct dedicated controlled motion grids.

Counterfactual composition. We construct 1,000 pairs of visually distinct Kubric scenes (2,000 scenes total) with randomly sampled objects and backgrounds. Within each scene, variants share the same initial state and span six camera translations c and six object translations o, covering both directions along each of the three spatial axes. This yields $6 \times 6 = 36$ non-static camera-object combinations for each scene $A(c_i,o_j)$, where $i,j \in \{1,...6\}$. For a source scene A, we extract p_i from a camera-only variant $A(c_i,0)$ and r_j from an object-only variant $A(0,o_j)$. We apply the previously unpaired tokens to the initial feature map of a different scene B, obtaining $\hat{B}(c_i,o_j)$, and retrieve the closest ground-truth future among all 36 variants $B(c_x,o_y),x,y \in \{1,...6\}$ using feature-space MSE. We then compare its ground truth camera and object motion to the motion combination transferred using p_i and r_j. For the matched unstructured two-token model, we select the better global slot-to-motion assignment once on 200 held-out scene pairs and fix it for the remaining 800 test pairs, yielding 28,800 queries. We additionally evaluate a 4×4 planar subset that excludes depth motion.

The unstructured model nearly matches SMM on camera recovery but remains at chance on object recovery, indicating that it captures the dominant camera motion without learning an independently usable residual-motion component. Restricting the evaluation to planar motion increases exact retrieval to 88.0% for SMM compared with 24.8% for the unstructured model.

Motion magnitude resolution. We generate 1,000 static-camera Kubric scenes containing one moving foreground object and evaluate four image-plane motion directions. For each direction, we render 17 uniformly spaced magnitudes from 0 to 0.8 world units, corresponding to approximately 0–22.5 px displacement with a mean adjacent spacing of 1.41 px. Scenes are paired disjointly and evaluated in both directions. For each moving query, we directly compare its residual token r by cosine similarity against the 17 candidate residual tokens, including the static candidate, from a different scene rendered with the same motion direction. We report absolute magnitude error in pixels.

Table 17: **Controlled Kubric evaluations.** (a) Counterfactual composition evaluates retrieval after independently transferring camera and object motion components. The full setting contains 6×6 camera–object motion combinations. The planar setting contains 4×4 combinations and excludes depth motion. (b) Motion-magnitude resolution reports residual-token retrieval error and the absolute-error tolerance required to contain a given fraction of retrieved magnitudes. SMM results are reported as mean and standard deviation over three checkpoints.

(a) **Counterfactual composition.**

	3D translations			planar
	exact ↑	camera ↑	object ↑	exact ↑
SMM	53.1±1.2	90.0±3.5	58.5±2.5	88.0±1.4
unstructured two-token	14.9	89.1	16.7	24.8
chance	2.8	16.7	16.7	6.25

(b) **Motion-magnitude resolution.**

metric	error / tolerance [px]
MAE	2.51±0.09
50% within	1.55±0.03
60% within	2.73±0.03
70% within	2.90±0.03
80% within	4.15±0.05
90% within	5.64±0.11
95% within	7.12±0.14
99% within	11.00±0.09

B.8 Additional visualization

Figure 7 shows an additional visualization of the error between predicted and ground-truth feature maps.

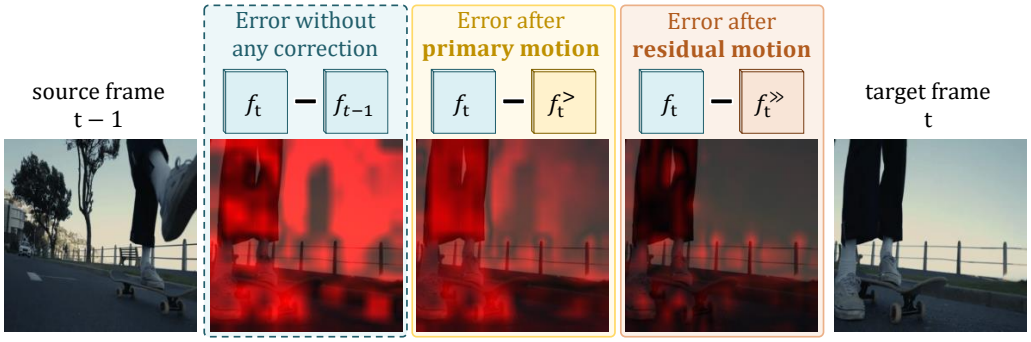


Figure 7: **Structured feature prediction.** We visualize feature-map error to the target after primary and residual compensation of a CameraBench sample. Errors are highlighted in red. Compared to the error without any compensation, the primary-motion-compensated feature map reduces the error significantly for camera-motion-induced differences, particularly in the background. The residual compensation further decreases the residual error and compensates for the error introduced by scene motion.

B.9 Further downstream tasks

We further evaluate whether frozen SMM representations transfer to established motion and video benchmarks beyond ProbeMotion. These experiments use lightweight task-specific heads or fixed correspondence pipelines and are intended to evaluate representation transfer rather than compete with specialized task-specific systems.

KITTI Odometry. We evaluate metric ego-motion decoding on KITTI Odometry (Geiger et al., 2012), training a shallow pose head on sequences 00-08, selecting the checkpoint on sequence 09, and evaluating on sequence 10. All models supporting multiple frames use a context of up to five frames, while DINOv2r and DeltaTok use consecutive frame pairs. All representations are frozen and use the same shallow pose head architecture: LayerNorm, two MLP layers with hidden dimensions 256 and 128, GELU activations and dropout 0.2, followed by separate linear heads for metric 3D translation and axis-angle rotation. This is trained using Smooth-L1 loss and standardized training targets. Predicted relative poses are integrated directly without scale or trajectory alignment, smoothing, bundle adjustment, or test-time optimization. We report the standard KITTI relative-drift metrics over 100-800 m subsequences sampled every 10 frames. Rotation drift is reported in deg/100 m.

As shown in Table 18, SMM's primary token substantially improves metric pose decoding over descriptors taken directly from DINOv2r and matches resolution-matched DeltaTok-224 on both metrics. The substantially larger, strongly supervised geometry representations remain stronger, with the best results reaching $9.02 \pm 0.56\%$ translation drift and 2.56 ± 0.86 deg/100 m rotation drift.

KITTI optical flow. We evaluate on the KITTI 2015 (Menze & Geiger, 2015) public training set using a fixed 140/30/30 train/validation/test split. Frozen final-layer DINOv2r patch features are locally matched at radii $r \in \{4,12\}$ (56 and 168 pixels) using the top-3 cosine matches with temperature 0.07. A learned binary gate mixes the two coarse-flow estimates using matching confidence, entropy, forward-backward consistency and validity, and disagreement between the two estimates. We initialize the gate with $p(r=12)=0.95$, biasing the early training toward the wider search radius. Conditioning variants modulate the gate through FiLM (Perez et al., 2018) using either a DINOv2r `CLS`/AVG pair or $[p,r]$. A learned convex upsampler (Teed & Deng, 2020) produces full-resolution flow. The gate and upsampler are pretrained for 15 epochs on FlyingChairs2 (Ilg et al., 2018) and fine-tuned for 20 epochs on KITTI. All DINOv2r and SMM representations remain frozen.

Conditioning the same flow head on SMM improves the unconditioned gate on all four metrics, reducing Fl-fg by 3.34 points, Fl-all by 1.05 points, EPE-fg by 0.67 pixels, and EPE-all by 0.76 pixels (see Table 20). In contrast, conditioning on the dimension-matched DINOv2r descriptors degrades performance.

Video object segmentation. We evaluate training-free semi-supervised mask propagation on DAVIS2017 (Pont-Tuset et al., 2017) and YouTube-VOS 2019 (Xu et al., 2018b;a) by adapting DINO's nearest-neighbor label-propagation protocol (Caron et al., 2021). We use L_2-normalized features, top-10 correspondences over joint space-time memory, radius 12, temperature 0.1, and seven recent

Table 18: **KITTI Odometry.** Frozen-representation pose decoding on held-out sequence 10 using the same shallow pose head architecture. DeltaTok-224 is resolution matched to SMM. For the strongly supervised geometry models, we report the best result for each metric across the descriptors evaluated in Table 19.

representation	translation drift ↓ [%]	rotation drift ↓ [deg/100 m]
strongly supervised		
VGGT best	9.09±0.35	2.56±0.86
DA3 best	13.15±2.47	4.04±0.93
Pi3X best	9.02±0.56	3.11±0.87
baselines		
CLS diff.	54.95±1.14	22.38±0.77
AVG diff.	55.80±0.16	21.97±0.04
CLS concat.	51.06±10.46	23.75±3.12
AVG concat.	49.49±2.13	23.36±2.44
DeltaTok-224	22.79±2.16	7.79±2.49
SMM p	**20.83**±1.23	**7.55**±0.69
SMM r	30.04±4.15	12.91±2.92
SMM $[p,r]$	22.33±5.27	9.42±3.16

Table 19: **KITTI Odometry: supervised geometry representations.** All representations are frozen and evaluated with the same pose-decoding protocol and shallow pose head architecture as Table 18. For each model, the best result for each metric is highlighted in bold.

descriptor	translation drift ↓ [%]	rotation drift ↓ [deg/100 m]
VGGT		
camera token diff.	14.85±1.88	**2.56±0.86**
camera token concat.	**9.09±0.35**	3.14±0.52
AVG feat. diff.	12.66±1.77	3.23±0.60
AVG feat. concat.	12.59±1.28	3.98±0.40
DA3		
camera token diff.	15.17±2.96	**4.04±0.93**
camera token concat.	17.21±2.66	7.00±1.10
AVG feat. diff.	14.82±0.91	5.12±1.13
AVG feat. concat.	**13.15±2.47**	6.44±1.08
Pi3X		
AVG registers diff.	15.41±5.18	4.02±1.43
AVG registers concat.	25.38±10.14	12.35±4.49
AVG feat. diff.	**9.02±0.56**	**3.11±0.87**
AVG feat. concat.	11.80±1.38	5.49±1.20

predicted-mask frames in addition to permanent annotated anchors. DINOv2r transfers masks directly between observed features $\mathcal{M}_t \to f_{t+1}$, where $\mathcal{M}_t$ denotes the memory. For SMM, the predicted final feature map acts as an intermediate correspondence space, bridging the gap between the two timesteps: $\mathcal{M}_t \to f^{\gg}_{t+1} \to f_{t+1}$. For the 224-window variants at 448-pixel resolution, DINOv2r and SMM are extracted using overlapping 224×224 windows with stride 112. Fusion averages the output-resolution object probabilities from independent DINOv2r and SMM propagation branches with equal weight. Fused predictions are not fed back into either branch's memory. On DAVIS2017, all objects are initialized from the first annotated frame. On YouTube-VOS, each object becomes active at its first annotation, which is retained as a permanent anchor.

At SMM's native resolution, the final bridge improves DAVIS2017 $\mathcal{J}\&\mathcal{F}$ from 28.94 to 32.54 and YouTube-VOS $\mathcal{G}$ from 33.09 to 34.65. At higher resolution, standalone windowed SMM is weaker than windowed DINOv2r, but provides complementary transition information: replacing the windowed DINOv2r fusion branch with SMM increases DAVIS2017 $\mathcal{J}\&\mathcal{F}$ from 62.95 to 66.08 and YouTube-VOS $\mathcal{G}$ from 60.10 to 61.35.

C LIMITATIONS

While a large portion of our training is fully self-supervised, the decomposition part is also anchored by weak scene-level supervision on synthetic data (see Section 3.5). Replacing this with fully self-supervised constraints would improve scalability. Longer temporal context is most useful for temporally coherent motion and is not uniformly beneficial on unconstrained real-world videos (see Section B.4). Moreover, repeated token extrapola-

Table 20: **KITTI 2015 optical flow.** Results on the fixed 30-pair held-out split. We report foreground and overall KITTI outlier rate (Fl) and endpoint error (EPE).

Variant	Fl-fg [%] ↓	Fl-all [%] ↓	EPE-fg [px] ↓	EPE-all [px] ↓
Zero flow	90.05±0.00	87.71±0.00	57.29±0.00	43.18±0.00
Fixed radius 4	66.12±1.70	67.57±0.70	25.98±0.14	24.96±0.03
Fixed radius 12	60.49±0.55	68.96±0.25	11.94±0.07	20.67±0.06
Adaptive $r4/r12$ gate	55.48±1.11	66.03±0.21	11.59±0.21	21.00±0.15
+ DINOv2r `CLS` pair	66.94±2.52	68.71±0.81	21.44±8.03	23.64±2.52
+ DINOv2r AVG pair	64.98±3.44	68.24±0.68	21.12±8.13	23.50±2.49
+ SMM $[p,r]$	**52.14±0.64**	**64.98±0.08**	**10.92±0.08**	**20.24±0.04**

Table 21: **Training-free video object segmentation.** We evaluate the fixed mask-propagation protocol on DAVIS2017 and YouTube-VOS 2019. SMM results are reported as mean and standard deviation over three checkpoints, DINOv2r uses a single deterministic frozen representation.

(a) **DAVIS2017.**

method	setting	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
DINOv2r	224	33.44	24.44	28.94
SMM	224	38.15±1.10	26.92±0.48	32.54±0.67
DINOv2r	448	58.84	61.76	60.30
DINOv2r	448, 224-window	60.95	64.47	62.71
SMM	448, 224-window	57.55±0.97	61.31±1.20	59.43±1.07
fusion	DINOv2r 448 + DINOv2r window	61.28	64.63	62.95
fusion	DINOv2r 448 + SMM window	**63.91±0.38**	**68.25±0.62**	**66.08±0.50**

(b) **YouTube-VOS 2019.**

method	setting	$\mathcal{J}_s$	$\mathcal{F}_s$	$\mathcal{J}_u$	$\mathcal{F}_u$	$\mathcal{G}$
DINOv2r	224	43.87	26.75	34.91	26.82	33.09
SMM	224	46.91±0.64	29.58±0.80	35.42±0.39	26.69±0.43	34.65±0.54
DINOv2r	448	61.05	58.43	53.64	60.77	58.47
DINOv2r	448, 224-window	61.68	60.17	54.74	62.57	59.79
SMM	448, 224-window	60.13±0.57	58.38±1.08	51.16±0.30	56.84±0.77	56.62±0.61
fusion	DINOv2r 448 + DINOv2r window	62.21	60.56	54.86	62.76	60.10
fusion	DINOv2r 448 + SMM window	**63.62±0.28**	**63.10±0.48**	**55.32±0.14**	**63.38±0.18**	**61.35±0.17**

tion and cross-video transfer drift over longer horizons (see Section 3.6), and residual-token motion-magnitude retrieval has finite resolution (2.51 ± 0.09 px MAE, see Section B.7). The additional odometry, optical-flow and VOS evaluations depend on deliberately lightweight task-specific heads or correspondence pipelines, so their absolute performance is sensitive to these design choices and is intended to evaluate representation transfer rather than competition with specialized systems (see Section B.9). Finally, ProbeMotion relies on estimated camera motion for constructing static DAVIS2017 and YouTubeVOS (see Section A.2), which can introduce filtering noise, and linear probes only measure linearly decodable information (see Section 3.2).

D Information about used assets

Table 22 lists more detailed information about the pretrained models and datasets used.

E Hardware resources

The experiments were conducted on a local cluster using a single node with 4 NVIDIA H100 GPUs and 128 CPU cores. One SMM training run takes approximately 11 hours on such a node and uses up to 75GB of RAM and 25GB of VRAM per GPU. The evaluation on ProbeMotion takes approximately 25 minutes.

The reported main results use three seeds, for a total of approximately $11\text{h} \times 4 \times 3 = 132$ GPU-hours, where 4 is the number of GPUs per node. The 12 ablation runs require approximately 528 GPU-hours.

Table 22: **Existing assets.** Summary of existing datasets and pretrained models used in this paper.

Asset	Use in this paper	Version / subset	License / terms	Citation
Pretrained models				
DINOv2-reg-B/14	Main frozen image encoder	public checkpoint	Apache License 2.0	(Darcet et al., 2024)
MAE-B/16	Frozen encoder for backbone ablation	public checkpoint	Apache License 2.0	(He et al., 2022)
SigLIP-B/16	Frozen encoder for backbone ablation	public checkpoint	Apache License 2.0	(Zhai et al., 2023)
DINOv3-B/16	Frozen encoder for backbone ablation	public checkpoint	DINOv3 License	(Siméoni et al., 2025)
DeltaTok	Self-supervised baseline	released pretrained checkpoint	Apache License 2.0	(Kerssies et al., 2026)
VGGT-1B	Supervised 3D representation baseline and camera-motion filtering for static-camera benchmarks	public checkpoint	CC BY-NC 4.0	(Wang et al., 2025b)
Depth Anything 3 (GIANT-1.1)	Supervised 3D representation baseline	public checkpoint	CC BY-NC 4.0	(Lin et al., 2025a)
Pi3X	Supervised 3D representation baseline	public checkpoint	CC BY-NC 4.0	(Wang et al., 2025d)
Datasets				
Kubric / MOVi-E	training and evaluation	Custom generated clips based on MOVi-E setup	Apache License 2.0	(Greff et al., 2022)
Something-Something v2	training and evaluation	Filtered train/validation split	license	(Goyal et al., 2017)
DL3DV	training and evaluation	`1K–7K` subsets	DL3DV-10K Terms of use	(Ling et al., 2024)
CameraBench	evaluation	Official train/test splits, 15 binary motion tasks	CC-BY-4.0	(Lin et al., 2025b)
DAVIS2017	ProbeMotion and VOS evaluation	filtered ProbeMotion split, standard DAVIS2017 semi-supervised evaluation	BSD 3-Clause License	(Pont-Tuset et al., 2017)
YouTubeVOS	ProbeMotion and VOS evaluation	filtered ProbeMotion split, VOS2019 dense validation	CC BY 4.0 (annotations) / non-commercial research purpose only (data)	(Xu et al., 2018b;a)
KITTI	odometry and optical-flow evaluation	odometry sequences 00–10, KITTI 2015 public training set	CC BY-NC-SA 3.0	(Geiger et al., 2012; Menze & Geiger, 2015)
FlyingChairs2	optical-flow-head pretraining	official release	research purposes only	(Ilg et al., 2018; Dosovitskiy et al., 2015)