Towards Data-Efficient Video Pre-training with Frozen Image Foundation Models

Supplementary Material

The offline evaluation protocol, including readout architectures and training procedures, follows [4, 25]. In this supplementary material, we explain the streaming evaluation, which reflects the real-world scenario of receiving video frames one by one and processing them as soon as they arrive. In this setting, the readout operates on a single frame's tokens at each time step, and all temporal context must reside in the recurrent state of the temporal module.

6. Streaming Tasks

Table 4 provides an overview of each task, including the training dataset, loss function, evaluation metric, and readout head parameters. The readout heads are based on the cross-attention architecture from [25], adapted to operate on single-frame tokens (N tokens per frame) instead of the full spatio-temporal sequence ($T \times N$ tokens). The encoder is always frozen; only the temporal module and readout head receive gradients. In the case of pre-trained RVM models, only the readout is trained.

6.1. Action Recognition (SSv2)

The offline readout attends to all $T \times N$ spatio-temporal tokens with learned temporal positional embeddings. In the streaming setting, the readout instead attends to the current frame's N tokens only (12 heads, d=768), without temporal positional embeddings, since temporal context is captured entirely by the recurrent state. A single learned query is projected to 174 classes. The final prediction $\hat{y}_T$ from the last frame is used for evaluation. The loss is standard cross-entropy.

6.2. Object Tracking (Waymo)

Following the offline protocol [4, 25], the initial bounding box $[c_x, c_y, w, h]$ is encoded via 16 Fourier frequencies and processed through an MLP (hidden dim 512) to produce a single query token. The difference is in the readout: instead of attending to all $T \times N$ tokens and predicting all frames at once, the streaming readout processes one frame at a time. At each frame t, the query attends to the current frame's N tokens via cross-attention (d=1024, 4 heads), and the updated query is projected through an MLP to 4 bounding box coordinates. The refined query is then carried over to frame $t{+}1$, acting as an evolving tracker state alongside the temporal module's recurrent state. The loss combines GIoU (weight 2.0) and L1 (weight 5.0) on the predicted coordinates.

6.3. Point Tracking (Perception Test)

Models are trained on the synthetic Kubric MOVi-E dataset [9] and evaluated on real videos from Perception Test, constituting a synthetic-to-real transfer setting. Each sample uses 64 query points, initialized with their ground-truth (x, y) position at the first frame. Each query point is encoded via 16 Fourier frequencies and processed through an MLP (2×512 hidden units), then linearly projected to d=1024.

The offline readout replicates each query 8 times with learnable temporal embeddings, and each of the 8 queries predicts 2 consecutive frames while attending to the full $T \times N$ spatio-temporal tokens. In the streaming setting, the readout instead uses a single query per point, without temporal embeddings: at each frame, the query attends to the current frame's N tokens via cross-attention (d=1024, 8 heads) and directly predicts position (x, y), visibility, and uncertainty for that frame. The loss combines Huber loss (δ=0.05, weight 100.0) on positions (visible points only), binary cross-entropy on visibility (weight 0.1), and binary cross-entropy on uncertainty (weight 0.1).

6.4. Depth Estimation (ScanNet)

The offline readout uses spatio-temporal $2 \times 8 \times 8$ patches as queries over all frames, while the streaming readout uses $\frac{H}{8} \times \frac{W}{8}$ spatial patches per frame. Since we use a fixed input resolution of 224×224, this gives $28 \times 28 = 784$ learned queries per frame. Each query attends to the current frame's N tokens via cross-attention (d=1024, 16 heads) and predicts $8 \times 8 = 64$ depth values for its patch, which are rearranged to produce a full-resolution 224×224 depth map.

6.5. Camera Pose Estimation (NuScenes)

At each frame t, the readout aggregates the N spatial tokens via mean pooling and passes the pooled vector through an MLP (LayerNorm, Linear $d{\to}512$, GELU, Linear $512{\to}9$) to predict a 9-dimensional pose delta: 3 values for translation (dx, dy, dz) and a 6-dimensional rotation representation [24], which avoids the discontinuities of quaternions. Predictions are frame-to-frame deltas (the pose change from frame $t{-}1$ to frame t).

The loss uses learnable translation/rotation balancing following Kendall & Cipolla [12]:

$$\mathcal{L} = \mathcal{L}_{\text{trans}}\, e^{-s_t} + s_t + \mathcal{L}_{\text{rot}}\, e^{-s_r} + s_r, \qquad (20)$$

where $\mathcal{L}_{\text{trans}}$ and $\mathcal{L}_{\text{rot}}$ are the L1 losses on the translation

Table 4. **Streaming task overview.** Training and evaluation setup for each downstream task.

Task	Dataset	Train Loss	Eval Metric	Readout (dim, heads)
Action Recognition	SSv2 [8]	Cross-entropy	Top-1 Acc. (%) ↑	(768, 12)
Object Tracking	Waymo Open [20]	GIoU + L1	mIoU ↑	(1024, 4)
Point Tracking	Kubric MOVi-E [9] (train), PT [16] (eval)	Huber + BCE	Avg. Jaccard ↑	(1024, 8)
Depth Estimation	ScanNet [6]	Log-space L2	AbsRel ↓	(1024, 16)
Pose Estimation	NuScenes [3]	L1 (learned balancing)	RPE_{tr} (mm) ↓	(1024, MLP)

Table 5. **Training regimes.** All regimes use AdamW, cosine schedule to $\eta_{\min} = 10^{-7}$, and bf16 mixed precision.

Regime	Trainable	Train steps	Warmup
Frozen	readout only	40,000	1,000
RNN fine-tuning	RNN + readout	100,000	5,000
Streaming	RNN + readout	20,000	1,000

Table 6. **Peak learning rates (offline).** Frozen and RNN fine-tuning regimes. Dv3: DINOv3.

Regime	Model	SSv2	Waymo	Kubric
Frozen	RVM (frozen)	3e-4	3e-4	1e-4
	Dv3 (no RNN)	1e-4	1e-4	5e-5
RNN fine-tune	Dv3 + RVM_{RNN}	1e-4	1e-4	5e-5
	Dv3 + Mamba	1e-4	1e-4	5e-5
	Dv3 + MMix	1e-4	1e-4	5e-5
	Dv3 + GMMix	1e-4	1e-4	5e-5

and 6D rotation components, respectively, and s_t, s_r are learnable log-variance parameters that automatically balance the two terms.

7. Training Settings

All tasks share the same training configuration unless stated otherwise. We use AdamW [14] with $(\beta_1, \beta_2) = (0.9, 0.999)$, weight decay 10^{-4}, a cosine learning rate schedule decaying to $\eta_{\min} = 10^{-7}$, and linear warmup. Training uses mixed precision (bf16). Our protocol is based on 4DS [4], which used 40K steps for frozen training (only the readout trainable) and 80K steps for fine-tuning. We train for 40K steps in the frozen regime and 100K steps in the RNN fine-tuning regime. Streaming experiments use a dedicated protocol of 20K training steps. The three regimes differ in which components are trained and in the warmup length, as summarized in Tab. 5.

The only setting that varies across tasks and models is the peak learning rate, reported in Tabs. 6 and 7.

Table 7. **Peak learning rates (streaming).** All models trained for 20K steps with 1K warmup. Dv3: DINOv3.

Model	SSv2	Waymo	Kubric	ScanNet	NuScenes
RVM (frozen)	1e-4	1e-4	3e-4	1e-4	1e-3
Dv3 + RVM_{RNN}	1e-4	5e-5	1e-4	5e-5	3e-4
Dv3 + Mamba	1e-4	5e-5	1e-4	5e-5	3e-4
Dv3 + MMix	1e-4	5e-5	1e-4	5e-5	3e-4
Dv3 + GMMix	1e-4	5e-5	1e-4	5e-5	3e-4

8. Evaluation Metrics

Top-1 accuracy (SSv2). Standard classification accuracy on the validation set.

mIoU (Waymo). Mean Intersection over Union between predicted and ground-truth bounding boxes, averaged over all objects and frames.

Average Jaccard (PT). Following the Perception Test benchmark [16], AJ is defined as the average of Jaccard values at position thresholds of 1, 2, 4, 8, and 16 pixels, where a point is considered correctly tracked if its predicted position is within the threshold and its visibility is correctly predicted.

AbsRel (ScanNet). Absolute relative error:

$$\text{AbsRel} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \frac{|d_p - \hat{d}_p|}{d_p}, \tag{21}$$

where d_p and $\hat{d}_p$ are the ground-truth and predicted depth at pixel p, and $\mathcal{P}$ is the set of valid pixels.

RPE_{tr} and RPE_{rot} (NuScenes). Translational (mm) and rotational (degrees) components of the relative pose error between consecutive frames.

Normalized average. Each score is divided by the column-best across all models in the table, and the ratios are averaged. For metrics where lower is better (AbsRel, RPE_{tr}), we use (column-best / score) instead of (score / column-best).