

# TimeBridge: Self-Supervised Video Representation Learning via Start-End Joint Embedding and In-Between Frame Prediction

## Supplementary Material

### 1. Technical Appendix

#### 1.1. Implementaion details

**Pretraining.** Our methods are pretrained on Kinetics-400 datasets, the detailed training recipe is shown in Table 1. The code is based on the iBOT code online <https://github.com/bytedance/ibot>, we will publish our code and models. The hyperparameters are followed by iBOT [2]. We fixed the effective batch size, Effective batch size = Batch size per GPU  $\times$  Number of GPUS.

Table 1. Hyperparameters of our proposed methods TimeBridge. The pretraining is on Kinetics400 dataset.

Configuration	TimeBridge
Optimizer	AdamW
Arch	ViTS
Effective batch size	512
Learning rate	0.005
Learning rate schedule	cosine annealing
Frame gap	16
Patch size	16/8
Warmup epochs	10
Epochs	400
Global crops scale	[0.25, 1.0]
Local crops scale	[0.05, 0.25]
Teacher patch temp	0.07
Teacher temp	0.07
Patch out dimension	8192

**Video dense prediction.** The evaluation on video dense prediction follows the prior work, CropMAE. We evaluate the output patch tokens using the DAVIS-2017 video object segmentation benchmark. Following the experimental protocol of DINO, we do not train or finetune any network, instead relying on nearest neighbor matching between consecutive frames. For pose tracking on the JHMDB benchmark. Given 15 ground truth human pose keypoints in the first frame, we propagate them to the remaining frames. For VIP dataset, we also propagate in a same way as the JHMDB, but propagate the semantic segmentation maps of human parts.

**Video action classification.** We also evaluate our pre-trained model using linear probing on video action classification datasets. For each video clip, a single frame is randomly sampled. The training details are summarized in Table 3.

Table 2. Hyperparameters for Video dense prediction tasks.

Configuration	DAVIS 2017	VIP	HMDB51
Top-k	7	10	7
Queue Length	20	20	20
Neighborhood Size	20	20	20

Table 3. Hyperparameters of linear probing for video action classification.

Configuration	TimeBridge
Optimizer	AdamW
Arch	ViTS
Effective batch size	128
Learning rate	0.005
Frame	1
Patch size	16/8
Epochs	20
Learning rate schedule	cosine annealing
Augmentation	crop, flipping

#### 1.2. Computation resources

The pretraining process for Vision Transformers (ViT-S/8) was conducted using 8 compute nodes, each equipped with 4 NVIDIA A100 GPUs. One complete run for pretraining ViT-S/8 on Kinetics-400 for 400 epochs takes around 36 hours. For evaluation, we use a single NVIDIA A100 GPU for video dense prediction, which takes 4 hours to complete the evaluation across all three datasets. We use one NVIDIA H100 GPU for linear probing on video action classification datasets. HMDB51 and UCF101 are smaller than Kinetics-400, taking only about 2 hours to complete training and evaluation, while Kinetics400 requires around 16 hours. Overall we spent 1200 GPU-h per run for the research conducted for this paper.

With the default two-decoder setting, our method uses approximately 50% fewer parameters (3.18M vs. 6.32M) and 50% fewer FLOPs (0.64G vs. 1.24G) compared with the attention-based decoders in SiamMAE and CropMAE.

#### 1.3. Augmentations

To enhance the robustness and generalization of the model, a diverse set of data augmentations were applied. These include spatial and color-based transformations such as random resized cropping to a fixed size of  $224 \times 224$ , color jittering with adjustable brightness, contrast, saturation, and hue, and Gaussian blur with a variable radius. Additional

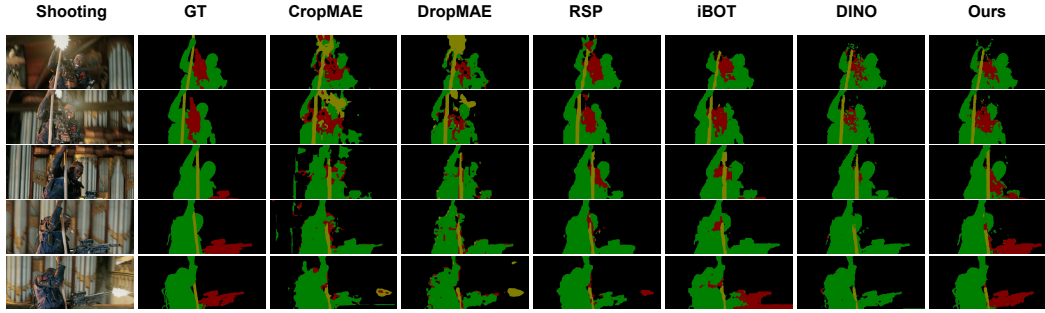


Figure 1. Comparison of segmentation across methods on **shooting** clip. From top to bottom, the frames correspond to frame index 1, 25% of the total frames, 50%, 75%, and the last frame. This ordering is consistent across the comparison figures that follow.

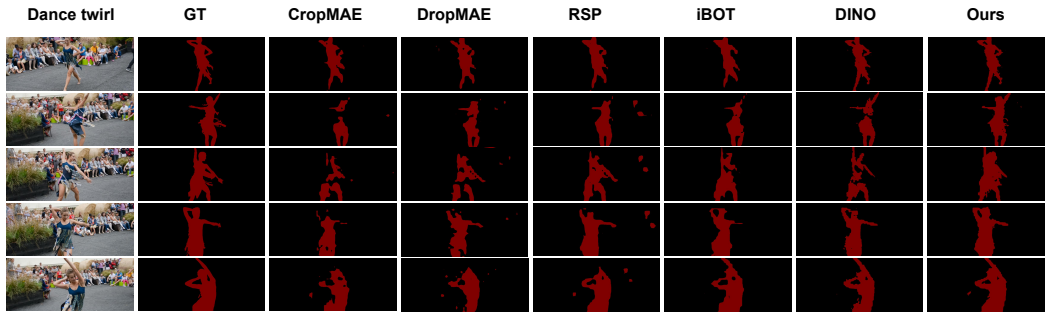


Figure 2. Comparison of segmentation across methods on **dance twirl** clip.

Table 4. Considered augmentations.

Augmentation $T$	Parameters	Mag. range	Prob.
Random resized crop	size	(224,224)	1
Color jittering	B, C, S, H	$[-0.4, 0.4]$ $[-0.1, 0.1]$	0.8
Gaussian blur	radius $\sigma$	$[0.1, 2]$	1
Grayscale	—	—	0.2
Flip	—	—	0.5
Solarization	—	—	0.2

stochastic augmentations include grayscale conversion (20% probability), horizontal flipping (50% probability), and solarization (20% probability). Each transformation is applied with a specific magnitude range and probability as detailed in Table 4.

## 2. More qualitative results

In this section, we present additional qualitative results, including comparisons with the top-performing models from competitors. We select some representative video clips for comparison across the methods. Segmentation masks from representative videos, for both our method and the competing methods, are included in the folder **qualitative\_comparison**.

We use the provided checkpoints from different methods to evaluate the DAVIS 2017 dataset following the official evaluation protocol [1]. The configurations of these check-

points are summarized in Table 5. All reproduced qualitative results are based exclusively on these checkpoints, which were obtained from the original publications.

Table 5. Configurations of Methods on DAVIS 2017 Using Provided Checkpoints and ours

Method	Backbone	Pretraining Dataset	Epochs
CropMAE	ViT-S/16	Kinetics-400	400
DINO	ViT-S/8	ImageNet	800
DropMAE	ViT-B/16	Kinetics-400	1600
iBOT	ViT-S/16	ImageNet	800
RSP	ViT-S/16	Kinetics-400	400
Ours	ViT-S/8	Kinetics-400	400

**Shooting.** Shooting is one of the video clips from the DAVIS 2017 dataset. Figure 1 illustrates the segmentation results across video frames for different methods. Most competitors fail to accurately segment the guns in the later frames. In contrast, our method successfully recovers the segmentation in the final 50% of the frames, accurately segmenting not only the human figures but also the guns.

Unlike DINO, which struggles to distinguish the gun from the person, our model provides more precise object segmentation, correctly identifying and separating the guns from the people.

**Dance twirl** The segmentation masks generated for the

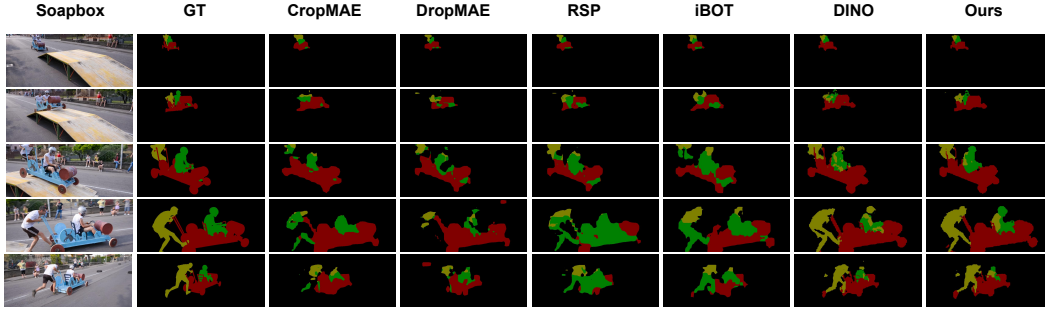


Figure 3. Comparison of segmentation across methods on **soap box** clip.

Dance Twirl video clip using different methods are presented in Figure 2. In the final frames of the sequence, DINO demonstrates relatively strong performance compared to other approaches. However, our method surpasses all existing techniques, including DINO, particularly in preserving fine details such as the body contours, the skirt, and the arms.

**Soap box.** For the video clip soap box, all MAE-based methods perform poorly. Even in the early frames, the object boundaries in the segmentation masks are unclear. In the later frames, the performance degrades further—objects are often misclassified. For example, RSP incorrectly segments the car, while both RSP and CropMAE misclassify the human. DropMAE fails to identify the human body altogether.

In comparison, our method and DINO perform similarly, although some artifacts remain—for instance, inaccuracies around the feet in the final frame. Overall, our method produces more accurate object segmentation with clearer boundaries.

**Lab coat.** We also observe relatively poor segmentation performance on certain sequences—lab-coat, for example. In the later frames of our results, the faces of the middle and right women are frequently misclassified as belonging to the left one. This confusion indicates challenges in maintaining identity consistency over time. Furthermore, most methods struggle with accurate object segmentation in this sequence. Notably, the cell phone is consistently missegmented or entirely missed; none of the evaluated approaches successfully identify or segment it correctly. In general, all methods—including ours—exhibit varying degrees of failure in preserving distinct segmentation masks for the three individuals, suggesting a broader limitation in handling complex multi-object scenes with fine-grained details.

**Gold fish.** All methods demonstrate relatively strong performance on the gold-fish sequence, indicating their general capability to segment complex objects in video. However, DropMAE underperforms on certain objects within this clip, suggesting limitations in its ability to handle fine-grained segmentation in multi-object scenes.

**Motocross jump** Although our model outperforms others

in most clips, there are cases, such as the motocross jump, where it does not provide the most benefit. Figure 6 presents a comparison of different methods. In the early frames, our results are comparable to iBOT’s, and even better in terms of fine details, for example, the motorcycle’s wheels are more accurately segmented by our model. However, in the later frames, our model misclassifies parts of the rider as part of the motorcycle, resulting in worse performance compared to iBOT.

**Car roundabout.** Another example where our method does not perform well is the ‘Car Roundabout’ clip, shown in Figure 7. Although our model correctly identifies the contour of the car, it also segments the curbstones, resulting in false positives. This issue is also observed in DINO and RSP. In contrast, iBOT not only segments the car accurately but also avoids false positives, making it the best-performing method for the ‘Car Roundabout’ clip.

### 3. Comparison to iBOT pretrained on Kinetics-400

Originally, iBOT was not designed for pretraining on sequential data. In this work, we propose an adaptation that enables the use of iBOT loss for sequence-based pretraining, producing meaningful representations for downstream video tasks. Specifically, instead of applying manual augmentations to a single image to create two global views—as done in the original iBOT—we sample two different frames from a video sequence to serve as the global views. Each frame is then independently augmented using standard transformations. This setup constitutes a special case of our method, where we discard the reconstruction loss entirely and rely solely on the iBOT loss. A comparison between the ImageNet-1K pretrained baseline, Kinetics-400 pretrained iBOT, and our approach is presented in Table 6.

With our adaptation, iBOT is able to learn representations that perform well on dense video tasks. However, it still underperforms compared to our full method. Using the same patch size and frame gap, and after tuning the learning rates, our approach achieves a 1.8 gain in percentage points on the

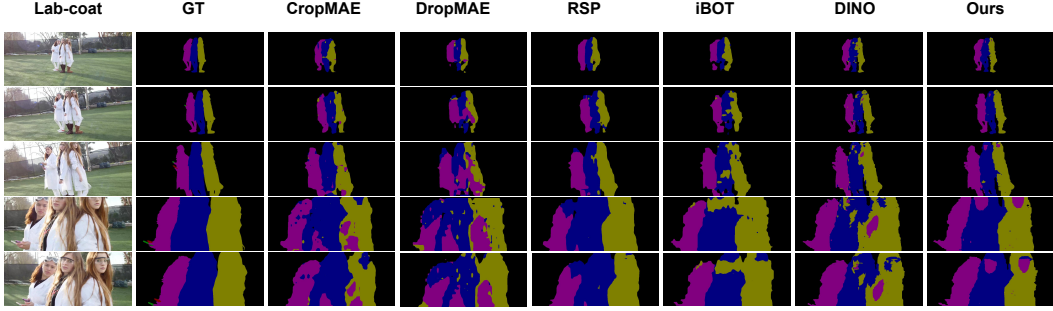


Figure 4. Comparison of segmentation across methods on **lab coat** clip.

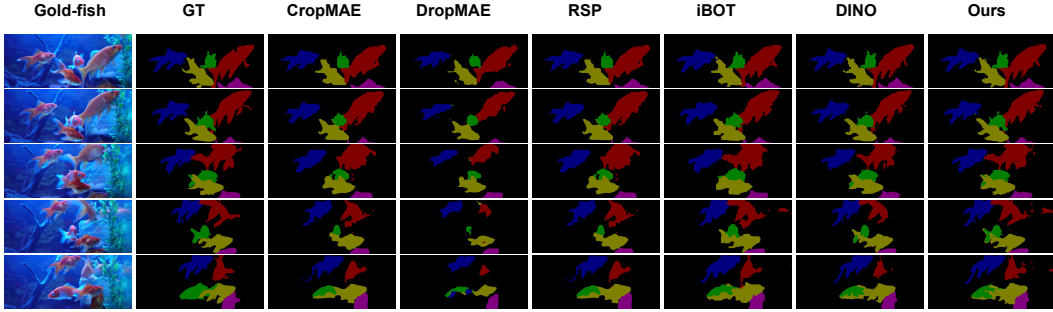


Figure 5. Comparison of segmentation across methods on **gold fish** clip.

Table 6. **Comparison with iBOT pretraining on different datasets.** ‡ uses published checkpoints; † reproduced by us.

Method	Backbone	Dataset	Epoch	$\mathcal{J} \& \mathcal{F}_m$	$\mathcal{J}_m$	$\mathcal{F}_m$
iBOT ‡	ViT-S/16	IN1K	800	62.8	61.2	64.5
iBOT †	ViT-S/8	K400	400	71.7	68.8	74.6
Ours	ViT-S/16	K400	400	65.8	63.7	67.8
Ours	ViT-S/8	K400	400	<b>73.5</b>	<b>70.6</b>	<b>76.5</b>

$\mathcal{J} \& \mathcal{F}_m$  metric over the adapted iBOT by incorporating our auxiliary reconstruction objective.

#### 4. mIoU across video frames

We plot the mIoU vs single video frames for the video clips and methods considered. As the segmentation method is initialized with the ground truth mask, the mIoU is 1 for all considered methods at the first frame (indexed by 0). In this section, we present results for representative video clips also shown qualitatively in Section 2, providing insight into how different methods perform over time. More results are attached in the folder **plots\_mIoU**.

**Shooting.** These results (Figure 8) are consistent with the qualitative observations in Section 2. Notably, our method continues to improve in the later frames, even outperforming its performance in the initial frames. In contrast, the other methods do not show similar improvements over time, highlighting the superior temporal consistency and refinement

capability of our approach.

**Dance twirl.** Our method consistently outperforms all baseline approaches, with especially strong performance observed in the final frames.

**Soapbox.** In most frames, our method achieves a higher mIoU than all competitors. Although performance drops below DINO in the final frames, our approach still achieves the highest average mIoU overall. We also observe a noticeable decline in RSP’s performance during the last 40 frames.

**Lab coat.** The lab coat is one of videos that our method that not perform really The Lab Coat video is one of the cases where our method does not perform particularly well, similar to other approaches. While our method performs slightly better than the competitors, the overall results are less favorable compared to other sequences.

**Motocross jump.** From Figure 12, we can conclude that, in the first half of the frames, our method outperforms all others in terms of mIoU. However, in the later frames, the improvement of our method is less significant compared to iBOT and partly RSP. Although iBOT experiences a performance drop in the middle of the video, it recovers and achieves the best performance in the second half compared to ours.

**Car roundabout.** This example shows one of the rare cases our methods perform worse than our competitors. As shown in Figure 13, all methods perform well on the ‘Car Roundabout’ clip, especially iBOT and DropMAE, which do

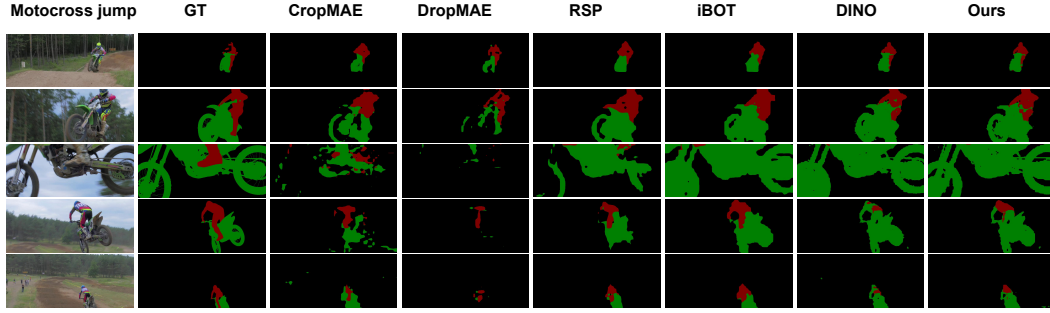


Figure 6. Comparison of segmentation across methods on **motocross jump** clip.

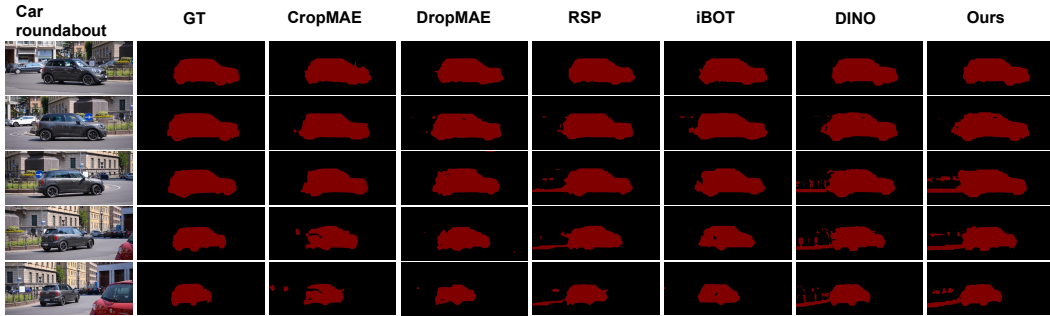


Figure 7. Comparison of segmentation across methods on **car roundabout** clip.



Figure 8. Comparison of segmentation mIoU across different methods on the **shooting** clip.

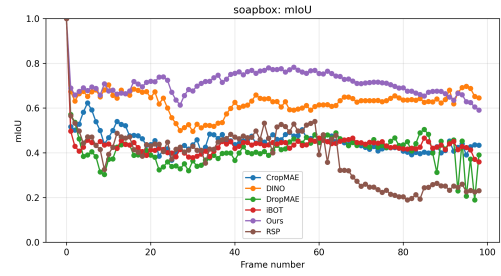


Figure 10. Comparison of segmentation mIoU across different methods on the **soap box** clip.

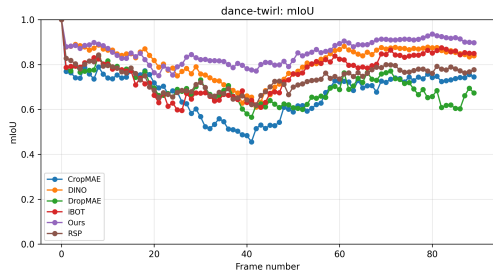


Figure 9. Comparison of segmentation mIoU across different methods on the **dance-twirl** clip.

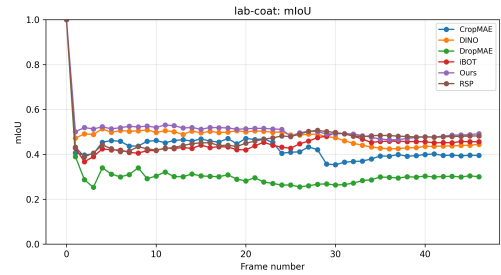


Figure 11. Comparison of segmentation mIoU across different methods on the **lab-coat** clip.

not show a significant performance drop in the later frames. In contrast, our method and DINO, while leading in the first

quarter of the sequence, perform worse in the later frames. This is because the presence of false positives, as discussed



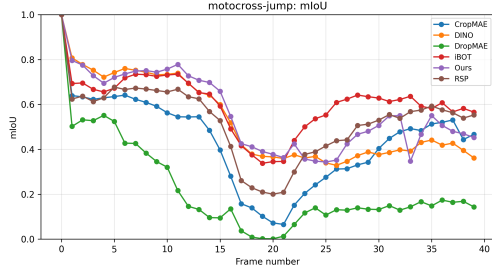


Figure 12. Comparison of segmentation mIoU across different methods on the **motocross jump** clip.

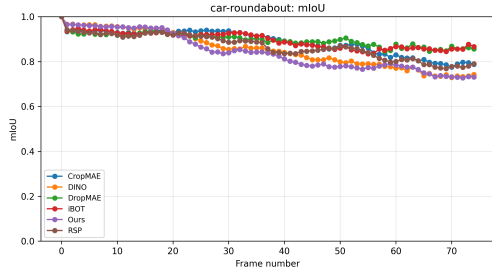


Figure 13. Comparison of segmentation mIoU across different methods on the **car-roundabout** clip.

in Section 2, negatively impacts the final results.

## 5. Results per sequence

The results on the DAVIS 2017 dataset, presented in Table 7 and Table 8, show that our method outperforms the competitors in terms of both  $\mathcal{J}_m$  and  $\mathcal{F}_m$  for nearly all sequences. The underscore (e.g., in bike-packing\_1) indicates different object instances or segments within a single video sequence.

## References

- [1] 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. 2
- [2] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT: Image BERT pre-training with online tokenizer. *International Conference on Learning Representations*, 2022. 1

Table 7.  $\mathcal{J}_m$  Scores Comparison Across Methods on DAVIS 2017

Sequence	CropMAE	DropMAE	RSP	iBOT	DINO	Ours
bike-packing_1	0.473	0.406	0.324	0.622	0.465	<b>0.709</b>
bike-packing_2	0.666	0.663	0.701	0.789	0.726	<b>0.820</b>
blackswan_1	0.899	0.890	0.893	0.892	0.898	<b>0.936</b>
bm-x-trees_1	0.286	0.250	0.305	0.305	0.354	<b>0.414</b>
bm-x-trees_2	0.474	0.490	0.522	0.531	0.535	<b>0.623</b>
breakdance_1	0.643	0.708	0.695	0.642	0.675	<b>0.796</b>
camel_1	0.720	0.724	0.655	0.743	0.711	<b>0.802</b>
car-roundabout_1	0.887	<b>0.896</b>	0.868	0.894	0.842	0.863
car-shadow_1	0.791	0.768	<b>0.855</b>	<b>0.855</b>	0.850	0.833
cows_1	0.852	0.841	0.862	0.872	0.873	<b>0.932</b>
dance-twirl_1	0.671	0.693	0.737	0.757	0.785	<b>0.861</b>
dog_1	0.804	0.827	0.851	0.884	0.882	<b>0.921</b>
dogs-jump_1	0.217	0.167	0.108	0.677	0.666	<b>0.745</b>
dogs-jump_2	0.448	0.387	0.265	0.710	0.763	<b>0.851</b>
dogs-jump_3	0.755	0.661	0.722	0.727	0.723	<b>0.823</b>
drift-chicane_1	0.564	0.376	0.625	0.752	0.691	<b>0.835</b>
drift-straight_1	0.543	0.394	0.561	0.716	0.796	<b>0.867</b>
goat_1	0.818	0.778	0.814	0.838	0.843	<b>0.873</b>
gold-fish_1	0.689	0.512	0.689	0.801	0.771	<b>0.834</b>
gold-fish_2	0.635	0.422	0.550	0.656	0.647	<b>0.826</b>
gold-fish_3	0.770	0.636	0.755	0.763	0.771	<b>0.866</b>
gold-fish_4	0.788	0.600	0.795	0.813	0.795	<b>0.889</b>
gold-fish_5	0.836	0.732	0.848	0.868	0.824	<b>0.907</b>
horsejump-high_1	0.693	0.633	0.695	0.701	0.700	<b>0.779</b>
horsejump-high_2	0.438	0.311	0.503	0.560	0.572	<b>0.662</b>
india_1	0.580	0.569	<b>0.601</b>	0.590	0.581	<b>0.601</b>
india_2	0.460	0.434	0.495	0.473	0.492	<b>0.516</b>
india_3	<b>0.582</b>	0.487	0.527	0.430	0.555	0.493
judo_1	0.672	0.720	0.710	0.662	0.731	<b>0.844</b>
judo_2	0.620	0.647	0.647	0.493	0.703	<b>0.794</b>
kite-surf_1	0.192	0.165	0.205	0.226	0.203	<b>0.255</b>
kite-surf_2	0.001	0.000	0.006	0.001	0.001	<b>0.104</b>
kite-surf_3	0.529	0.376	0.493	0.590	0.561	<b>0.732</b>
lab-coat_1	0.000	0.000	0.000	0.000	0.000	0.000
lab-coat_2	0.000	0.000	0.000	0.000	0.000	0.000
lab-coat_3	0.714	0.560	0.812	0.708	0.785	<b>0.852</b>
lab-coat_4	0.737	0.535	0.758	0.708	0.760	<b>0.835</b>
lab-coat_5	0.659	0.375	0.727	0.788	0.809	<b>0.836</b>
libby_1	0.710	0.690	0.707	0.704	0.708	<b>0.813</b>
loading_1	0.854	0.864	0.761	0.882	0.878	<b>0.935</b>
loading_2	0.319	0.134	0.288	0.425	0.361	<b>0.612</b>
loading_3	0.640	0.570	0.594	0.703	0.743	<b>0.771</b>
mbike-trick_1	0.497	0.394	0.528	0.521	0.524	<b>0.596</b>
mbike-trick_2	0.596	0.571	0.609	0.637	0.590	<b>0.662</b>
motocross-jump_1	0.433	0.303	0.452	<b>0.479</b>	0.453	0.468
motocross-jump_2	0.409	0.105	0.572	0.712	0.691	<b>0.753</b>
paragliding-launch_1	0.757	0.770	0.769	0.760	0.754	<b>0.800</b>
paragliding-launch_2	0.425	0.413	0.502	0.529	0.578	<b>0.682</b>
paragliding-launch_3	0.016	0.020	0.023	0.048	0.047	<b>0.068</b>
parkour_1	0.764	0.608	0.819	0.792	0.797	<b>0.884</b>
pigs_1	0.632	0.792	0.803	0.824	0.827	<b>0.873</b>
pigs_2	0.443	0.491	0.549	0.501	0.547	<b>0.655</b>
pigs_3	0.859	0.873	0.894	0.891	0.898	<b>0.941</b>
scooter-black_1	0.118	0.073	0.152	0.256	0.222	<b>0.320</b>
scooter-black_2	0.570	0.375	0.696	0.695	0.680	<b>0.731</b>
shooting_1	0.171	0.144	0.221	0.255	0.485	<b>0.590</b>
shooting_2	0.625	0.688	0.766	0.728	0.793	<b>0.867</b>
shooting_3	0.535	0.466	0.717	0.658	0.735	<b>0.760</b>
soapbox_1	0.760	0.648	0.472	0.633	0.657	<b>0.827</b>
soapbox_2	0.386	0.288	0.305	0.323	0.286	<b>0.643</b>
soapbox_3	0.211	0.288	0.369	0.336	0.220	<b>0.658</b>

Table 8.  $\mathcal{F}_m$  Scores Comparison Across Methods on DAVIS 2017

Sequence	CropMAE	DropMAE	RSP	iBOT	DINO	Ours
bike-packing_1	0.660	0.627	0.513	0.732	0.583	<b>0.854</b>
bike-packing_2	0.712	0.687	0.728	0.809	0.702	<b>0.879</b>
blackswan_1	0.929	0.912	0.922	0.928	0.932	<b>0.964</b>
bm-x-trees_1	0.663	0.618	0.681	0.726	0.682	<b>0.830</b>
bm-x-trees_2	0.610	0.636	0.675	0.697	0.700	<b>0.861</b>
breakdance_1	0.627	0.713	0.688	0.651	0.668	<b>0.796</b>
camel_1	0.731	0.739	0.735	0.836	0.769	<b>0.874</b>
car-roundabout_1	0.801	0.783	0.752	0.788	0.690	<b>0.765</b>
car-shadow_1	0.791	0.690	<b>0.817</b>	0.797	0.761	0.745
cows_1	0.792	0.790	0.821	0.867	0.864	<b>0.966</b>
dance-twirl_1	0.638	0.653	0.710	0.751	0.737	<b>0.886</b>
dog_1	0.803	0.760	0.845	0.868	0.868	<b>0.954</b>
dogs-jump_1	0.359	0.215	0.181	0.664	0.659	<b>0.843</b>
dogs-jump_2	0.455	0.343	0.323	0.696	0.783	<b>0.870</b>
dogs-jump_3	0.855	0.735	0.831	0.836	0.827	<b>0.907</b>
drift-chicane_1	0.599	0.428	0.676	0.808	0.716	<b>0.965</b>
drift-straight_1	0.459	0.311	0.427	0.617	0.561	<b>0.795</b>
goat_1	0.765	0.676	0.760	0.834	0.839	<b>0.896</b>
gold-fish_1	0.624	0.426	0.617	0.784	0.709	<b>0.830</b>
gold-fish_2	0.652	0.413	0.552	0.727	0.681	<b>0.914</b>
gold-fish_3	0.760	0.555	0.735	0.801	0.795	<b>0.913</b>
gold-fish_4	0.788	0.515	0.813	0.845	0.820	<b>0.947</b>
gold-fish_5	0.766	0.522	0.744	0.864	0.748	<b>0.930</b>
horsejump-high_1	0.752	0.692	0.751	0.815	0.763	<b>0.905</b>
horsejump-high_2	0.674	0.561	0.712	0.767	0.774	<b>0.847</b>
india_1	0.482	0.446	0.513	0.518	0.494	<b>0.551</b>
india_2	0.387	0.358	0.442	0.424	0.451	<b>0.493</b>
india_3	0.495	0.365	0.474	0.436	0.469	<b>0.528</b>
judo_1	0.660	0.727	0.711	0.734	0.765	<b>0.882</b>
judo_2	0.615	0.609	0.609	0.571	0.701	<b>0.809</b>
kite-surf_1	0.104	0.100	0.117	0.166	0.154	<b>0.418</b>
kite-surf_2	0.005	0.013	0.121	0.013	0.008	<b>0.396</b>
kite-surf_3	0.758	0.508	0.731	0.792	0.752	<b>0.918</b>
lab-coat_1	0.000	0.000	0.000	0.000	0.000	0.000
lab-coat_2	0.000	0.000	0.000	0.000	0.000	0.000
lab-coat_3	0.579	0.502	0.672	0.578	0.725	<b>0.759</b>
lab-coat_4	0.550	0.439	0.567	0.589	0.659	<b>0.698</b>
lab-coat_5	0.580	0.413	0.722	0.718	0.753	<b>0.787</b>
libby_1	0.850	0.839	0.848	0.852	0.849	<b>0.960</b>
loading_1	0.722	0.717	0.677	0.796	0.821	<b>0.903</b>
loading_2	0.368	0.256	0.395	0.434	0.412	<b>0.603</b>
loading_3	0.629	0.609	0.638	0.770	0.803	<b>0.841</b>
mbike-trick_1	0.582	0.596	0.653	0.571	0.548	<b>0.719</b>
mbike-trick_2	0.565	0.598	0.568	0.562	0.510	<b>0.708</b>
motocross-jump_1	0.458	0.357	0.483	0.488	0.466	<b>0.552</b>
motocross-jump_2	0.393	0.182	0.497	0.563	0.551	<b>0.729</b>
paragliding-launch_1	0.815	0.851	0.810	0.835	0.852	<b>0.922</b>
paragliding-launch_2	0.702	0.680	0.733	0.792	0.814	<b>0.936</b>
paragliding-launch_3	0.063	0.052	0.091	0.108	<b>0.167</b>	0.161
parkour_1	0.800	0.588	0.876	0.824	0.843	<b>0.964</b>
pigs_1	0.577	0.704	0.717	0.795	0.775	<b>0.886</b>
pigs_2	0.481	0.569	0.661	0.609	0.642	<b>0.788</b>
pigs_3	0.733	0.739	0.808	0.825	0.816	<b>0.932</b>
scooter-black_1	0.288	0.293	0.336	0.515	0.467	<b>0.554</b>
scooter-black_2	0.491	0.477	0.588	0.604	0.577	<b>0.657</b>
shooting_1	0.253	0.309	0.222	0.295	0.439	<b>0.581</b>
shooting_2	0.508	0.612	0.755	0.706	0.717	<b>0.804</b>
shooting_3	0.708	0.630	0.892	0.857	<b>0.929</b>	0.927
soapbox_1	0.773	0.661	0.529	0.604	0.613	<b>0.848</b>
soapbox_2	0.528	0.458	0.388	0.401	0.406	<b>0.717</b>
soapbox_3	0.376	0.375	0.509	0.491	0.411	<b>0.749</b>