Trajectory-aligned Space-time Tokens for Few-shot Action Recognition

Pulkit Kumar[1], Namitha Padmanabhan[1], Luke Luo[1], Sai Saketh Rambhatla[1,2], and Abhinav Shrivastava[1]

[1] University of Maryland, College Park
{pulkit,namithap,lluo1,abhinav2}@umd.edu
[2] GenAI, Meta
rssaketh@meta.com

Abstract. We propose a simple yet effective approach for few-shot action recognition, emphasizing the disentanglement of motion and appearance representations. By harnessing recent progress in tracking, specifically point trajectories and self-supervised representation learning, we build trajectory-aligned tokens (TATs) that capture motion and appearance information. This approach significantly reduces the data requirements while retaining essential information. To process these representations, we use a Masked Space-time Transformer that effectively learns to aggregate information to facilitate few-shot action recognition. We demonstrate state-of-the-art results on few-shot action recognition across multiple datasets. Our project page is available here.

Keywords: Few-Shot Action Recognition · Space-time Transformers · Trajectory-aligned representations

1 Introduction

We present a remarkably simple yet highly effective approach for few-shot action recognition [9, 55, 57, 61, 73]. Unlike traditional large-scale training paradigms, the few-shot setting [30, 41, 54, 56] demands a nuanced understanding of what constitutes an action. This requires discerning subtle cues in both motion and appearance, which are often confined to just a few pixels. We argue that, distinct from the assumptions made in large-scale action recognition, where such information can be implicitly learned in deep networks using lots of training samples, few-shot scenarios demand a more deliberate approach to model the interplay between motion and appearance cues. To address this, we propose an approach that disentangles motion and appearance representations. By leveraging recent advances in each domain, it selects a limited yet relevant subset of information to aid in learning what constitutes an action.

Our approach first models motion and then builds appearance representation aligned with this motion, which is better suited for a low-shot training paradigm with a higher signal-to-noise ratio. To model motion, we utilize the recent works on tracking a set of points across frames based on its motion [17,18,24,29,63,72].

This point-tracking paradigm for capturing motion has an inherent advantage over motion representations, such as optical flow and frame differences. First, they can persist through occlusions and capture multi-frame context, resulting in longer-term and robust motion information; and second, the set of points being tracked are often orders of magnitude smaller than a number of pixels, resulting in fewer data (i.e., point trajectories) with minimal loss of information. Next, our approach is to build appearance representations aligned with these point tracks or trajectories to capture semantics. Towards this, we leverage recent advances in self-supervised methods (*e.g.*, DINOv2 [38]) that are pre-trained on large datasets resulting in general-purpose vision features. The DINO family of models is particularly suited for this approach due to their self-distillation framework, which encourages consistency between different views of the same image and results in semantic representations that robustly and efficiently capture the variations introduced by actions. Concretely, we select trajectory-aligned tokens given the point trajectories and space-time DINO tokens, which effectively capture both motion and appearance cues.

Finally, we use a masked space-time Transformer architecture [2, 6] that operates on these trajectory-aligned space-time tokens and aggregates information for each point over time and between different trajectories. The output representation of this Transformer, which captures both motion and appearance information, is then used for few-shot learning in a standard metric learning formulation [7, 9, 37, 40, 57, 58, 60, 68, 70, 71]. In particular, we use the bidirectional Mean Hausdorff Metric (Bi-MHM) from [57] to analyze semantic correlations across videos and align frames to perform few-shot recognition. To summarize, our key contributions are:

- We propose a simple yet effective approach for few-shot action recognition that disentangles motion and appearance representations.
- We leverage recent advances in tracking (*i.e.*, point trajectories) and self-supervised representation learning (*e.g.*, DINO) to get trajectory-aligned tokens that result in drastically less data with minimal loss of information.
- We propose a Masked Space-time Transformer to learn from trajectory-aligned features and aggregate information for few-shot action recognition.

2 Related Work

Few-shot action recognition. Video contains rich spatio-temporal context interactions and temporal variations [30, 41, 54, 56]. Existing methods for few-shot action recognition predominantly apply the metric-based learning paradigm but with different focuses. Some methods focus on feature representation enhancement and spatio-temporal modelling [48, 57, 61, 64, 73, 74]. For instance, CMN [73] integrates a memory structure and multi-saliency algorithm to encode video representations.MTFAN [61] and GgHM [65] prioritize task-specific feature learning with MTFAN using a motion encoder to integrate global motion patterns and GgHM optimizing intra- and inter-class feature correlations explicitly. SloshNet [64] employs a feature fusion module to combine low- and

high-level spatial features, and a long-term temporal modelling module captures global temporal relations from spatial appearance features. MoLo [55] introduces a motion autodecoder with a long-short contrastive objective to extract motion dynamics explicitly. Other methods focus on designing effective metric learning strategies [7, 9, 21, 37, 40, 57, 58, 60, 68, 70, 71]. OTAM [9] uses a differentiable dynamic time warping algorithm [34] with an ordered temporal alignment for estimating the distance between the query video and the support set videos. ARN [68] introduces a self-supervised permutation invariant strategy and learns long-range temporal dependencies by building on a 3D backbone [51]. TRX [40] matches each query sub-sequence with all sub-sequences in the support set to classify actions. HyRSM [57, 58] uses hybrid relation modelling and introduces the bidirectional mean Hausdorff metric (Bi-MHM), which is adopted by several other works. ITANET [70] and STRM [48] introduce joint spatio-temporal modelling techniques for robust few-shot matching. Nguyen et al. [36] and Huang et al. [28] utilize multiple similarity functions to compare videos accurately. On the other hand, our method aims to leverage advancements in point-tracking literature and image representation learning literature to tackle few-shot action recognition.

Point Tracking. TAP-Vid [17] introduced the problem of tracking any physical point in a video and proposed the TAP benchmark with a simple baseline method, TAPNet. However, this method cannot track occluded points. PIPs [24] builds on the classic concept of "particle video" [42] that acts as a bridge between local feature matching [5, 15, 32, 44, 49] and optical flow [19, 35, 45, 47, 66, 67] based methods to leverage long-range temporal priors in video motion estimation. PIPs process videos in fixed-sized temporal windows and track points through occlusions, but they lose the target for longer occlusions that extend beyond the window size. TAPIR [18] integrates the global matching strategy from TAPNet with the refinement step of PIPs to enhance tracking accuracy. PointOdyssey [72] addresses long-term tracking with PIPs++, a modification of PIPs that significantly extends its temporal receptive field, and introduces a benchmark for the task of fine-grained long-term tracking. OmniMotion [53] also addresses per-pixel per-frame trajectory estimation by ensuring global consistency of estimated motion for robust tracking through occlusions.DOT [33] proposes a faster and more memory-efficient dense point tracking algorithm for use in practice. While the above methods estimate individual point trajectories independently, CoTracker [29] proposes to jointly estimate point trajectories to take advantage of the correlation between points, resulting in improved performance. We adopt the point representation of motion information in our work. Specifically, we integrate CoTracker in our method as it can efficiently and jointly track multiple points.

Image Representation Learning. In image representation learning, two key approaches are intra-image self-supervised training and discriminative self-supervised learning. Intra-image methods utilize pretext tasks like predicting image patches

[16], re-colorizing [69], or inpainting [39] to learn features. Patch-based architectures like ViTs have revived inpainting for pre-training [4, 20, 25], with masked auto-encoders (MAEs) demonstrating effectiveness in learning transferable features [25, 50]. On the other hand, Discriminative self-supervised learning focuses on learning features through discriminative signals between images or groups of images. This approach has evolved from early deep learning work [22] to instance classification methods [1, 8, 62] and further improvements based on instance-level objectives [12, 14, 26, 27] or clustering [3, 10, 11]. We leverage DINOv2 [38] for feature extraction in our work on few-shot action recognition. DINO [12] and DINOv2 stand out for their ability to learn rich, transferable visual representations without relying on labels. DINO introduced a self-distillation framework for learning semantically meaningful features without labels. DINOv2 builds on this, offering improved scalability and applicability across tasks. By employing DINOv2, we aim to harness the advancements in self-supervised learning for effective feature extraction in our pipeline.

3 Method

In this section we describe the details of our approach. We begin by refreshing a few preliminaries in Sec. 3.1 that we used in to build our method and then proceed to describe our method in Sec. 3.2

3.1 Preliminaries

Task Details. In the traditional supervised learning setting, the model is trained and tested on the same set of classes. However, in the few-shot regime, the model is trained on a specific set of base classes, denoted as C_{base}, and then evaluated on a separate set of *novel* classes, C_{novel}. There is no overlap in the classes between the base and novel classes, i.e., $C_{\text{base}} \cap C_{\text{novel}} = \phi$. For few-shot action recognition, prior works [9, 40, 55, 57] typically adopt the *N-way K-Shot* episode-based meta-learning strategy [52]. Each episode comprises a support set S consisting of N classes with K labelled samples per class and a query set Q containing query samples to be classified into one of the N classes. The objective of few-shot action recognition is to learn to predict the label of a sample video from the query set using information from the support set. For training, the models train for a certain number of episodes consisting of the training data. To report the final numbers, 10000 random episodes are formed from the test data and average accuracy is reported.

Point Details. Given an RGB video with T frames and a point initialised at frame q with coordinates (x_q, y_q), where $q \in [1, T)$, a point tracker [29] tracks the point's movement across frames, yielding outputs at each time step, (x_t, y_t), where $t \in [q, T]$.

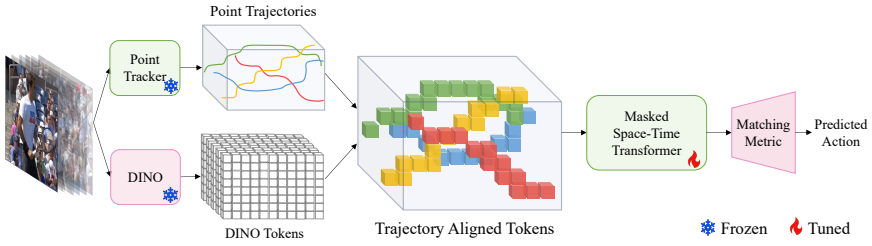

Fig. 1: Overview of our method. We take in video frames as input and extract point trajectories and DINO patch tokens using a Point Tracker and DINO respectively. These trajectories and tokens are then aligned using a grid sampler to form trajectory-aligned tokens (TATs). Finally, we pass the TATs through a masked space-time transformer and use a matching metric on the output embedding to predict the query action.

DINO. Vision Transformer (ViT) is a transformer-based model designed, where convolutions are substituted with self-attention mechanisms. The input image is divided into fixed-size patches, $\mathbf{x}_p \in \mathbb{R}^{N \times C}$, where M signifies the number of patches and C denotes the patch dimensions. The resulting output is $\mathbf{x}_o \in \mathbb{R}^{M \times D}$, where D is the hidden dimension of the network. By leveraging ViTs, DINO [38] integrates self-training and knowledge distillation without relying explicitly on labels, thereby facilitating self-supervised learning. Using a teacher-student network, DINO synthesizes global and local views from an image, with the teacher distilling its insights to inform the student network. This approach enables DINO to encapsulate a significant amount of semantic information.

3.2 Method

We first provide an overview of our method and then give more details about each component. As shown in Fig.1, given a video, we sample and track a set of points on a uniform grid using a point tracker. Simultaneously, patch tokens are extracted from each frame using DINOv2 [38], and the patch token corresponding to each point's location serves as its semantic descriptor. The semantic information of each point, combined with its corresponding patch token, forms the Trajectory Aligned Tokens (TATs), which are then processed by the Masked Space-Time Transformer. The weights of the point tracker and DINOv2 models are frozen, and the Space-Time Transformer is trained using a combination of cross-entropy and contrastive losses.

Point extraction For an input video of dimensions $H \times W \times T$, we utilize Co-Tracker [29] to extract a set of point trajectories across the video. Specifically, we uniformly sample a set of points on a grid in the first frame and track them using Co-Tracker. We initialize points uniformly across multiple frames to successfully track points on objects that appear at a later frame. This ensures that points

on objects that appear after the first frame are tracked faithfully. To prevent tracking duplicates, we discard any new trajectory that is similar to the existing ones. Finally, N trajectories are randomly sampled to form the set $P \in R^{T \times N}$ described by $\{(x_i^t, y_i^t), \forall t \in [q, T], \forall i \in [1, N], \forall q \in [1, T-1]\}$. We ablate over the number of points and the sampling strategies in Section 4.3.

Trajectory-aligned Tokens (TATs) The spatio-temporal points in P, extracted from the video, lack semantic information. To enrich these points with semantics, we leverage the DINOv2 patch tokens extracted from each frame. We extract DINO patch tokens for all the frames in the video denoted as $\mathbf{x}_{\text{DINO}} \in \mathbb{R}^{T \times M \times D}$, where M is the number of patches and D is the dimension of the features. For a point $p \in P$ at frame q, we use the feature of the patch token at this location as its feature descriptor. We employ a grid sampler, that operates on P and $\mathbf{x}_{\text{DINO}}$, and uses nearest neighbor interpolation to obtain $\mathbf{x}_{\text{TATs}} \in \mathbb{R}^{T \times N \times D}$. We term the points equipped with semantic information as the trajectory-aligned tokens (TATs).

Masked Space-Time Transformer. Given the TATs, our objective is twofold: 1) to assimilate the information of each point along its trajectory, and 2) to model inter-point relationships. To achieve this, we employ a transformer equipped with the space-time attention mechanism [2, 6]. The temporal attention module helps gather point-specific information along their trajectories, while the spatial attention mechanism aggregates information across all points within each frame. To consolidate the spatio-temporal information, an additional [CLS] token is appended to the input.

The Transformer produces an embedding $\tilde{f} \in \mathbb{R}^{T*N+1 \times D}$. Previous approaches aggregate information along the spatial dimension while retaining the temporal dimension intact. Similarly, our method aggregates point information at each frame, resulting in an embedding $f \in \mathbb{R}^{T \times D}$. Additionally, the [CLS] token undergoes separate processing through a head layer, producing an embedding $f^{\texttt{CLS}} \in \mathbb{R}^{1 \times C_{\text{base}}}$, where C_{base} denotes the number of base classes.

It is important to note that not all points utilized by TATs are initialized from the initial frame; some points are initialized in subsequent frames. To facilitate processing by the transformer, we integrate a masked attention module. When considering a point initialized at time t, the trajectory of the point before time t will be masked out during attention calculation.

Set matching metric and losses. The embeddings f and $f^{\texttt{CLS}}$ are extracted for all samples in the support and the query set as described above. To classify a query sample using support samples, we adopt the set matching strategy utilized in previous methods [55, 57, 58], which employ the bidirectional Mean Hausdorff Metric (Bi-MHM) [57]. Our approach also incorporates losses akin to those used in [55]. Specifically, we apply a cross-entropy loss on the [CLS] token to capture global class information and use a contrastive loss based on the Bi-MHM metric. For the equations, we refer to [55] and [57].

4 Experiments

This section begins by detailing the experimental configurations of our method in Section 4.1. We then compare our approach with established state-of-the-art methods across several widely-used benchmarks in Section 4.2. Additionally, in Section 4.3, we provide ablations of our method, explaining the various design decisions made. Finally, Section 4.4 presents qualitative results that compare our method with previous works.

4.1 Experimental setup

Datasets. To demonstrate the efficacy of our approach, we evaluate the few-shot splits of Something-Something [23], Kinetics [13], UCF101 [46], and HMDB51 [31] adopting the same splits as previous works [55, 57, 58, 65] for a fair comparison. Two splits of Something-Something were introduced [9] - SSv2-Small and SSv2-Full. SSv2-Full includes all classes, while SSv2-Small has 100 samples per class. For Kinetics, we used the split information from [73]. For UCF101 and HMDB51, we used the splits from [61, 68]. Additionally, to evaluate our method on fine-grained actions, we created a few-shot split of the Fine-Gym [43] dataset and reported its results.

Implementation details. For extraction of points, we used Co-Tracker [29] for its ability to jointly and efficiently track multiple points. Since only the information of points is being used from the point tracker and they are *not* fine tuned, any other methods [18, 24, 53, 72] could be used also. The points were extracted at 10 FPS. DINOv2 [38] was used for extracting the DINO features. Following previous methods [55, 57, 58, 65], eight frames ($T = 8$) were uniformly sampled for every video. Note that the frame-wise DINO features and the points used by our method correspond to the sampled frames. As for our transformer architecture similar to [55] was adopted. Only the transformer is trained in our setup, while the rest of the modules are frozen that leads to significant drop in training parameters as compared to previous methods; discussed in detail in the supplementary. Furthermore, we used, similar training and inference strategies were used in previous works [55, 57, 58, 65]. Final metric reported is the mean accuracy across 10,000 random episodes from the test set.

4.2 Comparison with state-of-the-art methods

We compare our method with previous state-of-the-art methods under the 5-way K-shot setting. We report results with K varying from 1 to 5 on Kinetics and SSV2-Full in Table 1. We also report results on SSV2-small, UCF101 and HMDB51 for $K = \{1, 3, 5\}$ in Table 2. Additionally, we report results on a few-shot setting on Fine-Gym in Table 3

We observe a consistent improvement in performance across both the few-shot settings of Kinetics and SSv2 Full datasets. In the 1-shot setting of Kinetics,

Table 1: Comparison of our method with contemporary methods of few-shot action recognition on Kinetics and SSV2 Full datasets for classification accuracy from the 1-shot up to the 5-shot settings. Entries in "-" mean the data is not available in published works. The best results for each are bolded. Second best results are underlined.

		Kinetics					**SSV2 Full**				
Method	Reference	1-shot	2-shot	3-shot	4-shot	5-shot	1-shot	2-shot	3-shot	4-shot	5-shot
OTAM [9]	CVPR'20	72.2	75.9	78.7	81.9	84.2	42.8	49.1	51.5	52.0	52.3
TRX [40]	CVPR'21	63.6	76.2	81.8	83.4	85.2	42.0	53.1	57.6	61.1	64.6
STRM [48]	CVPR'22	62.9	76.4	81.1	83.8	86.7	43.1	53.3	59.1	61.7	68.1
MTFAN [61]	CVPR'22	74.6	-	-	-	87.4	45.7	-	-	-	60.4
HYRSM [57]	CVPR'22	73.7	80.0	83.5	84.6	86.1	54.3	62.2	65.1	67.9	69.0
HCL [71]	ECCV'22	73.7	79.1	82.4	84.0	85.8	47.3	54.5	59.0	62.4	64.9
Nguyen et al [36]	ECCV'22	74.3	-	-	-	87.4	43.8	-	-	-	61.1
Huang et al [28]	ECCV'22	73.3	-	-	-	86.4	49.3	-	-	-	66.7
MoLo [55]	CVPR'23	74.0	80.4	83.7	84.7	85.6	<u>56.6</u>	62.3	<u>67.0</u>	68.5	<u>70.6</u>
SloshNet [64]	AAAI'23	70.4	-	-	-	87.0	46.5	-	-	-	68.3
GgHM [65]	ICCV'23	74.9	-	-	-	87.4	54.5	-	-	-	69.2
CCLN [59]	PAMI'24	<u>75.8</u>	<u>82.1</u>	<u>85.0</u>	<u>86.1</u>	<u>87.5</u>	46.0	-	-	-	61.3
HYRSM++ [58]	PR'24	74.0	80.8	83.9	85.3	86.4	55.0	<u>63.5</u>	66.0	<u>68.8</u>	69.8
Ours	-	**81.9**	**86.5**	**89.9**	**90.6**	**91.1**	**57.7**	**67.1**	**70.0**	**70.6**	**74.6**

we observed a 6.1% increase compared to CCLN [59], along with a 4.3% enhancement in the 2-shot setting. The 3-shot setting showed an increase of 4.9%, and the 4-shot setting demonstrated a gain of 5.3%. Finally, in the 5-shot setting, we observed a 3.6% improvement. In SSv2 Full, we report improvements on almost all shots compared to the previous methods. Specifically, in the 1-shot, 2-shot, and 3-shot settings, we observed gains of 2.7%, 3.9%, and 3% respectively. We further noticed an improvement of 2.1% and 4% in 4-shot and 5-shot settings, respectively.

For SSv2 Small, we notice a 5.1% gain compared to [58] in 1-shot setting. For the 3-shot setting, we notice a performance gain of 6% as compared to MoLo [55] and for the 5-shot setting, we notice a gain of 3.8% as compared to [28]. For UCF-101, we notice a 5.1% gain in the 1-shot setting and a 2.4% gain in the 3-shot setting as compared to CCLN [59]. That said, we do see a drop of 0.8% in the 5-shot setting. For HMDB-51, we notice a drop of 5.1%, 4.4% and 1.8% as compared to the CCLN [59]. We noticed this as a limitation to our method and have discussed this in the limitations of the paper.

We observe similar trends in the Kinetics and SSV2 splits for the few-shot setting in Fine-Gym. We compare the performance of MoLo [55] with our method and find that in the 1-shot setting, our method shows a performance gain of 8.5%. In the 3-shot setting, an increase of 5.8% is observed, and in the 5-shot setting, there is a gain of 3.1%. This indicates the effectiveness of our method on fine-grained actions.

Number of frames. Similar to previous approaches, we assess our performance with varying input frames and present the results in Fig 2. Focusing on the 5-way 1-shot setup in the SSV2-Full configuration, our method initially shows

Table 2: Comparison of our method with contemporary methods of few-shot action recognition on SSV2 Small, and UCF-101, and HMDB-51 datasets for classification accuracy on the 1-shot, 3-shot, and 5-shot settings. Entries with "-" mean the data is not available in published works. The best results for each are bolded. Second best results are underlined.

		SSV2 Small			**UCF-101**			**HMDB-51**		
Method	**Reference**	1-shot	3-shot	5-shot	1-shot	3-shot	5-shot	1-shot	3-shot	5-shot
OTAM [9]	CVPR'20	36.4	45.9	48.0	79.9	87.0	88.9	54.5	65.7	68.0
TRX [40]	CVPR'21	36.0	51.9	56.7	78.2	92.4	96.1	53.1	66.8	75.6
STRM [48]	CVPR'22	37.1	49.2	55.3	80.5	92.7	**96.9**	52.3	67.4	77.3
MTFAN [61]	CVPR'22	-	-	-	84.8	-	95.1	-	-	-
HYRSM [57]	CVPR'22	40.6	52.3	56.1	83.9	93.0	94.7	60.3	71.7	76.0
HCL [71]	ECCV'22	38.7	49.1	55.4	82.5	91.0	93.9	59.1	71.2	76.3
Nguyen et al [36]	ECCV'22	-	-	-	-	-	-	59.6	-	76.9
Huang et al [28]	ECCV'22	38.9	-	<u>61.6</u>	71.4		91.0	60.1	-	77.0
MoLo [55]	CVPR'23	42.7	<u>52.9</u>	56.4	86.0	93.5	95.5	60.8	72.0	<u>77.4</u>
GgHM [65]	ICCV'23	-	-	-	85.2	-	<u>96.3</u>	61.2	-	76.9
CCLN [59]	PAMI'24	-	-	-	<u>86.9</u>	<u>94.2</u>	96.1	**65.1**	**76.2**	**78.8**
HYRSM++ [58]	PR'24	<u>42.8</u>	52.4	58.0	85.8	93.5	95.9	<u>61.5</u>	<u>72.7</u>	76.4
Ours		**47.9**	**60.0**	**64.4**	**92.0**	**96.8**	95.5	60.0	71.8	77.0

Table 3: Few-shot accuracy on Fine-Gym dataset.

	FineGym		
Method	1-shot	3-shot	5-shot
MoLo [55]	73.3	80.2	84.8
Ours w/o points	77.6	81.8	86.4
Ours	**81.8**	**86.0**	**87.9**

Table 4: GFLOPS, inference time and training parameters. MST - Masked transformer

	GFLOPS			
Method	**Train**	**Test**	**Time**	**Params**
MoLo [55]	252	84	0.67s	89.6M
Ours w/o points	327	245	0.73s	10.8M
Ours	363	281	0.96s	10.8M
CoTracker/Dino/MST	(36/204/123)	(36/204/41)	(.23s/.24s/.49s)	(-/-/10.8M)

subpar performance with only two input frames, primarily due to its reliance on point trajectories. With just two frames, limited information leads to reduced performance. However, as the number of input frames increases, our method surpasses all previous methods, benefiting from the richer trajectory information provided by more frames.

Compute analysis In Table 4, we compare our model's GFLOPS, inference time, and trainable parameters to MoLo's in a 5-way 1-shot setting. Integrating Cotrakcer and DINO increases speed and GFLOPS compared to MoLo, but the performance improvement justifies this trade-off. Additionally, our model has significantly fewer trainable parameters since neither the point tracker nor DINO are trained. Also, both can run separately offline, mitigating the impact on speed and GFLOPS.

Different N-way settings. We also ran experiments on different *N-way* settings. For both Kinetics and SSv2 Full, we run experiments on N ranging from

Table 5: Comparison of our method with contemporary methods of recognition on Kinetics and SSV2 Full datasets for classification accuracy on N-way 1-shot settings. Results are reported for 5 to 10 way settings. The best results for each are bolded. Second best results are underlined.

	Kinetics						**SSV2 Full**					
Method	5-way	6-way	7-way	8-way	9-way	10-way	5-way	6-way	7-way	8-way	9-way	10-way
OTAM [9]	72.2	68.7	66.0	63.0	61.9	59.0	42.8	38.6	35.1	32.3	30.0	28.2
TRX [40]	63.6	59.4	56.7	54.6	53.2	51.1	42.0	41.5	36.1	33.6	32.0	30.3
HyRSM [57]	73.7	69.5	66.6	65.5	63.4	61.0	54.3	50.1	45.8	44.3	42.1	40.0
MoLo [55]	<u>74.0</u>	<u>69.7</u>	<u>67.4</u>	<u>65.8</u>	<u>63.5</u>	<u>61.3</u>	<u>56.6</u>	<u>51.6</u>	<u>48.1</u>	<u>44.8</u>	<u>42.5</u>	<u>40.3</u>
Ours	**81.9**	**79.0**	**76.1**	**75.2**	**72.2**	**72.0**	**57.7**	**55.7**	**52.5**	**50.0**	**47.0**	**45.8**

5 to 10. We observe that on Kinetics, we get almost a 10% increase on previous methods. The performance gain ranges from 2%-5% on the SSv2-Full dataset. Both of these results indicate the efficacy of our proposed method.

4.3 Ablations

In this section, we investigate the different components of our method through a series of ablation studies. We present experiments where no points are used and where points are initialized only on the first frames. Additionally, we demonstrate the impact of sampling varying numbers of points on overall performance. Moreover, we explore the effects of sampling points across different grid sizes. These ablations provide a comprehensive understanding of how each component and configuration contributes to the final performance of our method.

No-point baseline and point initialisation. To study the performance gain of our method, we created the baseline where no point information was used. Patch tokens extracted from DINO [38] were directly passed through the transfer and used for generating tokens. We report the number in the first row of Table 6. We also report performance when the points are initialised in the first frame. We report these numbers in the second row of Table 6. Our method adopts the uniform temporal sampling strategy where the points are reinitialised after a few frames. We report the performance in the third row of the table.

We notice that in a 1-shot setting, the model performed worse in the first frame initialisation regime than in the baseline, where we sampled no points but a gain of performance in the uniform temporal sampling strategy. However, for both 3-shot and 5-shot kinetics settings, both point sampling strategies performed better than the baseline of no points. A similar trend in performance is noticed in SSv2-full as well as SSV2-Small. Our strategy of a) using the points and b) reinitialising the points uniformly in time performs consistently better across all settings amongst the reported datasets. This is mainly attributed to the fact that not all objects are present in the initial frame of a video; some may appear after the first frame. The uniform temporal initialisation strategy works better for catering to such objects.

Table 6: Comparison of uniformly re-initialized point sampling to baseline

	Kinetics			SSv2-Full			SSv2-Small		
Strategy	1-shot	3-shot	5-shot	1-shot	3-shot	5-shot	1-shot	3-shot	5-shot
No points	80.2	88.8	90.2	56.2	68.0	72.5	44.6	57.7	62.5
First frame only	78.6	89.3	**91.1**	56.0	68.5	73.3	46.1	57.4	63.6
Uniform temporal sampling	**81.9**	**89.9**	**91.1**	**57.7**	**70.0**	**74.6**	**47.9**	**60.0**	**64.4**

Table 7: Comparison of different number of sampled points for 1-shot, 3-shot, and 5-shot settings.

	Kinetics			SSv2-Full			SSv2-Small		
Points	1-shot	3-shot	5-shot	1-shot	3-shot	5-shot	1-shot	3-shot	5-shot
32	81.0	86.7	89.4	53.4	66.2	71.8	44.5	57.4	61.5
64	81.8	88.0	89.9	55.5	68.3	72.5	45.7	57.7	62.0
128	81.6	89.0	90.5	56.5	69.1	73.7	46.8	59.9	63.9
256	**81.9**	**89.9**	**91.1**	**57.7**	**70.0**	**74.6**	**47.9**	**60.0**	**64.4**

Number of points sampled. To manage memory constraints, we sample N points from the set of all initialized and tracked points. It is important to note that sampling fewer points reduces the number of tokens the network processes, thereby lowering memory usage. Table 7 presents our performance results with different numbers of randomly sampled points. We observe that while performance improves with an increased number of sampled points, the performance drop is not significant when the number of points is halved. For instance, sampling 128 points instead of 256 in the Kinetics dataset results in only a marginal performance decrease in both the 1-shot and 5-shot settings. Although the points are currently sampled randomly, future sampling could leverage trajectory information such as length and orientation. An effective sampling strategy could significantly improve memory efficiency as well as enhance performance. We further explore and report these results in the supplementary materials.

Grid size of points being sampled. Since no initial point information is provided, we sample them across a uniform grid of size G. In our previous experiments, we initialized the points uniformly on a 16-sized grid. We analyze the impact of grid size in Table 8, focusing on the 5-way 1-shot setting. Across the three dataset settings, we observe a slight performance drop when points are sampled on a 9-sized grid, but performance remains relatively constant when the grid size increases to 25. This finding aligns with our previous observations on ablating for the number of points. It's worth noting that as the grid size increases, so does the number of tokens and, consequently, the model's memory consumption. Hence, there exists a slight trade-off between grid size and memory consumption.

Table 8: Classification accuracy of 5-way 1-shot setting for several grid sizes.

Grid Size	Kinetics	SSv2-Full	SSv2-Small
9×9	81.3	56.0	45.7
16×16	81.9	57.7	**47.9**
25×25	**82.1**	**58.1**	47.7

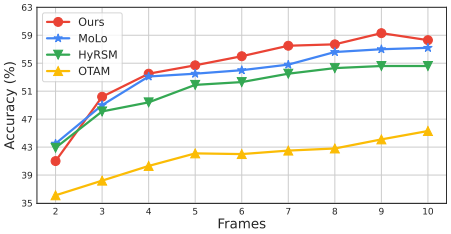

Fig. 2: Effect of number of input frames under the 5-way 1-shot setting on SSv2-Full.

(a) Kinetics

(b) SSv2 Full

(c) SSv2 Small

Fig. 3: Quantitative analysis of 5-way 1-shot setting compared to MoLo. Top: Kinetics dataset; Middle: SSv2 Full dataset; Bottom: SSv2 Small dataset. "S" is a shorthand for "Something".

4.4 Qualitative results

Following [55], we analyze the class-wise accuracy performance of a 5-way 1-shot setting across the three datasets and report them in Fig 3. We compare our method with MoLo [55] across the three dataset settings. In Kinetics, we gain

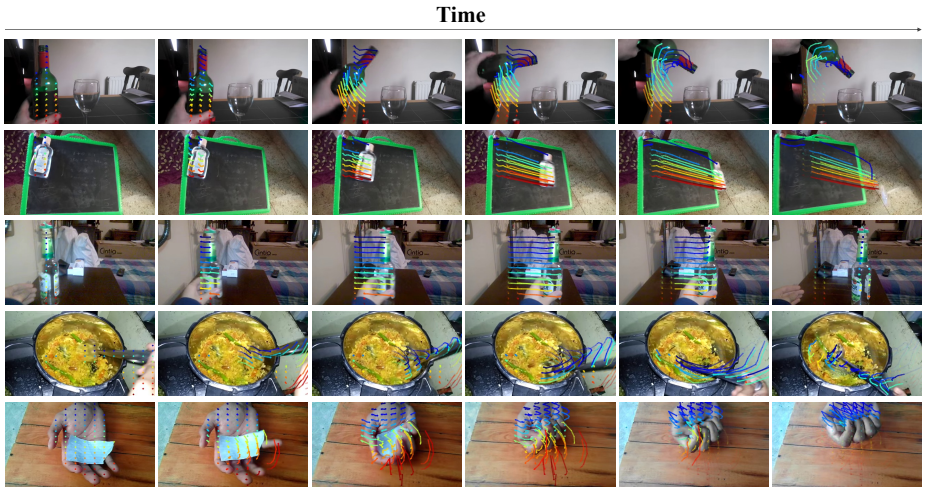

Fig. 4: Qualitative Results. We showcase some results where our method performs better than baselines, attributed to the motion information that is clearly discernible in these samples. The examples are drawn from the Something-Something [23] dataset. The six columns depict the frames of each video sampled across time, and the lines denote the trajectories of tracked points. For visualization purposes, only the points on the most salient object are visualised while the background points are omitted.

significant performance in high-motion classes such as hula hooping, sidekick, blasting sand, etc. But no gain was seen in classes such as cliff diving, unboxing, etc. This limitation is disucssed in Section 5.

In SSv2-full, performance gains were seen in classes like "approaching something with something", "pouring something out of something", "spilling something behind something" showed performance improvement. However, actions like "failing to put something into something", "removing something to reveal something", "showing something next to something", and "tipping something over" did not show improvements. The limitations of our method show that the actions that track points with a persistent spatial position are more challenging to classify.

In SSv2-small, our method shows an improvement in performance for queries such as "Putting something upright", "Rolling something", "Poking a stack of something so it collapses", and "Dropping something". These classes typically feature relevant objects moving into and out of frame, a type of obfuscation that conventional methods of few-shot action recognition struggle to identify. However, classes like "Scooping something", "Putting something on a surface", and "Unfolding something", in which the relative motion of relevant points is noticeably lesser, did not show improvements and is the limitation of our method.

In Fig 4, we visualize several videos from the SSV2 dataset, highlighting the point tracking information. For clarity, we display only the points associated with the salient object, omitting the background points. However, all points, including those in the background, are utilized in our approach.

5 Limitations

Since our methodology draws from two distinct domains, it potentially inherits their respective limitations and errors. Point tracking, for instance, proves highly susceptible to significant camera motion. In scenarios where motion is pronounced, the efficacy of point tracking diminishes, resulting in suboptimal performance for our method. Another challenge we encountered pertains to videos characterized by abrupt scene cuts, a phenomenon not prevalent in SSv2 but commonly found in Kinetics. Managing point tracking under such circumstances proves intricate. Furthermore, we observed a distinct limitation in cases where motion information provides limited assistance in action recognition (e.g., "smoking," "talking"), as our trajectory-aligned features fail to yield discernible improvements. Improvements in point tracking methods can help solve many of the limitations faced.

6 Discussion and future work

This work integrates points and their trajectories using DINOv2 patch tokens to form Trajectory-aligned Tokens (TATs). While we focused on utilizing TATs for few-shot action recognition, we believe they hold potential for other action recognition tasks as well. There are several aspects that we did not address in our current work but could be crucial for solving different tasks. Notably, our work does not account for the visibility of points, which is information provided by point trackers. Point visibility can be essential for fine-grained recognition. Additionally, we recognize the need for improved sampling strategies to select the points used by TATs. Since similar objects carry similar information, not all points on the objects are necessary for the downstream task. A more intelligent sampling strategy would enhance the efficiency of our method. We hope that future research will leverage TATs and explore innovative approaches to address other video-related tasks.

7 Conclusion

In conclusion, we present a straightforward yet impactful method for few-shot action recognition, focusing on separating motion and appearance representations. Leveraging advancements in tracking, particularly point trajectories and self-supervised learning, we introduce trajectory-aligned tokens (TATs) that encode both motion and appearance cues. This technique not only minimizes data needs but also preserves crucial information. Employing a Masked Space-time Transformer to process these representations, we effectively learn to consolidate information for improved few-shot action recognition. Our experimental results showcase leading performance on various datasets, underlining the efficacy of our approach. We believe that our work marks a significant step forward in the field of action recognition, offering a powerful tool for future research and practical applications.

Acknowledgements. We would like to thank Nirat Saini and Matthew Gwilliam for their helpful feedback while we prepared the manuscript. This work was partially supported by NSF CAREER Award (#2238769) to AS. The authors acknowledge UMD's supercomputing resources made available for conducting this research. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF or the U.S. Government.

References

1. Alexey, D., Fischer, P., Tobias, J., Springenberg, M.R., Brox, T.: Discriminative unsupervised feature learning with exemplar convolutional neural networks. IEEE TPAMI **38**(9), 1734–1747 (2016)
2. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: Vivit: A video vision transformer. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6836–6846 (2021)
3. Asano, Y.M., Rupprecht, C., Vedaldi, A.: Self-labelling via simultaneous clustering and representation learning. arXiv preprint arXiv:1911.05371 (2019)
4. Bao, H., Dong, L., Piao, S., Wei, F.: Beit: Bert pre-training of image transformers. arXiv preprint arXiv:2106.08254 (2021)
5. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9. pp. 404–417. Springer (2006)
6. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? In: Proceedings of the International Conference on Machine Learning (ICML) (July 2021)
7. Bishay, M., Zoumpourlis, G., Patras, I.: Tarn: Temporal attentive relation network for few-shot and zero-shot action recognition. arXiv preprint arXiv:1907.09021 (2019)
8. Bojanowski, P., Joulin, A.: Unsupervised learning by predicting noise. In: International Conference on Machine Learning. pp. 517–526. PMLR (2017)
9. Cao, K., Ji, J., Cao, Z., Chang, C.Y., Niebles, J.C.: Few-shot video classification via temporal alignment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10618–10627 (2020)
10. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: Proceedings of the European conference on computer vision (ECCV). pp. 132–149 (2018)
11. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. Advances in neural information processing systems **33**, 9912–9924 (2020)
12. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the International Conference on Computer Vision (ICCV) (2021)
13. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 4724–4733 (2017)

14. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15750–15758 (2021)
15. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 224–236 (2018)
16. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE international conference on computer vision. pp. 1422–1430 (2015)
17. Doersch, C., Gupta, A., Markeeva, L., Recasens, A., Smaira, L., Aytar, Y., Carreira, J., Zisserman, A., Yang, Y.: Tap-vid: A benchmark for tracking any point in a video. Advances in Neural Information Processing Systems **35**, 13610–13626 (2022)
18. Doersch, C., Yang, Y., Vecerik, M., Gokay, D., Gupta, A., Aytar, Y., Carreira, J., Zisserman, A.: Tapir: Tracking any point with per-frame initialization and temporal refinement. arXiv preprint arXiv:2306.08637 (2023)
19. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2758–2766 (2015)
20. El-Nouby, A., Izacard, G., Touvron, H., Laptev, I., Jegou, H., Grave, E.: Are large-scale datasets necessary for self-supervised pre-training? arXiv preprint arXiv:2112.10740 (2021)
21. Fu, Y., Zhang, L., Wang, J., Fu, Y., Jiang, Y.G.: Depth guided adaptive meta-fusion network for few-shot video recognition. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 1142–1151 (2020)
22. Goyal, P., Caron, M., Lefaudeux, B., Xu, M., Wang, P., Pai, V., Singh, M., Liptchinsky, V., Misra, I., Joulin, A., et al.: Self-supervised pretraining of visual features in the wild. arXiv preprint arXiv:2103.01988 (2021)
23. Goyal, R., Kahou, S.E., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fründ, I., Yianilos, P.N., Mueller-Freitag, M., Hoppe, F., Thurau, C., Bax, I., Memisevic, R.: The "something something" video database for learning and evaluating visual common sense. 2017 IEEE International Conference on Computer Vision (ICCV) pp. 5843–5851 (2017)
24. Harley, A.W., Fang, Z., Fragkiadaki, K.: Particle video revisited: Tracking through occlusions using point trajectories. In: European Conference on Computer Vision. pp. 59–75. Springer (2022)
25. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022)
26. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020)
27. Henaff, O.: Data-efficient image recognition with contrastive predictive coding. In: International conference on machine learning. pp. 4182–4192. PMLR (2020)
28. Huang, Y., Yang, L., Sato, Y.: Compound prototype matching for few-shot action recognition. In: European Conference on Computer Vision. pp. 351–368. Springer (2022)
29. Karaev, N., Rocco, I., Graham, B., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker: It is better to track together. arXiv preprint arXiv:2307.07635 (2023)

30. Kliper-Gross, O., Hassner, T., Wolf, L.: One shot similarity metric learning for action recognition. In: Similarity-Based Pattern Recognition: First International Workshop, SIMBAD 2011, Venice, Italy, September 28-30, 2011. Proceedings 1. pp. 31–45. Springer (2011)
31. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T.A., Serre, T.: Hmdb: A large video database for human motion recognition. 2011 International Conference on Computer Vision pp. 2556–2563 (2011)
32. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**, 91–110 (2004)
33. Moing, G.L., Ponce, J., Schmid, C.: Dense optical tracking: Connecting the dots. arXiv preprint arXiv:2312.00786 (2023)
34. Muller, M.: Dynamic time warping in information retrieval for music and motion. Dynamic time warping Information retrieval for music and motion pp. 69–84 (2007)
35. Neoral, M., Šerỳch, J., Matas, J.: Mft: Long-term tracking of every pixel. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 6837–6847 (2024)
36. Nguyen, K.D., Tran, Q.H., Nguyen, K., Hua, B.S., Nguyen, R.: Inductive and transductive few-shot video classification via appearance and temporal alignments. In: European Conference on Computer Vision. pp. 471–487. Springer (2022)
37. Ni, X., Liu, Y., Wen, H., Ji, Y., Xiao, J., Yang, Y.: Multimodal prototype-enhanced network for few-shot action recognition. arXiv preprint arXiv:2212.04873 (2022)
38. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.Q., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y.B., Li, S.W., Misra, I., Rabbat, M.G., Sharma, V., Synnaeve, G., Xu, H., Jégou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision. ArXiv **abs/2304.07193** (2023)
39. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2536–2544 (2016)
40. Perrett, T., Masullo, A., Burghardt, T., Mirmehdi, M., Damen, D.: Temporal-relational crosstransformers for few-shot action recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 475–484 (2021)
41. Poppe, R.: A survey on vision-based human action recognition. Image and vision computing **28**(6), 976–990 (2010)
42. Sand, P., Teller, S.: Particle video: Long-range motion estimation using point trajectories. International journal of computer vision **80**, 72–91 (2008)
43. Shao, D., Zhao, Y., Dai, B., Lin, D.: Finegym: A hierarchical video dataset for fine-grained action understanding. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
44. Shi, J., et al.: Good features to track. In: 1994 Proceedings of IEEE conference on computer vision and pattern recognition. pp. 593–600. IEEE (1994)
45. Shi, X., Huang, Z., Bian, W., Li, D., Zhang, M., Cheung, K.C., See, S., Qin, H., Dai, J., Li, H.: Videoflow: Exploiting temporal cues for multi-frame optical flow estimation. arXiv preprint arXiv:2303.08340 (2023)
46. Soomro, K., Zamir, A., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. ArXiv **abs/1212.0402** (2012)
47. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. pp. 402–419. Springer (2020)

48. Thatipelli, A., Narayan, S., Khan, S., Anwer, R.M., Khan, F.S., Ghanem, B.: Spatio-temporal relation modeling for few-shot action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19958–19967 (2022)
49. Tomasi, C., Kanade, T.: Detection and tracking of point. Int J Comput Vis **9**(137-154), 3 (1991)
50. Tong, Z., Song, Y., Wang, J., Wang, L.: Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. Advances in neural information processing systems **35**, 10078–10093 (2022)
51. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 4489–4497 (2015)
52. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. Advances in neural information processing systems **29** (2016)
53. Wang, Q., Chang, Y.Y., Cai, R., Li, Z., Hariharan, B., Holynski, A., Snavely, N.: Tracking everything everywhere all at once. arXiv preprint arXiv:2306.05422 (2023)
54. Wang, X., Qing, Z., Huang, Z., Feng, Y., Zhang, S., Jiang, J., Tang, M., Gao, C., Sang, N.: Proposal relation network for temporal action detection. arXiv preprint arXiv:2106.11812 (2021)
55. Wang, X., Zhang, S., Qing, Z., Gao, C., Zhang, Y., Zhao, D., Sang, N.: Molo: Motion-augmented long-short contrastive learning for few-shot action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18011–18021 (2023)
56. Wang, X., Zhang, S., Qing, Z., Shao, Y., Gao, C., Sang, N.: Self-supervised learning for semi-supervised temporal action proposal. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1905–1914 (2021)
57. Wang, X., Zhang, S., Qing, Z., Tang, M., Zuo, Z., Gao, C., Jin, R., Sang, N.: Hybrid relation guided set matching for few-shot action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19948–19957 (2022)
58. Wang, X., Zhang, S., Qing, Z., Zuo, Z., Gao, C., Jin, R., Sang, N.: Hyrsm++: Hybrid relation guided temporal set matching for few-shot action recognition. Pattern Recognition **147**, 110110 (2024)
59. Wang, X., Yan, Y., Hu, H.M., Li, B., Wang, H.: Cross-modal contrastive learning network for few-shot action recognition. IEEE Transactions on Image Processing (2024)
60. Wang, X., Ye, W., Qi, Z., Zhao, X., Wang, G., Shan, Y., Wang, H.: Semantic-guided relation propagation network for few-shot action recognition. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 816–825 (2021)
61. Wu, J., Zhang, T., Zhang, Z., Wu, F., Zhang, Y.: Motion-modulated temporal fragment alignment network for few-shot action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9151–9160 (2022)
62. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3733–3742 (2018)
63. Xiao, Y., Wang, Q., Zhang, S., Xue, N., Peng, S., Shen, Y., Zhou, X.: Spatialtracker: Tracking any 2d pixels in 3d space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20406–20417 (2024)

64. Xing, J., Wang, M., Mu, B., Liu, Y.: Revisiting the spatial and temporal modeling for few-shot action recognition. In: AAAI Conference on Artificial Intelligence (2023), https://api.semanticscholar.org/CorpusID:255999953
65. Xing, J., Wang, M., Ruan, Y., Chen, B., Guo, Y., Mu, B., Dai, G., Wang, J., Liu, Y.: Boosting few-shot action recognition with graph-guided hybrid matching. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1740–1750 (2023)
66. Xu, J., Ranftl, R., Koltun, V.: Accurate optical flow via direct cost volume processing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1289–1297 (2017)
67. Zhang, F., Woodford, O.J., Prisacariu, V.A., Torr, P.H.: Separable flow: Learning motion cost volumes for optical flow estimation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10807–10817 (2021)
68. Zhang, H., Zhang, L., Qi, X., Li, H., Torr, P.H., Koniusz, P.: Few-shot action recognition with permutation-invariant attention. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16. pp. 525–542. Springer (2020)
69. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14. pp. 649–666. Springer (2016)
70. Zhang, S., Zhou, J., He, X.: Learning implicit temporal alignment for few-shot video classification. arXiv preprint arXiv:2105.04823 (2021)
71. Zheng, S., Chen, S., Jin, Q.: Few-shot action recognition with hierarchical matching and contrastive learning. In: European Conference on Computer Vision. pp. 297–313. Springer (2022)
72. Zheng, Y., Harley, A.W., Shen, B., Wetzstein, G., Guibas, L.J.: Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19855–19865 (2023)
73. Zhu, L., Yang, Y.: Compound memory networks for few-shot video classification. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 751–766 (2018)
74. Zhu, L., Yang, Y.: Label independent memory for semi-supervised few-shot video classification. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(1), 273–285 (2020)