

Multi-entity Video Transformers for Fine-Grained Video Representation Learning

Matthew Walmer^{1*}
Keyur Muzumdar²

Rose Kanjirathinkal²
Taipeng Tian²

Kai Sheng Tai²
Abhinav Shrivastava¹

¹University of Maryland, College Park

²Meta

Abstract

The area of temporally fine-grained video representation learning focuses on generating frame-by-frame representations for temporally dense tasks, such as fine-grained action phase classification and frame retrieval. In this work, we advance the state-of-the-art for self-supervised models in this area by re-examining the design of transformer architectures for video representation learning. A key aspect of our approach is the improved sharing of scene information in the temporal pipeline by representing multiple salient entities per frame. Prior works use late-fusion architectures that reduce frames to a single-dimensional vector before modeling any cross-frame dynamics. In contrast, our **Multi-entity Video Transformer (MV-Former)** processes the frames as groups of entities represented as tokens linked across time. To achieve this, we propose a *Learnable Spatial Token Pooling* strategy to identify and extract features for multiple salient regions per frame. Through our experiments, we show that MV-Former outperforms previous self-supervised methods, and also surpasses some prior works that use additional supervision or training data. When combined with additional pre-training data from Kinetics-400, MV-Former achieves a further performance boost. Overall, our MV-Former achieves state-of-the-art results on multiple fine-grained video benchmarks and shows that parsing video scenes as collections of entities can enhance performance in video tasks.

1. Introduction

Self-Supervised Learning (SSL) has been a rapidly growing area of research, showing potential both to scale to massive data and to learn in environments where annotations are limited or expensive to generate [1, 2, 7, 10, 11, 23–25, 30, 53]. Video Representation Learning is a promising match for SSL [38, 39], though the majority of existing research in this space focuses on learning a single, video-level

*Work performed during an internship with Meta.

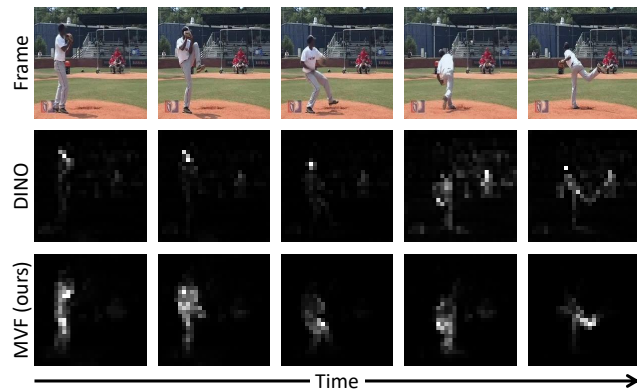


Figure 1. We present **MV-Former**, a **Multi-entity Video Transformer** architecture for fine-grained video representation learning. Our multi-entity approach naturally learns to isolate the main actor in a video using only contrastive self-supervision. As shown above, MV-Former attention has superior actor segmentation over DINO. MV-Former also achieves state of the art performance on multiple fine-grained video datasets and tasks.

representation [12, 17, 18, 34, 37, 43, 45, 48, 50]. However, there are many fine-grained video tasks which require not only video-level understanding, but also a temporally dense understanding too. Such tasks include action phase classification [15, 27, 42, 55], fine-grained frame retrieval [15, 21], and video temporal alignment [5, 15, 20, 21]. Methods designed for such tasks focus on generating frame-wise features that are expressive and discriminative not only for the high-level actions in a video, but also the subtle moment-to-moment steps of those actions [9, 15, 20, 21, 31, 40, 54]. SSL is especially desirable for this domain, as frame-level video annotations are rare and expensive to create.

Networks for fine-grained video learning typically use a two-stage approach. First, a frame-level encoder backbone is applied to reduce each frame into a single vector. Second, a temporal fusion module is applied, which allows information to flow between the representations of separate frames. Early works in this field utilized 3D convolutional networks to perform temporal fusion [8, 16, 44, 46], but recent works

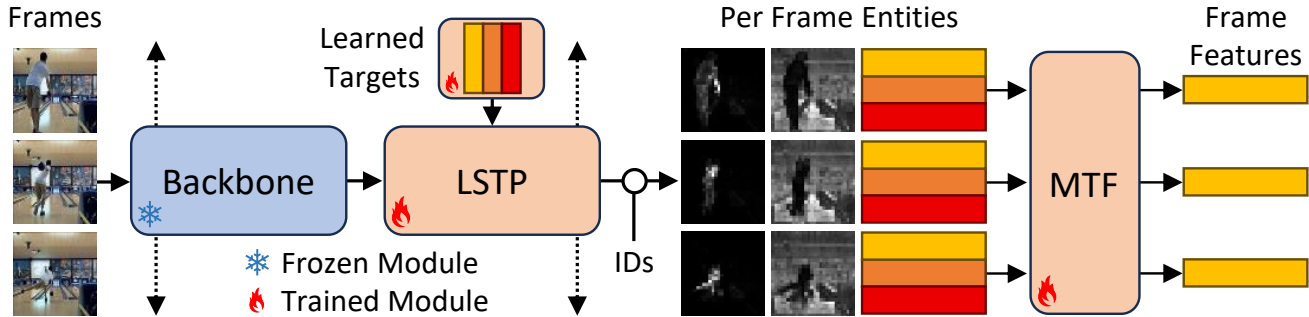


Figure 2. Overview of our **MV-Former** architecture. Multi-layer features are extracted from a frozen backbone and fed into our **Learnable Spatial Token Pooling (LSTP)** module which uses learned embedding targets to extract multiple entities per frame. Then our **Multi-entity Temporal Fusion (MTF)** transformer combines these entities to produce the final per-frame features. These temporally dense features are then applied to fine-grained downstream tasks including action phase classification and fine-grained frame retrieval.

have shifted to transformer-based temporal fusion [9, 54]. Transformer fusion allows models to learn long-range inter-frame dynamics through self-attention. However, past architectures are still limited in how they share information over the temporal axis. Prior approaches reduce each frame to a single 1D vector before any inter-frame information is combined. We hypothesize that this bottleneck restricts the ability of models to represent multiple entities, which limits their capacity to learn the temporal dynamics of a scene.

In this work, we re-examine the design of transformer-based architectures for self-supervised video representation learning and propose a new **Multi-entity Video Transformer (MV-Former)** which achieves state-of-the-art performance on multiple fine-grained video tasks. Central to our approach for MV-Former is the choice to not parse videos as collections of frames, but instead as collections of salient entities, such as the primary actor and the scene background. We call this method **Multi-entity Temporal Fusion (MTF)**. Our approach is based on the intuition that videos contain multiple elements with distinct temporal dynamics. For example, the main human actor in an action recognition video may move rapidly, while the background moves very little. We extract multiple entities per frame with consistent semantics using a **Learnable Spatial Token Pooling (LSTP)** strategy. MV-Former is able to effectively identify the primary actor in a scene using only contrastive self-supervision. As shown in Figure 1, MV-Former’s entity attention maps better segment the main actor than the best DINO CLS attention head. Overall, our MV-Former architecture advances the state-of-the-art for fine-grained tasks on the Penn Action [40], Pouring [40], and FineGym [41] datasets, and we demonstrate further improvements when combined with large-scale pre-training on Kinetics-400 [8]. In summary, our contributions are as follows:

- **MV-Former**, a Multi-entity Video Transformer architecture designed for self-supervised fine-grained video representation learning.

- **Multi-entity Temporal Fusion**, a video representation learning approach which fuses information across time by first parsing a scene into a collection of salient entities.
- A **Learnable Spatial Token Pooling** strategy to identify and extract multiple salient entities per frame with consistent semantics across time from learned targets.
- Experiments with MV-Former achieving state-of-the-art results on several fine-grained video benchmarks.

2. Related Work

Self-Supervised Visual Learning. Significant advances have been made in recent years in learning self-supervised visual representations, with many works focused on the recently popularized Vision Transformer (ViT) architecture [13]. As of writing, the most popular methods are contrastive or joint-embedding-based methods [7, 10, 19, 23, 33], and masked-reconstruction methods [3, 24]. Many works have been proposed that translate advances in image-level Self-Supervised Learning to the video domain. [17] demonstrated a unified framework to extend methods like MoCo [23], BYOL [19], SimCLR [10], and SwAV [6] into video-level methods by learning to maximize the similarity of representations of different clips from the same video. Several works focus on applying masked-reconstruction-based methods to video through strategies like temporal masking and reconstruction [18, 43, 45, 49, 50]. Other works focus on learning temporally stable representations by combining wide and narrow views of time [12, 16, 26, 35–37, 48, 51]. Note that these methods usually focus on learning video-level representations that are invariant to changes over time. While such representations are beneficial for high-level understanding, they are not helpful for tasks that require fine-grained, frame-level understanding.

Fine-Grained Video Representation Learning. Works in this domain of research aim to generate dense, frame-level features to enable video tasks such as fine-grained

frame retrieval [15, 21], per-frame action classification [15, 27, 42, 55], and video alignment [5, 15, 20, 21]. Much of the research in this field focuses on either self-supervised or weakly-supervised learning methods to reduce or remove the need for costly temporally dense annotations. [40] proposed Time-Contrastive Networks (TCN) and established the Pouring dataset, while [15] created Temporal Cycle-Consistency (TCC) learning and also created additional annotations and benchmarking procedures for the Pouring and Penn Action [55] datasets. [9] improved self-supervised performance by proposing Sequence Contrastive Loss (SCL) and also by applying transformer-based temporal fusion. [54] further proposes a statistically-motivated learning objective using Brownian Bridges. [9] and [54] use similar video learning architectures, which use a pre-trained frame-level encoder which greatly compresses frame entity information before sharing it with a video-level temporal fusion transformer. In this work, we revisit the design of video learning architectures, and present a new Multi-entity Temporal Fusion architecture with MV-Former.

3. MV-Former Architecture

3.1. Design Motivation

We propose a **Multi-entity Video Transformer** architecture (**MV-Former**), which is designed to parse video scenes not as individual frames, but instead as collections of entities. In this work, we use the term “entity” to describe any region of a video frame with a shared and consistent semantic meaning for purposes of scene parsing. Under this definition, an entity can describe a person, an object, or even the entire background of the scene. Though the background may be composed of many separate objects (floors, walls, furniture, *etc.*) for purposes of parsing the video it can often be grouped together as a single entity. In our MV-Former architecture, we extract multiple entities per frame and associate them across time through a shared ID-vector. Our motivation for this approach is based on the intuition that video scenes contain multiple entities with distinct temporal dynamics. For example, in a video of a human performing an action, the person’s pose is often highly dynamic, while the scene background is largely static. MV-Former can be separated into three main components: the per-frame Visual Backbone, the Learnable Spatial Token Pooling module, and the Multi-entity Temporal Fusion module. We show these components in Figure 2 and discuss them in more detail in the following sections.

3.2. Backbone and Frame-Level Features

Like prior works, we start by applying a pre-trained backbone on a frame-by-frame basis. However, unlike prior works, our goal for this backbone is not to extract a single compressed frame feature but instead to extract features

for multiple salient entities. For this reason, we select a DINO (v1) ViT backbone [7], and specifically use DINO ViT-B/8, as its larger size and smaller patch resolution provide finer detail for local object features. ViTs produce powerful representations both at a global level and at a local level in the form of spatial token features. Prior works have demonstrated that these local features align well with object boundaries and semantics [7], so they are well-suited for extracting features for multiple entities per frame. It has also been shown that the alignment of DINO feature semantics with local objects and object parts varies depending on their depth in the network [47]. For this reason, we use a multi-layer feature extraction strategy and take spatial token features from multiple intermediate layers. In addition, in Section 5.5 we present an ablation demonstrating that LSTP and MTF are also beneficial for a ResNet [22] backbone.

3.3. Learnable Spatial Token Pooling

After the frame-level backbone, MV-Former must identify and extract multiple entities from the spatial token features. Cross-attention is a desirable mechanism for this purpose, as it can flexibly and dynamically extract features from regions of different shapes and sizes. We draw inspiration from [32] which uses the text encoder of a CLIP model to guide self-supervised segmentation through cross-attention on the visual token features. As we are working with DINO, a vision-only model, there are no language encoder features to guide this cross-attention. Instead, we propose a **Learnable Spatial Token Pooling (LSTP)**, which uses learnable embedding vectors for the cross-attention input. These embedding vectors are trained as parameters along with the rest of the network, and they allow the network to learn which features are worth extracting from the scene. Our approach has some similarity with Q-Former [28], as both use cross-attention with learnable queries to extract information from a ViT backbone. However, while Q-Former is designed to bridge vision and language representations, our LSTP instead is created to extract salient per-frame visual entities for temporal fusion in video. The number of learnable embedding vectors determines the number of entities extracted per frame. In our primary results, we use 3 or 6 entities per frame depending on the dataset. We also present an ablation in Section 5.5 comparing different entity counts. These learnable embedding targets are held constant along the temporal dimension, which enables the LSTP module to extract consistent features for the most salient objects and image regions. In practice, LSTP naturally learns to segment out the primary actor in the scene and the scene background, as we show in Section 5.4.

3.4. Multi-Entity Temporal Fusion

Finally, we fuse the per-frame per-entity features across the temporal dimension using a **Multi-entity Temporal Fusion**

Table 1. Experimental results for three fine-grained video datasets. We achieve state-of-the-art self-supervised results on all metrics for Penn Action and FineGym, and in fine-grained frame retrieval for Pouring. For details on tasks and metrics, please see Section 4.

Method	Penn Action				Pouring				FineGym	
	Class.	Progress	Tau	Retrieval	Class.	Progress	Tau	Retrieval	Class. (99)	Class. (288)
CARL [9]	93.07	0.918	0.985	92.28	93.73	0.935	0.992	-	41.75	35.23
VSP [54]	93.12	0.923	0.986	92.56	93.85	0.942	0.990	91.85	43.12	36.95
MVF (ours)	94.21 \pm 0.04	0.931 \pm 0.006	0.989 \pm 0.002	92.99 \pm 0.06	93.34 \pm 2.00	0.919 \pm 0.015	0.985 \pm 0.003	92.33 \pm 1.02	44.77 \pm 0.71	38.30 \pm 0.26

(MTF) module. The purpose of this module is to generate dense, per-frame features that are enriched through temporal context. Like [9], we use a three block transformer to perform fusion of the frame-level features. However, rather than feed in a single token per frame, we input multiple tokens per frame to represent the multiple entities extracted through LSTP. To differentiate the entities, we append a one-hot ID vector to the end of each entity feature vector during token generation. For a standard transformer architecture, the number of input tokens will be equal to the number of output tokens, meaning a separate feature is generated per input entity. To reduce this to a single output feature per frame, we consider two options. The first is to simply take the average of all the per-frame tokens. The second is a “CLS-token-style” approach, where we designate one token of each frame to act as the output token. This approach provides greater flexibility than average pooling, which tends to provide similar gradients to each of the separate entity tokens. We find that the final pooling choice may favor some metrics over others. Specifically, we find that the averaging approach performs better for the Classification and Retrieval metrics, while the CLS-style approach works better for Phase Progression. Also note that under our multi-entity approach, the effective “width” of the transformer is multiplied by the number of entities, however, this does not increase the number of parameters. To provide a uniform baseline of comparison, we also present a “fixed-width” baseline in Section 5.5 which simulates the increased width of our MTF module.

4. Experimental Methods

Datasets. We follow the protocols of [15] and [9] and conduct benchmarking experiments on three video datasets: Penn Action [55], Pouring [40], and FineGym [41]. For FineGym, we conduct experiments on two different splits, FineGym99 and FineGym288. FineGym288 has additional category labels and training data. All of our experiments are trained self-supervised without labels, so for FineGym288 we utilize the extra training data but not the extra label information. Additionally, we conduct further experiments with large-scale pre-training on Kinetics-400 [8].

Tasks and Metrics. We assess MV-Former using four standard tasks and metrics. **(1) Fine-Grained Phase Classification:** In this task, videos are annotated on a frame-by-frame level by dividing the actions into key phases. After self-supervised pre-training, the model is frozen and a linear classifier is trained to predict the action phases. Classification accuracy is reported. **(2) Phase Progression:** For each frame, the model must predict how much time is left until the next action phase boundary. A linear regression model is trained on top of the frozen network, and the average R-squared metric is reported. **(3) Kendall’s Tau:** Given two pairs of frames from two different videos, the model must match the frames such that the pairs have the same temporal ordering. This is achieved through nearest neighbors matching. Fraction of correct matches is reported. **(4) Fine-Grained Frame Retrieval:** Given a query frame, return k frames with the same fine-grained frame action label as the query. We report results for Average Precision with $k = 5$ (AP@5). For additional details, please see [9, 15].

Baselines. We compare MV-Former with the most recent state-of-the-art methods for fine-grained self-supervised video representation learning, which are CARL [9] and VSP [54]. Over the years, many methods have been proposed that use different levels of weakly supervised data, including paired video samples [4, 15, 21, 29], or phase boundary time stamps [54]. We present a comprehensive comparison with additional self-, weakly-, and fully-supervised methods in Section 5.3. We also note that some works focus on learning video representations for human action recognition on pre-extracted sequences of human skeleton joint positions in place of or in addition to RGB frames [14, 52]. Such works show promising results, but they are also reliant on joint extraction models that are trained with extensive annotations. In this work, we focus on comparing with video representation learning methods that operate in RGB space.

Training Details. For our frame-level backbone, we utilize DINO ViT-B/8 [7]. We find that some datasets benefit from features derived from earlier layers, while others prefer features derived from only later layers. We extract features from layers 4, 8, and 12 for Penn Action and Kinetics-400, and from layers 10, 11, and 12 for FineGym. For Pouring,

Table 2. Results for our MV-Former and baselines on Penn Action with Kinetics-400 pre-training. MV-Former achieves state-of-the-art performance in Classification and Retrieval. The highest score for each metric is **bold** and the second highest is underlined.

Method	Pre.	Fine.	Class.	Progress	Tau	Retrieval
CARL	w/o Pre.		93.07	0.918	0.985	92.28
CARL	✓		91.9	0.903	0.949	-
CARL	✓	✓	93.9	0.908	0.977	-
VSP	w/o Pre.		93.12	0.923	0.986	92.56
VSP	✓		92.35	0.894	0.952	-
VSP	✓	✓	93.57	0.944	0.988	-
MVF	w/o Pre.		94.21 \pm 0.04	0.931 \pm 0.006	0.989 \pm 0.002	92.99 \pm 0.06
MVF	✓		91.62 \pm 0.30	0.927 \pm 0.005	0.895 \pm 0.004	88.38 \pm 0.20
MVF	✓	✓	94.56 \pm 0.32	0.924 \pm 0.004	0.980 \pm 0.002	93.40 \pm 0.08

we use only the final layer features, as using extra features is detrimental due to the very small size of the dataset. For Penn Action, Pouring, and Kinetics-400, we use 3 entities per frame, and for FineGym we increase this to 6 entities per frame. We use a CLS-style final feature selection strategy for Penn Action, Pouring, and Kinetics-400, and we use the average-pooling strategy for FineGym classification on both splits. We train MV-Former using Sequence Contrastive Loss (SCL) [9]. For Pouring, Penn Action, FineGym99/288, and Kinetics-400, we train for 1000, 500, 300, and 10 epochs respectively. On Penn Action and Pouring, we train with batch size 4 on 4 A100 GPUs, and for FineGym and Kinetics-400 pre-training we use batch size 8 on 8 A100 GPUs.

Codebase. Our codebase is built on the Contrastive Action Representation Learning framework [9], and can be found at: https://github.com/facebookresearch/video_rep_learning. We have maintained exact experiment configuration files tracking all hyperparameters to enable full reproducibility of all results presented.

Measuring the Impact of Initialization. For our evaluation protocols, we make one major change from prior works, as we choose to measure and report results for multiple random initializations per model configuration and dataset. For any optimization procedure, the final state of the network will depend on the initial state, but a well-performing network and objective together should converge to a good solution consistently, regardless of the initial state. We believe it is important to consider the impact of the random network initialization on the quality of the final network. While prior works report results for only one trial per model, we instead adopt a multi-trial protocol. Specifically, in each test we conduct three trials with different random initialization seeds, and we report all results as the mean plus/minus two standard deviations. Through this protocol, we show that MV-Former achieves state-of-the-art performance on Penn Action and FineGym by a statistically sig-

Table 3. Ablation study of MV-Former design details on Penn Action. “FWB” is a Fixed-Width Baseline which emulates the extra width of MTF without LSTP. “Last” uses only last-layer features instead of multi-layer features. “Ent.” denotes the entity count.

Feats.	Ent.	Class.	Progress	Tau	Retrieval
FWB	3*	88.28 \pm 0.48	0.913 \pm 0.008	0.995 \pm 0.0002	86.49 \pm 0.27
FWB	5*	87.97 \pm 0.12	0.911 \pm 0.004	0.995 \pm 0.0001	86.26 \pm 0.12
Last	1	93.35 \pm 0.61	0.924 \pm 0.006	0.990 \pm 0.0001	91.97 \pm 0.79
Last	3	93.92 \pm 0.24	0.926 \pm 0.004	0.989 \pm 0.0003	92.81 \pm 0.30
Last	5	93.50 \pm 0.27	0.911 \pm 0.006	0.992 \pm 0.001	92.50 \pm 0.3
Multi.	1	94.23 \pm 0.08	0.920 \pm 0.007	0.987 \pm 0.0005	93.02 \pm 0.18
Multi.	3	94.21 \pm 0.04	0.931 \pm 0.006	0.989 \pm 0.002	92.99 \pm 0.06
Multi.	5	94.11 \pm 0.18	0.926 \pm 0.005	0.989 \pm 0.002	93.07 \pm 0.01

nificant margin. This also allows us to measure the variance of the four commonly used benchmark tasks and metrics. We find that the Phase Progression task and metric has the highest sensitivity to the model initialization. We encourage future works to adopt a similar multi-trial methodology.

5. Results

5.1. Fine-Grained Datasets

We summarize our results for MV-Former on three fine-grained video representation learning benchmarks, Penn Action, Pouring, and FineGym, with results in Table 1.

Penn Action. MV-Former achieves state-of-the-art self-supervised performance in all four metrics for Penn Action. The largest gain is achieved in Classification, with an increase of 1.09% over VSP. For all four metrics, our improvement is at least two standard deviations above the best performing prior work. We note that Phase Progression has the highest standard deviation relative to its absolute value, showing that it is the most sensitive to the random network initialization. Meanwhile, Classification and Retrieval both have relatively low sensitivity to initialization.

Pouring. For Pouring we do not see consistent improvement for all metrics, and in addition we see much higher variance across trials for all four metrics as compared to Penn Action. We attribute both of these issues to the small size of the Pouring training set, which contains only 70 videos, making it roughly 16 times smaller than Penn Action and 45 times smaller than FineGym99. However, MV-Former does achieve state-of-the-art results for Fine-Grained Frame Retrieval on this benchmark.

FineGym. Like prior works, we report fine-grained classification accuracy on two splits, FineGym99 and FineGym288. MV-Former achieves state-of-the-art self-supervised results for Phase Classification on both splits. MV-Former’s average score surpasses VSP by 1.65% for FineGym99 and 1.35% for FineGym288. These results surpass the prior bests by at least two standard deviations.

Table 4. Comprehensive comparison with methods that use self-, weakly-, or fully-supervised learning. Our MV-Former is only surpassed by other methods that use additional supervision or data. The highest score for each metric is **bold** and the second highest is underlined.

Method	Labels	Pretrain	Penn Action				FineGym	
			Class.	Progress	Tau	Retrieval	Class. (99)	Class. (288)
TCC* [15]	Video	-	81.35	0.664	0.701	76.74	25.18	20.82
TCC [15]	Video	-	74.39	0.591	0.641	-	-	-
TCC [†] [9, 15]	Video	-	86.35	0.899	0.980	-	-	-
LAV* [21]	Video	-	84.25	0.661	0.805	79.13	-	-
LAV [21]	Video	-	78.68	0.625	0.684	-	-	-
GTA [20]	Video	-	-	0.789	0.748	-	-	-
VAVA [29]	Video	-	84.48	0.709	0.805	-	-	-
LRProp [4]	Video	-	93.25	0.930	0.991	92.46	-	-
SaL [31]	None	-	68.15	0.390	0.474	76.04	21.45	19.58
TCN [40]	None	-	68.09	0.383	0.542	77.84	20.02	17.11
TCN [†] [9, 40]	None	-	86.31	0.898	0.832	-	-	-
CARL [9]	None	-	93.07	0.918	0.985	92.28	41.75	35.23
VSP [54]	None	-	93.12	0.923	0.986	92.56	43.12	36.95
VSP-P [54]	Phase	-	93.27	-	-	<u>93.45</u>	44.58	38.23
VSP-F [54]	Frame	-	<u>94.24</u>	-	-	94.89	45.66	39.48
CARL [9]	None	K-400	93.9	0.908	0.977	-	-	-
VSP [54]	None	K-400	93.57	0.944	0.988	-	-	-
MVF (ours)	None	-	94.21 \pm 0.04	0.931 \pm 0.006	0.989 \pm 0.002	92.99 \pm 0.06	44.77 \pm 0.71	38.30 \pm 0.26
MVF (ours)	None	K-400	94.56 \pm 0.32	0.924 \pm 0.004	0.980 \pm 0.002	93.40 \pm 0.08	-	-

5.2. Large-Scale Pre-training

We show that MV-Former can achieve further performance improvements on the Penn Action dataset through large-scale pre-training on Kinetics-400. We again follow our 3-trial experimental protocol and pre-train three different MV-Formers on Kinetics-400 for 10 epochs, and then fine-tune them on Penn Action for 500 epochs. The results are summarized in Table 2. We see that Kinetics-400 pre-training boosts Classification performance by another 0.35%. We also see a 0.41% gain in Retrieval over our base MV-Former. We find that Kinetics-400 pre-training is not beneficial for Progress and Tau, which matches previous trends observed by [9] for CARL.

5.3. Methods using Additional Supervision

In Table 4 we present a comprehensive, unified comparison of recent approaches on the Penn Action and FineGym datasets for fine-grained video representation tasks. In addition to self-supervised methods, we also compare with methods using weak or full supervision, and for Penn Action we also include methods using Kinetics-400 pre-training. “Video labels” means the method requires a video-level label of the action category in order to sample video pairs of the same category. “Phase labels” means the method uses annotations for the positions of the action phase boundaries in time, but not the action phase categories. “Frame labels” means the method uses fully labelled data with phase boundaries and phase category labels. Methods TCC* and LAV* denote versions where a

separate model is trained for each Penn Action category. TCC[†] and TCN[†] denote reimplemented versions presented by [9] which utilize ViT-based temporal fusion architectures. MV-Former with Kinetics-400 pre-training achieves the overall highest performance in Penn Action Phase Classification, surpassing even the fully-supervised version of VSP (VSP-F). For FineGym classification, MV-Former is second only to the fully-supervised VSP-F. Overall, MV-Former is only surpassed by methods that use additional supervision or training data.

5.4. Visualizing Learnable Spatial Token Pooling

We visualize the attention maps created by our LSTP module, which correspond to the regions that contribute to each entity’s features. We take several sample videos from the Penn Action dataset and visualize the LSTP attention maps for our best performing MV-Former model in Figures 1 and 3. In addition, we present a comparison of our LSTP attention maps with DINO ViT-B/8’s CLS token attention maps. For sake of fair comparison, we manually select the DINO head that is most aligned with the scene actor. We find that our first entity’s attention maps (denoted MVF-E1) are very well-aligned with the primary scene actor throughout each video. Meanwhile, the DINO attention maps also include the actor, but they are typically far noisier and include large portions of the background. In the right example of Figure 3, DINO’s attention maps are distracted by the large crowd in the background, while MVF-E1 clearly focuses on the athlete. Under our CLS-style output configuration, the first

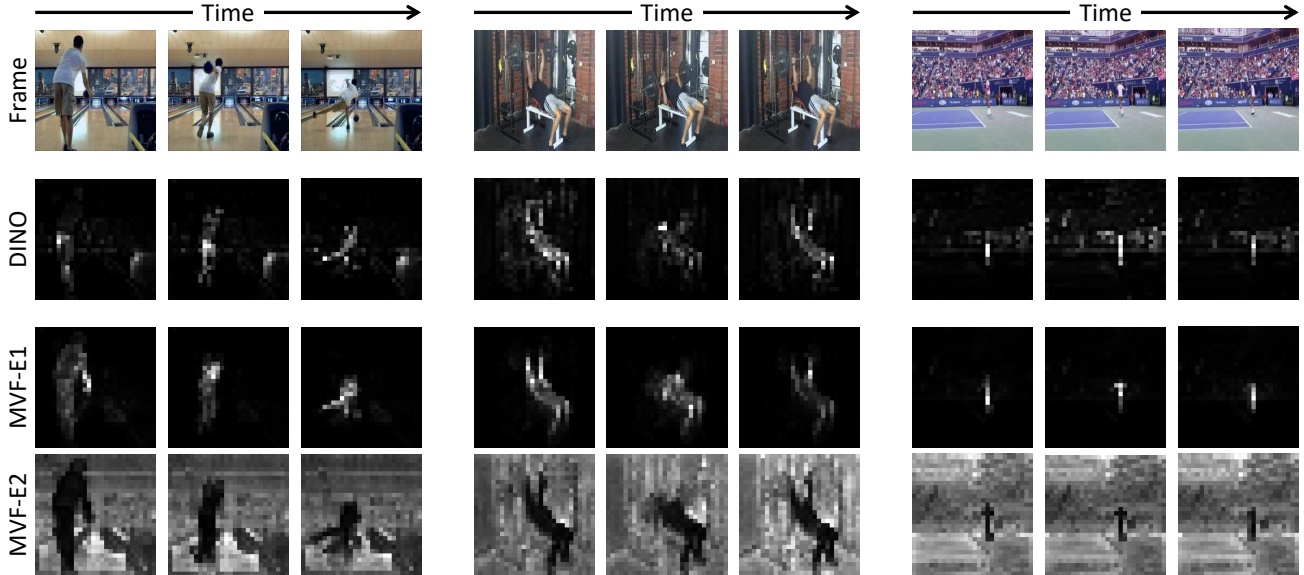


Figure 3. Visualizations of the attention maps of our LSTP module. Our first entity (MVF-E1) learns to attend to the primary actor, with a particular focus on the limbs. Our second entity (MVF-E2) focuses on the scene background. We compare with the most actor-aligned CLS attention head of DINO ViT-B/8. Our entity-learning approach better isolates and segments the primary actor in the scene over DINO.

entity is connected to the final output token, so it is intuitive that it learns to focus on the most important part of the scene. Additionally, in Figure 3 we also show examples of entity 2’s attention maps, which effectively segment out the scene background. Note that the third entity (not shown) also attends to the primary actor in the scene. Our LSTP implementation includes no restrictions to enforce that entities be disjoint, though this could be an area of future research. These visualizations show the strength of LSTP for isolating the primary actor and background, improving over DINO’s segmentation capabilities.

5.5. Ablations

We present ablations for our backbone and feature pooling methodologies on Penn Action. We can ablate MTF while still using LSTP by extracting only one entity per frame, denoted as “LSTP-1”. When using LSTP with MTF, the number of entities is 3. We also present ablations for multi-layer features, entity count, and a fixed-width baseline. We primarily focus on the Classification, Progress, and Retrieval metrics, as Tau shows only small variations in performance across configurations. Overall, the scores for Tau are quite close to 1.0, so we believe minor variations in the Kendall’s Tau score are not majorly representative of model quality.

LSTP and MTF with ResNet-50. We demonstrate that LSTP and MTF are beneficial for a ResNet-50 backbone, as shown in Table 5. For these experiments, we keep the ResNet backbone frozen. In row 1, we present results for a basic ResNet-50 backbone with the same temporal-fusion module as [9]. In row 2, the use of LSTP-1 with one entity

improves performance in Classification, Progress, and Retrieval, and leaves Tau roughly the same. In row 3, utilizing LSTP with three entities and MTF improves Progress, but slightly degrades Classification and Retrieval. Despite this, ResNet with LSTP and MTF still outperforms the baseline ResNet architecture in all metrics except Tau.

DINO Feature Extraction Methods. We test how our complete pipeline compares with a simpler model that uses one DINO feature token per frame with the same fusion transformer as [9]. We test three traditional methods for DINO feature selection: max pooling, average pooling, and the CLS token. For fair comparison, our methods use only final layer features for this analysis. The results are summarized in Table 6. Among the traditional methods, max pooling gives the lowest performance in most metrics, while the CLS token tends to be the best. We again compare with LSTP-1, which is functionally similar to the CLS token, though with the disadvantage of needing to be trained from scratch. LSTP-1 performs similarly to CLS, though it is slightly better in Progress and slightly worse in Classification and Retrieval. Finally, we compare with our full method, using LSTP and MTF with three entities per frame. Our proposed method provides a clear performance boost over DINO features alone, with the highest scores in Classification, Progress, and Retrieval.

Fixed Width Baseline. We propose a Fixed-Width Baseline (FWB) in order to demonstrate that the improved performance of MTF is not simply a consequence of its increased width. By processing 3 entities per frame, we ef-

Table 5. Ablation study applying LSTP and MTF to a ResNet-50 backbone instead of a DINO ViT backbone. Results presented on Penn Action. “LSTP-1” uses LSTP with only 1 learned entity, and thus does not use Multi-entity Temporal Fusion. “LSTP+MTF” uses 3 learned entities in this experiment. We find that our proposed methods are also beneficial for a ResNet backbone.

Method	Class.	Progress	Tau	Retrieval
RN50	91.96 \pm 0.48	0.900 \pm 0.008	0.993 \pm 0.001	90.88 \pm 0.38
RN50 + LSTP-1	92.77 \pm 0.08	0.909 \pm 0.008	0.991 \pm 0.001	91.50 \pm 0.23
RN50 + LSTP + MTF	92.46 \pm 0.24	0.915 \pm 0.002	0.992 \pm 0.0004	91.15 \pm 0.20

fectively make the transformer fusion module 3 times wider in terms of the number of tokens, however, this does not change the number of learnable parameters at all. To create a fixed-width baseline, we simulate the increased width of MTF by splitting the CLS token of DINO into three separate tokens using a linear layer. The new tokens are fed into the temporal fusion transformer just as they would be for MTF, which gives an architecture of equivalent computational width to our proposed method. We conduct this experiment for 3 or 5 tokens per frame. The results are shown in Table 3 rows 1 and 2. We find that this simple strategy for increasing the width of the fusion transformer does not improve performance, and actually hinders it. This demonstrates that MTF is not simply benefiting from increased width, and also shows that LSTP is an effective strategy to extract multiple useful entities per frame.

Entity Count. We experiment with our selection of entities extracted per frame for LSTP and MTF. We test configurations with 1, 3, and 5 entities. We present this ablation both with and without the use of multi-layer features, which we discuss more in the following section. These results are shown in Table 3, with rows 3-5 using last-layer features and rows 6-8 using multi-layer features. To summarize, we find that using 3 entities per frame gives the best all around performance, with a balance of high scores between the four metrics. This is the case both when using multi-layer features and when using last-layer features alone. Increasing the number of entities to 5 or decreasing to 1 can sometimes be slightly beneficial, but is usually detrimental.

Multi-Layer Features. We ablate our use of multi-layer DINO features, as shown by rows 3-8 of Table 3. We either extract features from the final layer only (Last) or from the ends of blocks 4, 8, and 12 (Multi.). We find that the extra features are generally beneficial for all metrics excluding Tau. This holds for all entity counts tested. While minor differences in the model settings may slightly favor some metrics over others, we find that using 3 entities with multi-layer features is the best all-around performer.

Table 6. Ablation comparing standard DINO feature aggregation methods (MAX-pool, AVG-pool, CLS token) against our proposed LSTP and MTF approaches on the Penn Action dataset. “LSTP-1” uses only one learnable entity and does not use MTF. “LSTP+MTF” uses 3 entities for the best all-around performance. For fair comparison, all methods use only last-layer features.

Method	Class.	Progress	Tau	Retrieval
DINO + MAX	87.61 \pm 0.12	0.883 \pm 0.005	0.996 \pm 0.0001	83.76 \pm 0.21
DINO + AVG	92.57 \pm 0.07	0.904 \pm 0.012	0.995 \pm 0.001	90.52 \pm 0.4
DINO + CLS	93.48 \pm 0.16	0.918 \pm 0.007	0.992 \pm 0.0002	92.31 \pm 0.16
DINO + LSTP-1	93.35 \pm 0.61	0.924 \pm 0.006	0.990 \pm 0.0001	91.97 \pm 0.79
DINO + LSTP + MTF	93.92 \pm 0.24	0.926 \pm 0.004	0.989 \pm 0.0003	92.81 \pm 0.30

6. Discussion & Conclusion

Limitations & Future Work. Our MV-Former is effective at isolating the main actor and background. However, it does not consistently isolate salient task-specific objects, like the weights in a weightlifting scene. In the future, we would like to improve this capacity by applying language-based weak supervision. In addition, future work could explore how the MV-Former architecture performs when combined with other video-based self-supervised learning objectives, like TCC, TCN, and VSP. Finally, as discussed in Section 3.4, MTF introduces additional tokens in the fusion transformer, which does not increase the parameter count but does increase the compute cost. Future work could potentially address this by performing token merging along the temporal dimension, exploiting the high redundancy of information between frames to reduce the token count.

Conclusion. In this work, we have presented MV-Former, a Multi-entity Video Transformer designed for fine-grained video representation learning. MV-Former parses scenes into multiple salient entities before performing transformer-based temporal fusion. We refer to this approach as Multi-entity Temporal Fusion, which is distinct from prior works where each frame must be reduced to a single vector representation before any information is shared between frames. To generate these entities, we have proposed a cross-attention-based Learnable Spatial Token Pooling strategy, which uses trainable query embeddings to learn which information to extract from each frame. We have also shown that this approach naturally learns to separate out the primary actor in each scene and the background with only contrastive self-supervision. MV-Former achieves state-of-the-art results in multiple fine-grained video tasks on the Penn Action, Pouring, and FineGym datasets. We also show how MV-Former benefits from large-scale pre-training on Kinetics-400, further advancing performance in classification and retrieval. Overall, our result with MV-Former show that parsing videos as collections of entities can improve fine-grained video representation learning.

Acknowledgements. We would like to thank our peers Chao-Yuan Wu, Sumedha Singla, and Florian Metze for their valuable feedback and suggestions for this work.

References

- [1] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32, 2019. 1
- [2] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, et al. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*, 2023. 1
- [3] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 2
- [4] Guy Bar-Shalom, George Leifman, and Michael Elad. Weakly-supervised representation learning for video alignment and analysis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6909–6919, 2024. 4, 6
- [5] Kaidi Cao, Jingwei Ji, Zhangjie Cao, Chien-Yi Chang, and Juan Carlos Niebles. Few-shot video classification via temporal alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10618–10627, 2020. 1, 3
- [6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020. 2
- [7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 1, 2, 3, 4
- [8] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 1, 2, 4
- [9] Minghao Chen, Fangyun Wei, Chong Li, and Deng Cai. Frame-wise action representations for long videos via sequence contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13801–13810, 2022. 1, 2, 3, 4, 5, 6, 7
- [10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 1, 2
- [11] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 1
- [12] Ishan Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. Tclr: Temporal contrastive learning for video representation. *Computer Vision and Image Understanding*, 219:103406, 2022. 1, 2
- [13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2
- [14] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2969–2978, 2022. 4
- [15] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1801–1810, 2019. 1, 3, 4, 6
- [16] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 1, 2
- [17] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3299–3309, 2021. 1, 2
- [18] Christoph Feichtenhofer, Yanghao Li, Kaiming He, et al. Masked autoencoders as spatiotemporal learners. *Advances in neural information processing systems*, 35:35946–35958, 2022. 1, 2
- [19] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent—a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 2
- [20] Isma Hadji, Konstantinos G Derpanis, and Allan D Jepson. Representation learning via global temporal alignment and cycle-consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11068–11077, 2021. 1, 3, 6
- [21] Sanjay Haresh, Sateesh Kumar, Huseyin Coskun, Shahram N Syed, Andrey Konin, Zeeshan Zia, and Quoc-Huy Tran. Learning by aligning videos in time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5548–5558, 2021. 1, 3, 4, 6
- [22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3
- [23] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 1, 2
- [24] 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*, pages 16000–16009, 2022. 2

- [25] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018. [1](#)
- [26] Haofei Kuang, Yi Zhu, Zhi Zhang, Xinyu Li, Joseph Tighe, Sören Schwertfeger, Cyrill Stachniss, and Mu Li. Video contrastive learning with global context. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3195–3204, 2021. [2](#)
- [27] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 780–787, 2014. [1](#), [3](#)
- [28] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. [3](#)
- [29] Weizhe Liu, Bugra Tekin, Huseyin Coskun, Vibhav Vineet, Pascal Fua, and Marc Pollefeys. Learning to align sequential actions in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2181–2191, 2022. [4](#), [6](#)
- [30] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6707–6717, 2020. [1](#)
- [31] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 527–544. Springer, 2016. [1](#), [6](#)
- [32] Jishnu Mukhoti, Tsung-Yu Lin, Omid Poursaeed, Rui Wang, Ashish Shah, Philip HS Torr, and Ser-Nam Lim. Open vocabulary semantic segmentation with patch aligned contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19413–19423, 2023. [3](#)
- [33] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafranec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. [2](#)
- [34] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6964–6974, 2021. [1](#)
- [35] Zhiwu Qing, Shiwei Zhang, Ziyuan Huang, Yi Xu, Xiang Wang, Mingqian Tang, Changxin Gao, Rong Jin, and Nong Sang. Learning from untrimmed videos: Self-supervised video representation learning with hierarchical consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13821–13831, 2022. [2](#)
- [36] Kanchana Ranasinghe, Muzammal Naseer, Salman Khan, Fahad Shahbaz Khan, and Michael S Ryoo. Self-supervised video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2874–2884, 2022.
- [37] Adria Recasens, Pauline Luc, Jean-Baptiste Alayrac, Luyu Wang, Florian Strub, Corentin Tallec, Mateusz Malinowski, Viorica Pătrăucean, Florent Altché, Michal Valko, et al. Broaden your views for self-supervised video learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1255–1265, 2021. [1](#), [2](#)
- [38] Madeline C Schiappa, Yogesh S Rawat, and Mubarak Shah. Self-supervised learning for videos: A survey. *ACM Computing Surveys*, 55(13s):1–37, 2023. [1](#)
- [39] Javier Selva, Anders S Johansen, Sergio Escalera, Kamal Nasrollahi, Thomas B Moeslund, and Albert Clapés. Video transformers: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. [1](#)
- [40] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1134–1141. IEEE, 2018. [1](#), [2](#), [3](#), [4](#), [6](#)
- [41] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2616–2625, 2020. [2](#), [4](#)
- [42] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 510–526. Springer, 2016. [1](#), [3](#)
- [43] Hao Tan, Jie Lei, Thomas Wolf, and Mohit Bansal. Vimpac: Video pre-training via masked token prediction and contrastive learning. *arXiv preprint arXiv:2106.11250*, 2021. [1](#), [2](#)
- [44] Graham W Taylor, Rob Fergus, Yann LeCun, and Christoph Bregler. Convolutional learning of spatio-temporal features. In *Computer Vision—ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part VI 11*, pages 140–153. Springer, 2010. [1](#)
- [45] 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. [1](#), [2](#)
- [46] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. [1](#)
- [47] Matthew Walmer, Saksham Suri, Kamal Gupta, and Abhinav Shrivastava. Teaching matters: Investigating the role

- of supervision in vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7486–7496, 2023. [3](#)
- [48] Jue Wang, Gedas Bertasius, Du Tran, and Lorenzo Torresani. Long-short temporal contrastive learning of video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14010–14020, 2022. [1](#), [2](#)
- [49] 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*, pages 14549–14560, 2023. [2](#)
- [50] Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Yu-Gang Jiang, Luowei Zhou, and Lu Yuan. Bevt: Bert pretraining of video transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14733–14743, 2022. [1](#), [2](#)
- [51] Fanyi Xiao, Kaustav Kundu, Joseph Tighe, and Davide Modolo. Hierarchical self-supervised representation learning for movie understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9727–9736, 2022. [2](#)
- [52] Hong Yan, Yang Liu, Yushen Wei, Zhen Li, Guanbin Li, and Liang Lin. Skeletonmae: graph-based masked autoencoder for skeleton sequence pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5606–5618, 2023. [4](#)
- [53] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021. [1](#)
- [54] Heng Zhang, Daqing Liu, Qi Zheng, and Bing Su. Modeling video as stochastic processes for fine-grained video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2225–2234, 2023. [1](#), [2](#), [3](#), [4](#), [6](#)
- [55] Weiyu Zhang, Menglong Zhu, and Konstantinos G Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *Proceedings of the IEEE international conference on computer vision*, pages 2248–2255, 2013. [1](#), [3](#), [4](#)