Learning by Aligning Videos in Time

Sanjay Haresh* Sateesh Kumar* Huseyin Coskun Shahram N. Syed
Andrey Konin M. Zeeshan Zia Quoc-Huy Tran

Retrocausal, Inc.
Seattle, WA
www.retrocausal.ai

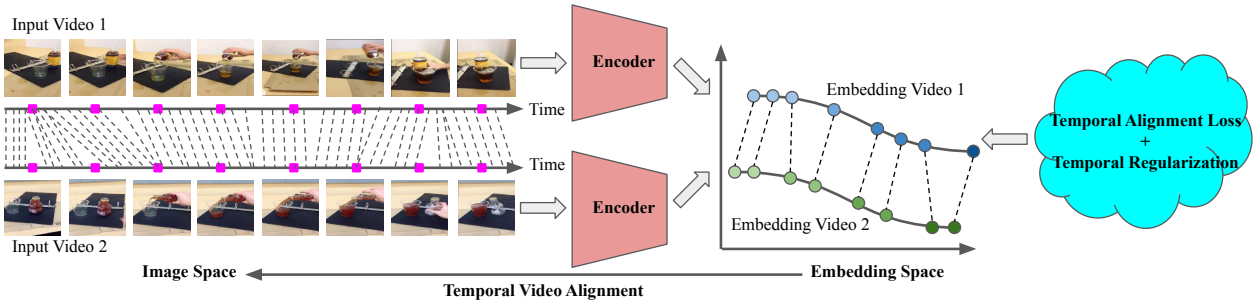


Figure 1: We propose a self-supervised method to learn video representations by aligning videos in time, despite many differences between the videos such as appearance, motion, and viewpoint. We optimize the embedding space by using both the temporal alignment loss between the videos and the temporal regularization applied separately on each video. Our learned representations can be useful for many video-based temporal understanding tasks such as temporal video alignment.

Abstract

We present a self-supervised approach for learning video representations using temporal video alignment as a pretext task, while exploiting both frame-level and video-level information. We leverage a novel combination of temporal alignment loss and temporal regularization terms, which can be used as supervision signals for training an encoder network. Specifically, the temporal alignment loss (i.e., Soft-DTW*) aims for the minimum cost for temporally aligning videos in the embedding space. However, optimizing solely for this term leads to trivial solutions, particularly, one where all frames get mapped to a small cluster in the embedding space. To overcome this problem, we propose a temporal regularization term (i.e.,* Contrastive-IDM*) which encourages different frames to be mapped to different points in the embedding space. Extensive evaluations on various tasks, including action phase classification, action phase progression, and fine-grained frame retrieval, on three datasets, namely* Pouring*,* Penn Action*, and* IKEA ASM*, show superior performance of our approach over state-of-the-art methods for self-supervised representation learning from videos. In addition, our method provides significant performance gain where labeled data is lacking.*

1. Introduction

> There are just three problems in computer vision: registration, registration, and registration.
>
> *Takeo Kanade*

Lukas-Kanade and Iterative Closest Point have been amongst the most ubiquitous building blocks in artificial perception literature. Yet spatio-temporal registration has received little attention in the present deep learning renaissance. Correspondingly, we add to a small number of recent approaches [14, 39] that have revived temporal alignment as a means of improving video representation learning. In order to learn perfect alignment of two videos, a learning algorithm must be able to disentangle phases of the activity in time while simultaneously associating visually similar frames in the two different videos. We demonstrate that learning in this manner generates representations that are effective for downstream tasks that rely on fine-grained

* indicates joint first author.
{sanjay,sateesh,huseyin,shahram,andrey,zeeshan,huy}@retrocausal.ai

temporal features.

In the context of using temporal alignment for learning video representations, some recent works [14, 39] use cycle-consistency losses to perform local alignment between individual frames. At the same time, some works have explored global alignment for video classification and segmentation [8, 5]. We adapt such global alignment ideas for video representation learning in this work.

A few of approaches have been proposed for supervised action recognition [45, 7, 49, 46] and action segmentation [15, 33]. Unfortunately, these approaches require fine-grained annotations which can be prohibitively expensive [40]. We note the seemingly infinite supply of public video data, and contrast it with the high cost of fine-grained annotation. This discrepancy emphasizes the importance of exploring self-supervised methods. We are further motivated by datasets and downstream tasks that specifically benefit from temporal alignment, such as video streams of semi-repetitive activities from manufacturing assembly lines to surgery rooms. It is desirable to measure the variability and anomalies [44, 23] across such datasets, where representations that optimize for temporal alignment may be highly performant.

Our approach, *L*earning by *A*ligning *V*ideos (*LAV*), utilizes the task of temporally aligning videos for learning self-supervised video representations. Specifically, we use a differentiable version of an alignment metric which has been widely used in the time series literature, namely Dynamic Time Warping (DTW) [4]. DTW is a global alignment metric, taking into account entire sequences while aligning. Unfortunately, in a self-supervised representation learning context, optimizing solely for DTW may converge to trivial solutions wherein the learned representations are not meaningful. To address this issue, we combine the above alignment metric with a regularization, as shown in Fig. 1. In particular, we propose a regularization term that optimizes for temporally disentangled representations, i.e., frames that are close in time are mapped to spatially nearby points in the embedding space and vice versa.

In summary, our contributions include:

- We introduce a novel self-supervised method for learning video representations by temporally aligning videos as a whole, leveraging both frame-level and video-level cues.
- We adopt the classical DTW as our temporal alignment loss, while proposing a new temporal regularization. The two components have mutual benefits, i.e., the latter prevents trivial solutions, whereas the former leads to better performance.
- Our approach performs on par with or better than the state-of-the-art on various temporal understanding tasks on *Pouring*, *Penn Action*, and *IKEA ASM* datasets. The best performance is sometimes achieved by combining our method with a recent work [14]. Further, our approach offers significant accuracy gain when lacking labeled data.
- We have made our dense per-frame labels for 2123 videos of *Penn Action* publicly available at https://bit.ly/3f73e2W.

2. Related Work

In this section, we review recent literature in self-supervised learning with a focus on image and video data.

Image-Based Self-Supervised Representation Learning. Early self-supervised representation learning methods explore image content as supervision signals. They propose pretext tasks based on artificial image cues as labels and train deep networks for solving those tasks [30, 31, 37, 34, 28, 19, 6, 16]. These pretext tasks include objectives such as image colorization [30, 31], object counting [37, 34], solving jigsaw puzzles [28, 6], and predicting image rotations [19, 16]. Even earlier approaches learn representations simply by reconstructing the input image [25] or recovering it from noise [47]. In this work, we focus on self-supervised representation learning from videos, which leverages both spatial and temporal information in videos.

Video-Based Self-Supervised Representation Learning. With the advent of deep architectures for video understanding [45, 7, 49, 46], various pretext tasks have been introduced as supervision signals for self-supervised representation learning from videos. One popular class of methods learn representations by predicting future frames [42, 48, 1, 12] or forecasting their encoding features [22, 27, 18]. Another group of methods leverage temporal information, for example, temporal order and temporal coherence are used as labels in [35, 32, 17, 53, 9] and [21, 36, 3, 56, 55, 20] respectively. Recently, Donglai et al. [50] train a deep model for classifying temporal direction, while Sermanet et al. [41] learn representations via consistency across different viewpoints and neighboring frames. The above methods usually optimize over a single video at a time, whereas our approach jointly optimizes over a pair of videos at once, potentially extracting more information from both videos.

Temporal Video Alignment. There exists a lot of literature on time series alignment, yet only a few ideas have been carried over to aligning videos. Unfortunately, traditional methods for time series alignment, for example, DTW [4], are not differentiable and hence can not be directly used for training neural networks. To address this weakness, a smooth approximation of DTW, namely Soft-DTW, is introduced in [11]. More recently, Soft-DTW formulations have been used in a weakly supervised setting for aligning a video to a transcript [8] or in a few-shot supervised setting for aligning videos [5]. In the present paper, we adapt Soft-DTW for learning self-supervised representations from videos, using temporal video alignment as

the pretext task. The closest work to ours is Temporal Cycle Consistency (TCC) [14], which learns self-supervised representations by finding frame correspondences across videos. While TCC aligns each frame separately, our approach aligns the video as a whole, leveraging both frame-level and video-level cues.

3. Our Approach

In this section, we discuss our main contribution which is a self-supervised method to learn video representations via temporal video alignment. Specifically, we learn an embedding space where two videos with similar contents can be conveniently aligned in time. We first aim to optimize the embedding space solely for the global alignment cost between the two videos, which can lead to trivial solutions. To overcome this problem, we regularize the embedding space such that for each input video, temporally close frames are mapped to nearby points in the embedding space, whereas temporally distant frames are correspondingly mapped far away in the embedding space. Fig. 2 shows an overview of our loss and regularization (right) and our encoder (left). Below we first define some notations and then provide the details of our temporal alignment loss, temporal regularization, final loss, and encoder network in Secs. 3.1, 3.2, 3.3, and 3.4 respectively.

Notations. We denote the embedding function as f_θ, namely a neural network with parameters θ. Our method takes as input two videos $X = \{x_1, x_2, \dots, x_n\}$ and $Y = \{y_1, y_2, \dots, y_m\}$, where n and m are the numbers of frames in X and Y respectively. For a frame x_i in X and y_j in Y, the embedding frames of x_i and y_j are written as $f_\theta(x_i)$ and $f_\theta(y_j)$ respectively. In addition, we denote $f_\theta(X) = \{f_\theta(x_1), f_\theta(x_2), \dots, f_\theta(x_n)\}$ and $f_\theta(Y) = \{f_\theta(y_1), f_\theta(y_2), \dots, f_\theta(y_m)\}$ as the embedding videos of X and Y respectively.

3.1. Temporal Alignment Loss

We adopt the classical DTW discrepancy [4] as our temporal alignment loss. DTW has been widely used with non-visual data, such as time series, and has just recently been applied to video data, but in a weakly supervised setup for video-to-transcript alignment [8] or in a few-shot supervised setup for video alignment [5]. Unlike [8, 5], we explore the use of DTW for self-supervised video representation learning by leveraging temporal video alignment as the pretext task.

Given two input videos X and Y and their embedding videos $f_\theta(X)$ and $f_\theta(Y)$, we can compute the distance matrix $D \in \mathbb{R}^{n\times m}$ with each entry written as $D(i,j) = ||f_\theta(x_i) - f_\theta(y_j)||^2$. DTW calculates the alignment cost between X and Y by finding the minimum cost path in D:

$$dtw(X,Y) = min_{A\in A_{n,m}}\langle A, D\rangle. \tag{1}$$

Here, $A_{n,m} \subset \{0,1\}^{n\times m}$ is the set of all possible (binary) alignment matrices, which correspond to paths from the top-left corner of D to the bottom-right corner of D using only $\{\downarrow, \rightarrow, \searrow\}$ moves. $A \in A_{n,m}$ is a typical alignment matrix, with $A(i,j) = 1$ if x_i in X is aligned with y_j in Y. DTW can be computed using dynamic programming, particularly solving the below cumulative distance function:

$$r(i,j) = D(i,j) + min\{r(i-1,j), r(i,j-1), r(i-1,j-1)\}. \tag{2}$$

Due to the non-differentiable min operator, DTW is not differentiable and unstable when used in an optimization framework. We therefore employ a continuous relaxation version of DTW, namely *Soft-DTW*, proposed by [11]. In particular, Soft-DTW replaces the discrete min operator in DTW by the smoothed min^γ one, defined as:

$$min^\gamma\{a_1, a_2, ..., a_n\} = -\gamma \log \sum_{i=1}^{n} e^{\frac{-a_i}{\gamma}}, \tag{3}$$

where $\gamma > 0$ is a smoothing parameter. Soft-DTW returns the alignment cost between X and Y by finding the soft-minimum cost path in D, which can be written as:

$$dtw^\gamma(X,Y) = min^\gamma_{A\in A_{n,m}}\langle A, D\rangle. \tag{4}$$

Note that since the smoothed min^γ operator converges to the discrete min one when γ approaches 0, Soft-DTW produces similar results as DTW when γ is near 0. In addition, although using min^γ does not make the objective convex, it does help the optimization by enabling smooth gradients and providing better optimization landscapes.

3.2. Temporal Regularization

Since (Soft-)DTW measures the (soft-)minimum cost path in D, optimizing for (Soft-)DTW alone can result in trivial solutions, wherein all the entries in D are close to 0, as we will show later in Sec. 5.1. In other words, all the frames in X and Y are mapped to a small cluster in the embedding space. To avoid that, we opt to add a temporal regularization, which is applied separately on $f_\theta(X)$ and $f_\theta(Y)$. Below we discuss our regularization for $f_\theta(X)$ only, while the same one can be applied for $f_\theta(Y)$.

Motivated by [43], we adapt Inverse Difference Moment (IDM) [10] as our regularization, which can be written as:

$$I(X) = \sum_{i=1}^{n}\sum_{j=1}^{n} W(i,j) S_X(i,j) \tag{5}$$

$$W(i,j) = \frac{1}{(i-j)^2 + 1},$$

where $S_X \in \mathbb{R}^{n\times n}$ is the self-similarity matrix of $f_\theta(X)$. Maximizing Eq. 5 encourages temporally close frames in X (with large $W(i,j)$) to be mapped to nearby points in the

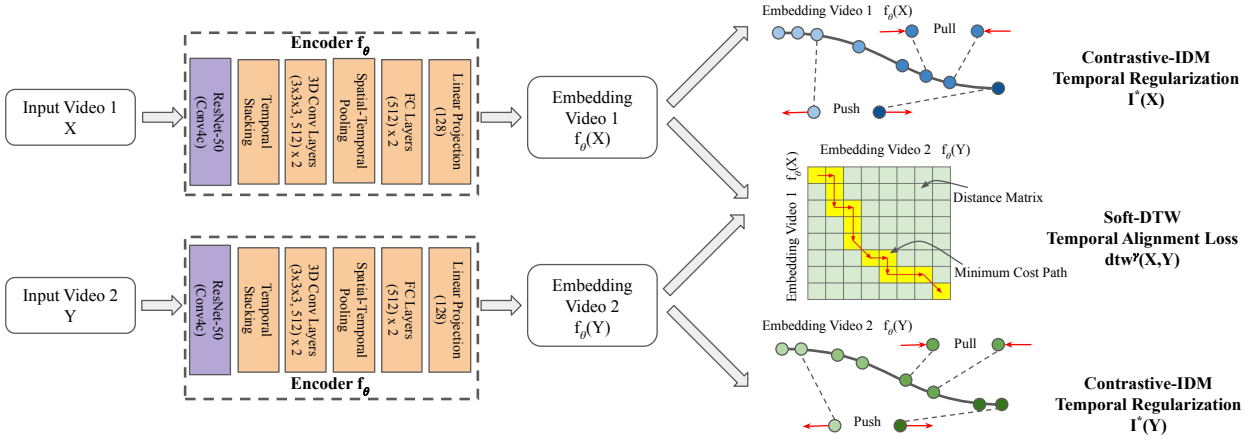


Figure 2: Given input videos X and Y, we feed them to the encoder f_θ to obtain the embedding videos $f_\theta(X)$ and $f_\theta(Y)$. We optimize the encoder parameters θ by applying the Soft-DTW temporal alignment loss $dtw^\gamma(X,Y)$ and the Contrastive-IDM temporal regularization $I^*(X)$ and $I^*(Y)$ on the embedding videos $f_\theta(X)$ and $f_\theta(Y)$.

embedding space (with large $S_X(i,j)$). Unlike [43], which applies IDM on the transport matrix between two (skeleton) sequences, we apply IDM separately on each (video) sequence. To be used as a loss function, we convert the above maximization objective to the below minimization:

$$\bar{I}(X) = \sum_{i=1}^{n}\sum_{j=1}^{n} \bar{W}(i,j)(-D_X(i,j)) \qquad (6)$$
$$\bar{W}(i,j) = (i-j)^2 + 1,$$

where $D_X \in \mathbb{R}^{n\times n}$ is the self-distance matrix of $f_\theta(X)$, and is defined as $D_X(i,j) = ||f_\theta(x_i) - f_\theta(x_j)||^2$. Minimizing Eq. 6 encourages temporally close frames in X (with small $\bar{W}(i,j)$) to be mapped to nearby points in the embedding space (with small $D_X(i,j)$).

However, we notice one problem with the above IDM regularization, in particular, it treats temporally close and far way frames in similar ways. In Eq. 5, it maximizes similarities between temporally far away frames, though with smaller weights. Similarly, for Eq. 6, it still maximizes distances between temporally close frames, though with smaller weights. To address that, we propose separate terms for temporally close and far away frames. Specifically, we introduce a contrastive version of Eq. 6, which we call *Contrastive-IDM*, as our regularization:

$$I^*(X) = \sum_{i=1}^{n}\sum_{j=1}^{n} y_{ij}\bar{W}(i,j)max(0, \lambda - D_X(i,j)) + (1-y_{ij})W(i,j)D_X(i,j), \qquad (7)$$
$$y_{ij} = \begin{cases} 1, & |i-j| > \sigma \\ 0, & |i-j| \leq \sigma \end{cases}.$$

Here, σ is a window size for separating temporally far away frames (y_{ij} = 1 or *negative* pairs) and temporally close frames (y_{ij} = 0 or *positive* pairs) and λ is a margin parameter. Contrastive-IDM encourages temporally close frames (positive pairs) to be nearby in the embedding space, while penalizing temporally far away frames (negative pairs) when the distance between them is smaller than margin λ in the embedding space. Note that, if we drop the weights $\bar{W}(i,j)$ and $W(i,j)$ in Eq. 7, it becomes equivalent to Slow Feature Analysis (SFA), also referred to as temporal coherence [21, 36, 20], which treats all pairs equally. We would emphasize that, leveraging temporal information by adding weights to different pairs based on their temporal gaps leads to performance gain, as we will show in Sec. 5.1.

3.3. Final Loss

Our final loss is a combination of Soft-DTW alignment loss in Eq. 4 and Contrastive-IDM regularization in Eq. 7:

$$L(X,Y) = dtw^\gamma(X,Y) + \alpha(I^*(X) + I^*(Y)). \qquad (8)$$

Here, α is the weight for the regularization. The final loss encourages embedding videos to have minimum alignment costs while encouraging discrepancies among embedding frames. Both the alignment loss and the regularization are differentiable and can be optimized using backpropagation.

3.4. Encoder Network

We use ResNet-50 [24] as our backbone network and extract features from the output of the $Conv4c$ layer. The extracted features have dimensions of $14\times14\times1024$. We then stack k context frame features along the temporal dimension for each frame. Next, the combined features are passed through two 3D convolutional layers for aggregating temporal information. It is then followed by a 3D global max pooling layer, two fully-connected layers, and a linear projection layer to output embedding frames, with each having 128 dimensions. We resize input video frames to 224×224 before feeding to our encoder network.

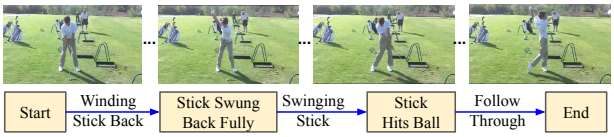


Figure 3: We annotate dense frame-wise labels (i.e., key events and phases) for 2123 videos of *Penn Action*. In the above *Golf Swing* video, key events represent specific events, e.g., *Stick Swung Back Fully*, while phases are periods between key events, e.g., *Winding Stick Back*.

4. Datasets, Annotations, and Metrics

Datasets and Annotations. We use three datasets, namely *Pouring* [41], *Penn Action* [54], and *IKEA ASM* [2]. While *Pouring* videos capture human hands interacting with objects, *Penn Action* and *IKEA ASM* videos show humans playing sports and assembling furniture respectively. We manually annotate dense frame-wise labels (i.e., key events and phases) for *Penn Action* using the same protocol of [14], since the authors of [14] do not release them. See Fig. 3 for an example. For *Pouring* and *IKEA ASM*, we obtain the labels from the authors of [14] and [2] respectively. Actions/videos in *IKEA ASM* (17 phases) are more complicated/longer than those in *Pouring* (5 phases) and *Penn Action* (2-6 phases). We use the training/validation splits from the original datasets. For *Pouring*, we use all videos (70 for training, 14 for validation). Following [14], we use 13 actions of *Penn Action* (for each action, 40-134 videos for training, 42-116 videos for validation). For *IKEA ASM*, we use all *Kallax_Drawer_Shelf* videos (61 for training, 29 for validation).

Evaluation Metrics. We use four evaluation metrics computed on the validation set. The network is first trained on the training set and then frozen. Next, an SVM classifier or linear regressor is trained on top of the frozen network features (without any fine-tuning of the network). For all metrics, a high score means a better model. We summarize the metrics below:

- *Phase Classification:* is the average per-frame phase classification accuracy, implemented by training an SVM classifier on top of the frozen network features to predict the phase labels.
- *Phase Progression [13, 52]:* measures the prowess of representations learnt to predict action progress temporally, implemented by training a linear regressor on top of the frozen network features to predict the phase progression values (defined using the key event labels).
- *Kendall's Tau [26, 51]:* measures how well videos are aligned temporally if we use nearest neighbor matching. It does not require any labels for evaluation.
- *Average Precision:* is the fine-grained frame retrieval accuracy, computed as the ratio of the retrieved frames with the same phase labels as the query frame.

We follow [14] to use the first three metrics above, while we add the last metric for our fine-grained frame retrieval experiments in Sec. 5.4. Phase Progression and Kendall's Tau assume no repetitive frames/labels in a video.

5. Experiments

In this section, we benchmark our approach (namely *LAV*, short for *L*earning by *A*ligning *V*ideos) against state-of-the-art methods for video-based self-supervised representation learning on various temporal understanding tasks on *Pouring*, *Penn Action*, and *IKEA ASM* datasets.

Implementation Details. We use the same encoder in Sec. 3.4 for all methods for *Pouring* and *Penn Action* experiments. For *IKEA ASM* experiments, since the actions are more complex, we opt to extract features from the output of the $Conv5c$ layer (instead of $Conv4c$) for all methods. We initialize ResNet-50 layers with pre-trained weights for ImageNet classification, while remaining layers are initialized randomly. We L2-normalize the frame-embeddings before feeding them to our loss (LAV). We use ADAM optimization [29] with a learning rate of 10^{-4} and a weight decay of 10^{-5}. We minimize our final loss in Eq. 8 computed over all video pairs in the training set. We randomly pair videos of the same action, *regardless* of their viewpoints. For datasets with a single action (e.g., *Pouring* and *IKEA ASM*), videos are randomly paired. For datasets with many actions (e.g., *Penn Action*), videos of the same action are randomly paired. For each video pair, we calculate the final loss using p sampled frames from each video, i.e., we divide a video into p uniform chunks and randomly sample one frame per chunk. We implement our network and loss in PyTorch [38]. For more details, please refer to supplementary materials.

Competing Methods. Below are the competing methods:

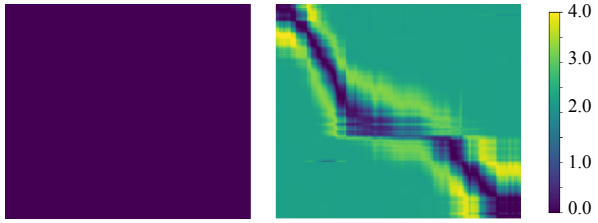
- *Self-Supervised Learning:* We compare LAV with recent self-supervised video representation learning methods, namely SAL [35], TCN [41], and TCC[14].
- *Fully-Supervised Learning:* We test LAV against a fully-supervised method with explicit supervision. Specifically, following [14], we train a network on the downstream task by attaching a 1-layer classifier to the encoder in Sec. 3.4.
- *Random/ImageNet Features:* For completeness, we include the results obtained by using random features or pre-trained features for ImageNet classification.

5.1. Ablation Study Results

Here, we perform ablation studies on *Pouring* dataset to show the effectiveness of our design choices in Sec. 3.

	Loss	Classification	Progress	τ
Individual	S-DTW [11]	48.35	0.2770	0.2144
	IDM (Eq. 6)	48.16	0.7241	0.5835
	SFA [21]	92.20	0.7533	0.8093
	C-IDM (Eq. 7)	92.82	0.7477	0.8318
Combined	S-DTW + IDM	68.73	0.6551	0.6408
	S-DTW + SFA	91.63	0.7146	0.8069
	S-DTW + C-IDM	**92.84**	**0.8054**	**0.8561**

Table 1: Ablation studies of individual losses (top) and combined losses (bottom). S-DTW and C-IDM denote Soft-DTW and Contrastive-IDM respectively. Best results are in **bold**, while second best ones are underlined.



(a) S-DTW (b) S-DTW+C-IDM

Figure 4: Distance matrices between embedding frames of two *Pouring* videos learned by various losses. S-DTW denotes Soft-DTW, while C-IDM means Contrastive-IDM.

Performance of Individual Losses. We first study the performance of individual components of our approach, i.e., Soft-DTW and Contrastive-IDM, as separate baselines. Also, we include other methods such as IDM in Eq. 6 and an SFA approach proposed in [21]. Tab. 1 (top) presents the quantitative results. We observe that the model trained with Soft-DTW alone achieves the lowest accuracy across all metrics. In fact, it has similar classification accuracy to that of random features in Tab. 2 (i.e., 48.35% vs. 45.10%), which shows that training solely with Soft-DTW yields trivial solutions and the network is unable to learn any useful representations. This is also confirmed by plotting the distance matrix between the embedding frames learned solely with Soft-DTW in Fig. 4(a), where all entries are near zero. In other words, the frames are mapped to a small cluster in the embedding space. Moreover, it can be seen from Tab. 1 (top) that Contrastive-IDM outperforms IDM by significant margins on all metrics (e.g., for Kendall's Tau, 0.8318 vs. 0.5835), showing the advantage of using separate terms for temporally close and far away frame pairs. Lastly, although SFA and Contrastive-IDM have competitive performances on classification and progression, Contrastive-IDM outperforms SFA significantly on Kendall's Tau (i.e., 0.8318 vs. 0.8093), supporting our idea of adding weights to different frame pairs based on their temporal gaps.

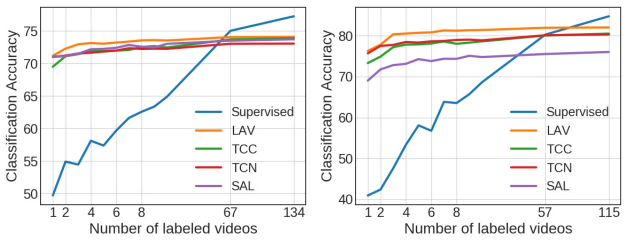
Performance of Combined Losses. We now study the impact of adding IDM, SFA, or Contrastive-IDM as regularization to Soft-DTW. Tab. 1 (bottom) presents the quantitative results. From the results, the addition of regularization boosts the performance of Soft-DTW significantly across all metrics (e.g., for progression, 0.2770 for Soft-DTW vs. 0.8054 for Soft-DTW+Contrastive-IDM). More importantly, utilizing our proposed Contrastive-IDM as regularization leads to the best performance across all metrics, outperforming using IDM or SFA as regularization by significant margins, especially on progression and Kendall's Tau (e.g., for progression, 0.8054 for Soft-DTW+Contrastive-IDM vs. 0.6551 and 0.7146 for Soft-DTW+IDM and Soft-DTW+SFA respectively). This validates our ideas of separating temporally close and far away frame pairs, as well as leveraging temporal gaps to weight the frame pairs accordingly. We also visualize the distance matrix between the embedding frames learned with Soft-DTW+Contrastive-IDM in Fig. 4(b), where entries have diverse values. Below, we use Soft-DTW+Contrastive-IDM as our method (LAV).

	% of labels →	0.1	0.5	1.0
Pouring	Supervised Learning	72.44	89.57	92.86
	Random Features	43.84	44.52	45.10
	Imagenet Features	52.40	71.10	78.46
	SAL [35]	87.63	87.58	88.81
	TCN [41]	89.67	87.32	89.53
	TCC [14]	90.65	91.11	91.53
	LAV (Ours)	91.61	**92.82**	92.84
	LAV + TCC (Ours)	**92.78**	92.56	**93.07**
Penn Action	Supervised Learning	68.92	81.17	84.34
	Random Features	47.05	47.19	47.65
	Imagenet Features	46.66	56.39	60.65
	SAL [35]	79.94	81.11	81.79
	TCN [41]	81.99	82.64	82.78
	TCC [14]	79.72	81.12	81.35
	LAV(Ours)	**83.56**	**83.95**	**84.25**
	LAV + TCC (Ours)	83.21	83.79	84.12
IKEA ASM	Supervised Learning	21.76	30.26	33.81
	Random Features	17.89	17.89	17.89
	Imagenet Features	18.05	19.27	19.50
	SAL [35]	21.68	21.72	22.14
	TCN [41]	25.17	25.70	26.80
	TCC [14]	24.74	25.22	26.46
	LAV (Ours)	**29.78**	29.85	30.43
	LAV + TCC (Ours)	24.58	**30.47**	**30.51**

Table 2: Phase classification results. Best results are in **bold**, while second best ones are underlined.

5.2. Phase Classification Results

In this section, we evaluate the utility of our representations for action phase classification. Tab. 2 presents the quantitative results of all methods on *Pouring*, *Penn Action*,



(a) Bowling (b) Tennis Serve

Figure 5: Few-shot phase classification results.

and *IKEA ASM* datasets. For *Penn Action* experiments, we follow [14] to train 13 different models (i.e., 1 encoder + 1 SVM classifier, for each action) and report the average results across all actions. It can be seen from Tab. 2 that our method (LAV) outperforms other self-supervised video representation learning methods, namely SAL [35], TCN [41], and TCC [14], on all datasets. This shows that LAV is more capable of learning useful features that allow good classification performance when combined with a relatively simple classifier. Moreover, the best accuracy on *Pouring* and *IKEA ASM* is achieved by the combined LAV+TCC, which is similar to the observation in [14], where combining multiple losses leads to better classification performance. Next, the relative gaps between LAV and other self-supervised methods are the largest on *IKEA ASM*, which has more complex actions than *Pouring* and *Penn Action*. This implies that LAV is more capable of handling complex actions. Finally, compared to the fully-supervised baseline, self-supervision with LAV provides a significant performance boost in the low labeled data regimes. Specifically, with just 10% labeled data, LAV achieves very similar performance to the fully-supervised baseline trained with 100% labeled data (e.g., on *Penn Action*, 83.56% vs. 84.34%).

Few-Shot Phase Classification Results. Following the above observation, we consider the application of our representations in a few-shot learning setting, i.e., there are many training videos, but only a few of them have frame-wise labels. We use the same setup as the above experiment, and compare our approach with other self-supervised methods and the fully-supervised baseline. For learning self-supervised features, all training videos are used, whereas the fully-supervised baseline is trained with a few labeled videos. Specifically, we study the classification performance with increasing the number of labeled videos. The results for two actions of *Penn Action* are reported in Fig. 5. Although all self-supervised methods offer a significant performance boost in the low labeled data settings, LAV provides the largest gain. Moreover, self-supervision using LAV with only 1 labeled video performs similarly to the fully-supervised baseline trained with the whole dataset. For instance, on *Bowling*, with just 1 labeled video, LAV achieves 71%, whereas the fully-supervised baseline trained with the entire dataset (134 labeled videos) obtains 77%.

	Method	Progress	τ
Pouring	SAL [35]	0.7728	0.7961
	TCN [41]	0.8044	0.8521
	TCC [14]	**0.8373**	**0.8636**
	LAV (Ours)	<u>0.8054</u>	<u>0.8561</u>
	LAV + TCC (Ours)	0.7716	0.7844
Penn Action	SAL [35]	0.6960	0.7612
	TCN [41]	**0.7217**	**0.8120**
	TCC [14]	0.6638	0.7012
	LAV (Ours)	0.6613	<u>0.8047</u>
	LAV + TCC (Ours)	<u>0.7038</u>	0.7729

Table 3: Phase progression and Kendall's Tau results. Best results are in **bold**, while second best ones are <u>underlined</u>.

5.3. Phase Progression and Kendall's Tau Results

We now evaluate the performance of our approach on action phase progression and Kendall's Tau. Tab. 3 presents the quantitative results of different self-supervised methods on *Pouring* and *Penn Action*. We do not evaluate on *IKEA ASM*, since its labels are repeated (i.e., the actions of picking up left side panel and picking up right side panel are both labeled as *Pick Up Side Panel*, thus *Pick Up Side Panel* is repeated). From the results, we achieve competitive numbers for both progression and Kendall's Tau on both *Pouring* and *Penn Action*. On *Pouring*, LAV marginally beats TCN on both metrics (e.g., for progression, 0.8054 vs. 0.8044), while on *Penn Action*, LAV significantly outperforms TCC on Kendall's Tau (i.e., 0.8047 vs. 0.7012). Moreover, on *Penn Action*, the combination of LAV+TCC yields a significant performance gain over TCC on both metrics (e.g., for Kendall's Tau, 0.7729 vs. 0.7012).

5.4. Fine-Grained Frame Retrieval Results

Here, we utilize our representations for the task of fine-grained frame retrieval. We perform evaluations using the validation set of *Pouring* and *Penn Action*. In particular, we alternatively consider each video of the validation set as a query video and all the remaining videos of the validation set as a support set. For each query frame in the query video, we retrieve its K most similar frames in the support set by finding its K nearest neighbors in the embedding space. We report Average Precision at K, which is the average percentage of the K retrieved frames with the same action phase labels as the query frame. Tab. 4 presents the quantitative results of various self-supervised methods on *Pouring* and *Penn Action*. It is evident from Tab. 4 that LAV consistently achieves the best performance across different values of K on both datasets (e.g., on *Pouring*, for AP@5, 89.13% for

	Method	AP@5	AP@10	AP@15
Pouring	SAL [35]	84.05	83.77	83.79
	TCN [41]	83.56	83.31	83.01
	TCC [14]	87.16	86.68	86.54
	LAV (Ours)	**89.13**	**89.13**	**89.22**
	LAV + TCC (Ours)	89.00	88.96	88.78
Penn Action	SAL [35]	76.04	75.77	75.61
	TCN [41]	77.84	77.51	77.28
	TCC [14]	76.74	76.27	75.88
	LAV (Ours)	**79.13**	**78.98**	**78.90**
	LAV + TCC (Ours)	78.98	78.83	78.70

Table 4: Fine-grained frame retrieval results. Best results are in **bold**, while second best ones are underlined.

Figure 6: Qualitative fine-grained frame retrieval results with $K = 5$. On the left is the query image. On the right are the **blue** and **red** boxes containing the 5 most similar images to the query image retrieved by LAV and TCC respectively.

LAV vs. 87.16%, 83.56%, and 84.05% for TCC, TCN, and SAL respectively). This shows that our method is better at learning fine-grained features, which are important to this task. Also, the combined LAV+TCC leads to a significant performance gain over TCC (e.g., 78.7% vs. 75.88%).

Moreover, we present some qualitative results with $K = 5$ in Fig. 6, showing that LAV is more capable of capturing fine-grained features than TCC. In Fig. 6(a), the person in the query image has one leg elevated above the ground, which is also seen in 4 out of 5 images retrieved by LAV, whereas TCC fails to capture that in all of its retrieved images (see cyan circles). In Fig. 6(b), the actor in the query image is at the start of *Golf Swing* with the ball on the ground, which is also seen in all of LAV's retrieved images, whereas TCC retrieves images with wrong phases (i.e., the person has finished *Golf Swing* with the ball not visible on the ground, see magenta circles).

	Method	Classification	Progress	τ
Penn Action	SAL [35]	68.15	0.3903	0.4744
	TCN [41]	68.09	0.3834	0.5417
	TCC [14]	74.39	0.5914	0.6408
	LAV (Ours)	**78.68**	**0.6252**	**0.6835**

Table 5: Joint all-action model results. Best results are in **bold**, while second best ones are underlined.

5.5. Joint All-Action Model Results

So far, we have followed [14] to train a separate model for each action of *Penn Action* and report the average results across all actions. This is not convenient both in terms of training time and memory requirement. In this section, we explore another experimental setup, where we jointly train a single model for all actions of *Penn Action*. In particular, we train 13 SVM classifiers (1 for each action) but share a single encoder. It is more challenging, since the network needs to jointly learn useful features for all actions. Tab. 5 shows the quantitative results of different self-supervised methods in the above setup. We observe that the performance of all methods is reduced as compared to Tabs. 2 and 3. Moreover, we notice LAV achieves the best performance across all metrics, outperforming TCC, TCN, and SAL in Tab. 5. This can be attributed to the fact that LAV leverages information from across videos in addition to cues from each individual video.

Additional Results. Note that due to space limits, we provide several additional experimental results, including training-from-scratch results and ablation results of hyperparameter settings, in supplementary materials.

6. Conclusion

In this work, we propose a novel fusion of temporal alignment loss and temporal regularization for learning self-supervised video representations via temporal video alignment, utilizing both frame-level and video-level cues. The two components are complementary to each other, i.e., temporal regularization prevents degenerate solutions while temporal alignment loss leads to higher performance. We show superior performance over prior methods for video-based self-supervised representation learning on various temporal understanding tasks on *Pouring*, *Penn Action*, and *IKEA ASM* datasets. Also, our method offers significant accuracy gain when lacking labeled data. Our future work will explore other temporal alignment losses, e.g., [43, 5], to allow local temporal permutations and arbitrary video starting/ending points.

Acknowledgements. We would like to thank D. Dwibedi for releasing the code and answering questions about TCC.

References

[1] Unaiza Ahsan, Chen Sun, and Irfan Essa. Discrimnet: Semi-supervised action recognition from videos using generative adversarial networks. *arXiv preprint arXiv:1801.07230*, 2018. 2

[2] Yizhak Ben-Shabat, Xin Yu, Fatemeh Sadat Saleh, Dylan Campbell, Cristian Rodriguez-Opazo, Hongdong Li, and Stephen Gould. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. *arXiv preprint arXiv:2007.00394*, 2020. 5

[3] Yoshua Bengio and James S Bergstra. Slow, decorrelated features for pretraining complex cell-like networks. In *Advances in neural information processing systems*, pages 99–107, 2009. 2

[4] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, USA:, 1994. 2, 3

[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. 2, 3, 8

[6] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019. 2

[7] 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. 2

[8] Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. D3TW: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3546–3555, 2019. 2, 3

[9] Jinwoo Choi, Gaurav Sharma, Samuel Schulter, and Jia-Bin Huang. Shuffle and attend: Video domain adaptation. In *European Conference on Computer Vision*, pages 678–695. Springer, 2020. 2

[10] Richard W Conners and Charles A Harlow. A theoretical comparison of texture algorithms. *IEEE transactions on pattern analysis and machine intelligence*, (3):204–222, 1980. 3

[11] Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *International Conference on Machine Learning*, pages 894–903, 2017. 2, 3, 6

[12] Ali Diba, Vivek Sharma, Luc Van Gool, and Rainer Stiefelhagen. Dynamonet: Dynamic action and motion network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6192–6201, 2019. 2

[13] Norman R Draper and Harry Smith. *Applied regression analysis*, volume 326. John Wiley & Sons, 1998. 5

[14] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1801–1810, 2019. 1, 2, 3, 5, 6, 7, 8

[15] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3575–3584, 2019. 2

[16] Zeyu Feng, Chang Xu, and Dacheng Tao. Self-supervised representation learning by rotation feature decoupling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10364–10374, 2019. 2

[17] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3636–3645, 2017. 2

[18] Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Predicting the future: A jointly learnt model for action anticipation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5562–5571, 2019. 2

[19] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018. 2

[20] Ross Goroshin, Joan Bruna, Jonathan Tompson, David Eigen, and Yann LeCun. Unsupervised learning of spatiotemporally coherent metrics. In *Proceedings of the IEEE international conference on computer vision*, pages 4086–4093, 2015. 2, 4

[21] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 2, 4, 6

[22] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019. 2

[23] Sanjay Haresh, Sateesh Kumar, M Zeeshan Zia, and Quoc-Huy Tran. Towards anomaly detection in dashcam videos. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1407–1414. IEEE. 2

[24] 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. 4

[25] Geoffrey E Hinton and Richard S Zemel. Autoencoders, minimum description length and helmholtz free energy. In *Advances in neural information processing systems*, pages 3–10, 1994. 2

[26] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938. 5

[27] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8545–8552, 2019. 2

[28] Dahun Kim, Donghyeon Cho, Donggeun Yoo, and In So Kweon. Learning image representations by completing damaged jigsaw puzzles. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 793–802. IEEE, 2018. 2

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[30] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European conference on computer vision*, pages 577–593. Springer, 2016. 2

[31] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6874–6883, 2017. 2

[32] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 667–676, 2017. 2

[33] S. Li, Yazan Abu Farha, Yun Liu, Ming-Ming Cheng, and Juergen Gall. Ms-tcn++: Multi-stage temporal convolutional network for action segmentation. *IEEE transactions on pattern analysis and machine intelligence*, PP, 2020. 2

[34] Xialei Liu, Joost Van De Weijer, and Andrew D Bagdanov. Leveraging unlabeled data for crowd counting by learning to rank. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7661–7669, 2018. 2

[35] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016. 2, 5, 6, 7, 8

[36] Hossein Mobahi, Ronan Collobert, and Jason Weston. Deep learning from temporal coherence in video. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 737–744, 2009. 2, 4

[37] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5898–5906, 2017. 2

[38] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[39] Senthil Purushwalkam, Tian Ye, Saurabh Gupta, and Abhinav Gupta. Aligning videos in space and time. In *European Conference on Computer Vision*, pages 262–278. Springer, 2020. 1, 2

[40] Alexander Richard. *Temporal Segmentation of Human Actions in Videos*. PhD thesis, Universitäts-und Landesbibliothek Bonn, 2019. 2

[41] 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. 2, 5, 6, 7, 8

[42] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852, 2015. 2

[43] Bing Su and Gang Hua. Order-preserving wasserstein distance for sequence matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1049–1057, 2017. 3, 4, 8

[44] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6479–6488, 2018. 2

[45] 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. 2

[46] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 2

[47] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 2

[48] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances in neural information processing systems*, pages 613–621, 2016. 2

[49] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 2

[50] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8052–8060, 2018. 2

[51] Wikipedia. Kendall rank correlation coefficient — Wikipedia, the free encyclopedia, 2020. 5

[52] Wikipedia contributors. Coefficient of determination — Wikipedia, the free encyclopedia, 2021. [Online; accessed 22-March-2021]. 5

[53] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10334–10343, 2019. 2

[54] 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. 5

[55] Will Zou, Shenghuo Zhu, Kai Yu, and Andrew Y Ng. Deep learning of invariant features via simulated fixations in video. In *Advances in neural information processing systems*, pages 3203–3211, 2012. 2

[56] Will Y Zou, Andrew Y Ng, and Kai Yu. Unsupervised learning of visual invariance with temporal coherence. In *NIPS 2011 workshop on deep learning and unsupervised feature learning*, volume 3, 2011. 2