Self-Supervised Video Representation Learning via Latent Time Navigation

Di Yang[1,2], Yaohui Wang[1,2,5*], Quan Kong[4], Antitza Dantcheva[1,2], Lorenzo Garattoni[3], Gianpiero Francesca[3], François Brémond[1,2]

[1]Inria, 2004 Rte des Lucioles, Valbonne, France
[2]Université Côte d'Azur, 28 Av. de Valrose, Nice, France
[3]Toyota Motor Europe, 60 Av. du Bourget, Brussels, Belgium
[4]Woven Planet Holdings, 3-2-1 Nihonbashimuromachi, Chuo-ku, Tokyo, Japan
[5]Shanghai AI Laboratory, 701 Yunjin Road, Shanghai, China
{di.yang, yaohui.wang, antitza.dantcheva, francois.bremond}@inria.fr,
{lorenzo.garattoni, gianpiero.francesca}@toyota-europe.com, quan.kong@woven-planet.global

Abstract

Self-supervised video representation learning aimed at maximizing similarity between different temporal segments of one video, in order to enforce feature persistence over time. This leads to loss of pertinent information related to temporal relationships, rendering actions such as 'enter' and 'leave' to be indistinguishable. To mitigate this limitation, we propose Latent Time Navigation (LTN), a time-parameterized contrastive learning strategy that is streamlined to capture fine-grained motions. Specifically, we maximize the representation similarity between different video segments from one video, while maintaining their representations *time-aware* along a subspace of the latent representation code including an orthogonal basis to represent temporal changes. Our extensive experimental analysis suggests that learning video representations by LTN consistently improves performance of action classification in fine-grained and human-oriented tasks (*e.g.*, on Toyota Smarthome dataset). In addition, we demonstrate that our proposed model, when pre-trained on Kinetics-400, generalizes well onto the unseen real world video benchmark datasets UCF101 and HMDB51, achieving state-of-the-art performance in action recognition.

Introduction

Contrastive learning (Hadsell, Chopra, and LeCun 2006) is a prominent variant in learning self-supervised visual representations. The associated objective is to minimize the distance between latent representations of positive pairs, while maximizing the distance between latent representations of negative pairs. For instance, a visual encoder aims at learning the invariance of multiple *views* of a scene, which constitute positive pairs, by extracting generic features of images (Bachman, Hjelm, and Buchwalter 2019; Caron et al. 2020; Chen et al. 2020; Grill et al. 2020; He et al. 2020; Hjelm et al. 2019; Jiao et al. 2020; Tian, Krishnan, and Isola 2020; Wu et al. 2018) or videos (Feichtenhofer et al. 2021; Han, Xie, and Zisserman 2020; Huang et al. 2019; Kong et al. 2020; Li et al. 2021a,b; Park et al. 2022; Yang et al. 2021b, 2022; Sun et al. 2021). Then, the trained visual encoder can be transferred onto other downstream tasks.

Remarkable results have been reported by augmentation-invariant contrastive learning. In this context, contrastive learning methods enable the visual encoder to find compact and meaningful image representations, invariant to data augmentation. The latent representation of two augmented views of the same instance are enforced to be similar via contrastive learning. In *image-based tasks*, a common augmentation method relates to random cropping (Chen et al. 2020; Wu et al. 2018). When extending this idea to *videos*, which are endowed with additional temporal information, cropping in the spatial dimension (Kong et al. 2020) is not sufficient for training an effective visual encoder. Therefore, recent works (Feichtenhofer et al. 2021; Li et al. 2021b; Sun et al. 2021) sample different views with a *temporal shift*, learning representations that are invariant to time changes. However, for downstream tasks involving temporal relationships, a representation invariant to temporal shifts might omit valuable information. For instance, in differentiating actions such as 'enter' and 'leave' the temporal order is fundamental. Hence, a trained visual encoder remains a challenge in handling downstream video understanding tasks such as fine-grained human action recognition (Das et al. 2019; Goyal et al. 2017; Li et al. 2021c).

Motivated by the above, we propose Latent Time Navigation (LTN), a time parameterization scheme streamlined to learn time-aware representations on top of the contrastive module. As illustrated in Fig. 1, deviating from current contrastive methods (Feichtenhofer et al. 2021; He et al. 2020; Tian, Krishnan, and Isola 2020; Wu et al. 2018) which directly maximize the similarity between representations obtained from the visual encoder for positive samples, LTN encompasses the following steps. Firstly, we decompose a subspace (*i.e.*, a learnable orthogonal basis and associated magnitudes) from the latent representation code for the video segment, namely 'time-encoded component', to do with temporal changes (*e.g.*, changes in appearances, motion, object locations). The other subspace ('time-invariant component') has to do with invariant information. Subsequently, we embed the *time shift value* used for generating data view into a high-dimensional vector as the magnitudes of the directions in the orthogonal basis and then encode this time information into the 'time-encoded component' by linear combination of the orthogonal basis and the magnitudes. Finally, we con-

*Work done while the author was at Inria

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Figure 1: Current methods (left) leverage on contrastive learning to maximize representation similarities of multiple positive views (segments with time spans and data augmentation) of the same video instance to represent them as a consistent representation. To further improve the representation capability for fine-grained tasks without losing important motion variance, our approach (right) incorporates a time-parameterized contrastive learning (LTN) to remain the video representations aware to time shifts (starting time) in a decomposed time-encoded subspace.

duct contrastive learning on the entire time-parameterized representations in order to maximize the similarity between positive pairs along the 'time-invariant component', while maintaining their representations *time-aware* along the 'time-encoded component'. We note that LTN incorporates time information for video representations and therefore is able to model subtle motions within an action. Consequently, the time-aware representation obtained from the trained visual encoder generalizes better to unseen action recognition datasets, especially to our target human-oriented fine-grained action classification dataset (Das et al. 2019).

In summary, the contributions of this paper include the following. (a) We propose Latent Time Navigation (LTN) to parameterize the time information (used for generating data views) on top of contrastive learning, in order to learn a *time-aware* video representation. (b) We demonstrate that LTN can effectively learn the consistent amount of temporal changes with the video segments on the decomposed 'time-encoded components'. (c) We set a new state-of-the-art with LTN on the real world dataset (*e.g.*, Toyota Smarthome) for fine-grained action recognition with self-supervised action representation learning. (d) We demonstrate that our proposed model, when pre-trained on Kinetics-400 dataset, generalizes well to unseen real-world video benchmarks (*e.g.*, UCF101 and HMDB51) with both linear evaluation and fine-tuning.

Related Work

Contrastive Learning. Contrastive learning and its variants (Bachman, Hjelm, and Buchwalter 2019; Caron et al. 2020; Chen et al. 2020; Grill et al. 2020; He et al. 2020; Hjelm et al. 2019; Jiao et al. 2020; Tian, Krishnan, and Isola 2020; Wu et al. 2018) have established themselves as a pertinent direction for self-supervised representation learning for a number of tasks due to promising performances. Recent video representation learning methods (Feichtenhofer et al. 2021; Huang et al. 2019; Kong et al. 2020) are inspired by image techniques. The objective of such techniques is to encourage representational invariances of different views (*i.e.*, positive pairs) of the same instance obtained by data augmentation, *e.g.*, random cropping (Chen et al. 2020; Wu et al. 2018), rotation (Misra and van der Maaten 2020), while spreading representations of views from different instances (*i.e.*, negative pairs) apart. To further improve the representation capability, CMC (Misra and van der Maaten 2020) scaled contrastive learning to any number of views. MoCo (He et al. 2020) incorporated a dynamic dictionary with a queue and a moving-averaged encoder. To omit a large number of negative pairs, BYOL (Grill et al. 2020) and SwAV (Caron et al. 2020) were targeted to solely rely on positive pairs. However, these methods miss a crucial Time element when they are straightforward applied to the *video* domain with views generated by *image* data augmentation technique. In our work, we adopt recent contrastive learning frameworks (Grill et al. 2020; He et al. 2020) and we focus on learning time-aware representations for videos by latent spatio-temporal decomposition and navigation in the representation space.

Self-supervised Video Representation Learning. Approaches for self-supervised video representation learning exploit spatio-temporal pretext tasks from numerous unlabeled data. Towards effective extraction of the pertinent motion information in the time dimension, a number of temporal pretext tasks were proposed, *e.g.*, pixel-level future generation (Mathieu, Couprie, and LeCun 2016; Srivastava, Mansimov, and Salakhutdinov 2015; Vondrick, Pirsiavash, and Torralba 2016b; Vondrick et al. 2018) and jigsaw-solving (Kim, Cho, and Kweon 2019). Additionally, in order to facilitate the learning process, numerous works focused on learning representations in a more abstract space including temporal order (Misra, Zitnick, and Hebert 2016; Xu et al. 2019) or arrow (Wei et al. 2018) prediction of video frames, future prediction (Vondrick, Pirsiavash, and Torralba 2016a), speed prediction (Benaim et al. 2020), motion prediction (Diba

Figure 2: Overview of the proposed LTN framework. At each training iteration, given an input video, (a) a query clip (q) and multiple positive key clips ($k_1^+, k_2^+, ..., k_P^+$) are generated by data augmentation with different temporal shifts $\mathbf{dt}$. All clips are then fed to a visual encoder that extracts spatio-temporal features for each clip. To learn time-aware representations for query and key clips, (b) we first pre-define a learnable orthogonal basis $\mathbf{D}_t$ ($\mathbf{d_1}, \mathbf{d_2}, ..., \mathbf{d_M}$) that represents the 'time-encoded component'. The video representations are expected to be time-aware along $\mathbf{D}_t$ in the training stage. To do so, we transform each query and key video representation (*i.e.*, $\mathbf{f}(q)$, $\mathbf{f}(k_p^+)$) by a linear combination of $\mathbf{D}_t$ and associated magnitudes learned from its time shift $\mathbf{dt}$ to a time-blended position (*i.e.*, $\mathbf{f}'(q, \mathbf{dt}_q)$, $\mathbf{f}'(k_p^+, \mathbf{dt}_p)$, abbreviated as $\mathbf{f}'(q)$, $\mathbf{f}'(k_p^+)$). Finally, we conduct (c) contrastive learning on top of $\mathbf{f}'$, so that the learned representation from the visual encoder can maintain temporal awareness.

et al. 2019) and a combination of these tasks (Bai et al. 2020). These methods are highly constrained by the limited quality of pretext tasks. Recently, video contrastive learning methods (Huang et al. 2019; Kong et al. 2020) have obtained promising results and a large-scale study (Feichtenhofer et al. 2021) has been conducted to compare state-of-the-art image-based contrastive methods (Caron et al. 2020; Chen et al. 2020; Grill et al. 2020; He et al. 2020) on videos using spatio-temporal cropping, color jitters and Gaussian blur data augmentation techniques to generate multiple video views. Further, to improve representation performance, (Ding et al. 2022; Huang et al. 2021; Ranasinghe et al. 2022) focused on view generation techniques, *e.g.*, context-motion decoupling (Huang et al. 2021), foreground-background merging (Ding et al. 2022), global and local sampling across space and time (Ranasinghe et al. 2022). In addition, some specific designs are incorporated in spatio-temporal representation learning including Gaussian probabilistic representations (Park et al. 2022), skeleton contrastive learning (Li et al. 2021a; Yang et al. 2021b; Das et al. 2021) and muti-modal learning with audio (Bruno, Du, and Lorenzo 2019; Dwibedi et al. 2019; Patrick et al. 2021; Recasens et al. 2021; Shuang et al. 2021; Xiaolong, Allan, and Alexei A 2019) or with optical flow (Han, Xie, and Zisserman 2020; Li et al. 2021b). Such contrastive methods aimed at learning video representations invariant to time shift. However, motion significantly changes with time shifts, leading to poor performance on downstream fine-grained action recognition tasks that highly rely on the motion variance. To address this issue, CATE (Sun et al. 2021) proposed to parameterize data augmentation relying on an additional Transformer head prior to contrastive learning. It demonstrated that awareness of the temporal data augmentation is particularly instrumental in fine-grained action recognition tasks. Deviating from CATE that shifts the entire visual representation along all dimensions by the time-shift values even for the action with small motion variances, we study variant time-parameterization strategies and propose to encode the time-shift values partially on certain orthogonal directions instead of on the entire visual representation. By our proposed LTN, the impact of time can be video specific and controlled by the number of the orthogonal directions so that the visual encoder can better capture motions.

Proposed Approach

In this section we introduce our Latent Time Navigation (LTN) framework. We start with the overall architecture, then we proceed to describe the design strategies focusing on time parameterization that enforces the learned video representation to be aware of motion variances.

Overall Architecture of LTN

Our objective is to train a generic visual encoder $\mathbf{f}$ for extracting accurate spatio-temporal features of video clips. We design our visual encoder to be efficient for downstream fine-grained action recognition tasks. We illustrate the overview of the architecture in Fig. 2. To train the visual encoder, a general data augmentation technique including random temporal shifts is applied to generate multiple positive views for a given input video, allowing us to obtain multiple representations from different views. Deviating from previous methods (He et al. 2020; Tian, Krishnan, and Isola 2020), which directly employ contrastive learning for these representations in order to make them invariant to spatio-temporal augmentation, we design an additional time parameterization module to blend temporal augmentation to a 'time-encoded component' prior to contrastive learning. We then perform the contrastive learning for the new time-blended represen-

tations in the training stage. The trained visual encoder can thus be aware of time shifts compared to other positive pairs and can capture the important motion variances of videos for improving fine-grained action recognition tasks.

View Generation and Embedding. Following the study (Feichtenhofer et al. 2021), we first spatio-temporally crop a segment by randomly selecting a segment and cropping out a fixed-size box from the same video instance. We then pull together image-based augmentations including random horizontal flip, color distortion and Gaussian blur following (Chen et al. 2020; He et al. 2020) to generate positive views of the input video at each training iteration. As demonstrated in (Feichtenhofer et al. 2021), multiple positive samples with large time spans between them are beneficial in downstream performance. In our work, we sample a query clip noted as q and multiple positive keys with large time spans, noted as $k_1^+, ..., k_P^+$ (see Fig. 2 (a)). We utilize a 3D-CNN network (Hara, Kataoka, and Satoh 2017) as the visual encoder to obtain dim-dimensional representations of all clips (*i.e.*, $\mathbf{f}(q), \mathbf{f}(k_1^+), ..., \mathbf{f}(k_P^+) \in \mathbb{R}^{1\times dim}$).

Awareness of Time in Latent Space. Large time spans between positive samples may depict significant changes in human motion. When directly matching $\mathbf{f}(q)$ to all positive pairs, the corresponding representations may lose pertinent motion variance caused by time shifts. This could compromise the accuracy of downstream tasks related to fine-grained human motion (*e.g.*, classification of 'Leave/Enter', 'Stand up/Sit down'). Hence, we expect positive pairs to be partially similar with each other (due to static object, scene) while also partially aware of their time shifts to preserve temporal dynamic information (e.g., changes in motion). To do so, we design several time parameterization methods (see Sec. Time Parameterization in Latent Space) to encode the time shift value (denoted as $\mathbf{dt_q}$ for q) used for data augmentation to a part (several orthogonal directions) of the visual representation while keep the remaining part unchanged. Such time-encoded pretext representation of q and each positive key can be computed and denoted as $\mathbf{f}'(q, \mathbf{dt_q})$ and $\mathbf{f}'(k_p^+, \mathbf{dt_p})$. We then maximize the mutual information between the pretext representations $\mathbf{f}'(q, \mathbf{dt_q})$ and $\mathbf{f}'(k_p^+, \mathbf{dt_p})$ by contrastive learning. The original (target) visual representations from different segments (*e.g.*, $\mathbf{f}(q)$, $\mathbf{f}(k_p^+)$) will be sensitive to time along the time-encoded part after learning and can be transferred onto downstream tasks.

Time Parameterization in Latent Space

We first introduce the latent space decomposition approach to split the representation space into 'time-encoded component' and 'time-invariant components', and then we introduce time encoding which is used as a parameter to transform the visual representation only along the 'time-encoded component' to reach a new time-blended position.

Latent Space Decomposition. To decompose the representation space, we set a learnable orthogonal basis (*i.e.*, a subspace) $\mathbf{D}_t = \{\mathbf{d_1}, \mathbf{d_2}, ..., \mathbf{d_M}\}$ with $M \in [1, dim)$, and $\mathbf{d} \in \mathbb{R}^{dim\times 1}$ to represent the 'time-encoded component', where each vector indicates a basic visual transformation. Due to $\mathbf{D}_t$ entailing an orthogonal basis, any two directions $\mathbf{d_i}, \mathbf{d_j}$ follow the constraint in Eq. 1. We implement $\mathbf{D}_t \in \mathbb{R}^{dim\times M}$ as a learnable matrix following (Wang et al. 2022), and we apply the Gram-Schmidt algorithm during each forward pass in order to satisfy the orthogonality.

$$< \mathbf{d_i}, \mathbf{d_j} >= \begin{cases} 0 & i \neq j \\ 1 & i = j. \end{cases} \tag{1}$$

Time Encoding. We decomposed the 'time-encoded component' $\mathbf{D}_t$ of the video representation from the latent space, to force the model to be aware of temporal variances along $\mathbf{D}_t$ with different time shifts. We propose to encode and parameterize the time shift values $\mathbf{dt}$ for the randomly selected query (and key) segment using their absolute starting point in seconds in the timestamps (*i.e.*, t_{start}). We used absolute time as we aimed at learning the representation of a single segment aware of time shift from a fixed 'reference view' (*i.e.*, the video beginning).

$$\epsilon(\mathbf{dt}, \mathbf{f}(q)) = \mathrm{MLP}\left([\mathrm{MLP}(t_{start}), \mathbf{f}(q)]\right). \tag{2}$$

Specifically, we encode $\mathbf{dt}$ into a high-dimensional vector $\epsilon(\mathbf{dt}, \mathbf{f}(q))$ by simple MLP (see Eq. 2), with the purpose of parameterizing the time shift considering different time-blending variants followed by contrastive learning. The time encoder also accepts $\mathbf{f}(q)$ as the input by concatenating with embedded $\mathbf{dt}$, towards learning a video specific encoding. We explore the idea of effective modeling for time shifts by proposing and comparing three time parameterization variants for the transformation from $\mathbf{f}$ to $\mathbf{f}'$. The first approach has to do with straightforward linear addition on video representation $\mathbf{f}(q)$ with $\epsilon(\mathbf{dt}, \mathbf{f}(q))$ (Variant 1). We then develop more efficient variants, which model the 'time-encoded component' more finely by learning the weights (Variant 2) or the magnitudes (Variant 3) only along the directions in the 'time-encoded component' $\mathbf{D}_t$.

Variant 1. Time-driven Linear Addition We implement $\epsilon(\mathbf{dt_q}, \mathbf{f}(q)) \in \mathbb{R}^{1\times dim}$ as the offset, from which positive pairs need to be pulled away from the representation 'time-encoded component' to obtain the time-blended representation in the latent space. The linear addition can be described as Eq. 3.

$$\mathbf{f}'(q, \mathbf{dt}_q) = \mathbf{f}(q) + \epsilon(\mathbf{dt_q}, \mathbf{f}(q)) \tag{3}$$

Variant 2. Time-driven Attention We then explicitly implement an attention mechanism to learn a set of attention weights for the positive pairs to be driven by $\mathbf{W} \in \mathbb{R}^{1\times M} = \{w_1, w_2, ..., w_M\} = \mathrm{Softmax}\left(\epsilon(\mathbf{dt_q}, \mathbf{f}(q))\right)$. The attention weights force $\mathbf{f}(q)$ to focus on the specific 'time-encoded component' in $\mathbf{D}_t$ according to different time encoding. This process can be described as follows

$$\mathbf{f}'(q, \mathbf{dt}_q) = \mathbf{f}(q) \cdot (\sum_{i=1}^{M} w_i \cdot \mathbf{d_i}). \tag{4}$$

Variant 3. Time-driven Linear Transformation. As shown in Fig. 2 (b), we finally propose a linear transformation method to encode the time shift information in the latent 'time-encoded component' $\mathbf{D}_t$. To implement linear transformation along $\mathbf{D}_t$, we learn the coefficient (*i.e.*, magnitude) on each direction of $\mathbf{D}_t$, noted as $\mathbf{A} \in \mathbb{R}^{1\times M} = \{a_1, a_2, ..., a_M\} = \epsilon(\mathbf{dt}_q, \mathbf{f}(q))$, by the time encoder. This linear transformation is able to enforce time variance and

Transformation	Top-1 (%)	Mean (%)
Base: w/o transformation	65.1	49.7
Variant 1: Linear w/o $\mathbf{D}_t$	66.0	49.8
Variant 2: Attention	66.7	51.6
Variant 3: Linear w/ $\mathbf{D}_t$		
w/o orthogonalization of $\mathbf{D}_t$	67.3	53.1
w/ orthogonalization of $\mathbf{D}_t$	**67.8**	**53.7**

Table 1: Top-1 accuracy and Mean accuracy on Smarthome CS in comparing proposed Time parameterization variants.

Method	P	Top-1 (%)	Mean (%)
MoCo (He et al. 2020)	2	61.5	47.2
ρMoCo (Feichtenhofer et al. 2021)	4	65.1	49.7
ρBYOL (Feichtenhofer et al. 2021)	4	61.7	42.4
LTN + MoCo	2	65.5	49.0
LTN + ρMoCo	4	**67.8**	**53.7**
LTN + ρBYOL	4	63.3	45.1

Table 2: Top-1 accuracy and Mean per-class accuracy on Smarthome CS signifying the impact of LTN on *different contrastive frameworks*. P: number of positive pairs.

to obtain different representations only along $\mathbf{D}_t$. The final time-blended representation $\mathbf{f}'(q, \mathbf{dt}_q)$ can be described as follows

$$\mathbf{f}'(q, \mathbf{dt}_q) = \mathbf{f}(q) + \sum_{i=1}^{M} a_i \cdot \mathbf{d_i} = \mathbf{f}(q) + \mathbf{A} \times \mathbf{D}_t^T. \quad (5)$$

All proposed time parameterization variants are effective in learning video representations aware of temporal changes and can improve the target downstream tasks by capturing such motion variances. Associated analysis is presented in Sec. Ablation Study, where we compare the three variants on their performance of downstream tasks. We find that the Linear Transformation with an orthogonal basis is the most effective and is beneficial as a generic methodology for learning time-aware spatio-temporal representations.

Self-supervised Contrastive Learning

In this section, we omit the parameterized time of all samples in the notations to simplified formulations (*e.g.*, $\mathbf{f}'(q, \mathbf{dt_q})$ is abbreviated as $\mathbf{f}'(q)$), and we provide details on the contrastive loss function. We apply general contrastive learning (see Fig. 2 (c)) to train our visual encoder $\mathbf{f}$ to encourage similarities between the time-blended positive representations, $\mathbf{f}'(q), \mathbf{f}'(k_1^+), ..., \mathbf{f}'(k_P^+)$, and discourage similarities between negative representations, $\mathbf{f}'(k_1^-), ..., \mathbf{f}'(k_N^-)$. The InfoNCE (Oord, Li, and Vinyals 2018) objective is defined as follows

$$\mathcal{L}_q = \sum_{p=1}^{P} \mathcal{L}_{NCE} = -\mathbb{E}\Big(\log \frac{\sum_{p=1}^{P} e^{\mathrm{Sim}\left(\mathbf{f}'(q), \mathbf{f}'(k_p^+)\right)}}{\sum_{n=1}^{N} e^{\mathrm{Sim}\left(\mathbf{f}'(q), \mathbf{f}'(k_n^-)\right)}} \Big), \quad (6)$$

where P represents the number of positive Keys, N denotes the number of negative Keys, and the similarity can be computed as:

$$\mathrm{Sim}(x, y) = \frac{\phi(x) \cdot \phi(y)}{\|\phi(x)\| \cdot \|\phi(y)\|} \cdot \frac{1}{Temp}, \quad (7)$$

#Layers	#Dimensions	Top-1 (%)	Mean (%)
None	-	65.1	49.7
1	128	66.7	50.5
1	1024	67.3	52.3
2	1024	67.1	52.8
2	2048	**67.8**	**53.7**
3	2048	67.9	53.2

Table 3: Top-1 accuracy and Mean per-class accuracy on Smarthome CS *w.r.t.Time Encoder*.

Size of $\mathbf{D}_t$ (M)	Top-1 (%)	Mean (%)
$M = 16$	65.2	51.6
$M = 64$	**67.8**	**53.7**
$M = 128$	67.3	52.2
$M = 512$	67.6	52.1
$M = 1024$	67.5	51.1
$M = 2000$	66.9	50.5

Table 4: Top-1 and Mean accuracy on Smarthome CS for study on number of directions in the orthogonal basis $\mathbf{D}_t$.

where $Temp$ refers to the temperature hyper-parameter (Wu et al. 2018), and ϕ is a learnable mapping function (*e.g.*, an MLP projection head (Feichtenhofer et al. 2021)) that can substantially improve the learned representations.

Experiments and Analysis

We conduct extensive experiments to evaluate LTN on four action classification datasets: **Toyota Smarthome**, **Kinetics-400**, **UCF101** and **HMDB51**. Firstly, we provide experimental results on tested variants, we investigate exhaustive ablations and further analyze on Toyota Smarthome (fine-grained action classification dataset) to better understand the design choices of our proposed time parameterization approaches. Secondly, we compare LTN with the best setting to state-of-the-art methods on all evaluated benchmarks. : Toyota Smarthome, UCF101 and HMDB51 without additional training data and with pre-training on Kinetics-400.

Ablation Study

As activities of Toyota Smarthome (Smarthome) are with similar motion and high duration variance (*e.g.*, 'Leave', 'Enter', 'Clean dishes', 'Clean up'), the temporal information is generally crucial for action classification. To understand the contribution of LTN for video representation learning, we conduct ablation experiments on Smarthome Cross-Subject (Das et al. 2019), with *linear evaluation* protocol (*i.e.*, pre-training without action labels, then training the classifiers only with the action labels) using RGB videos without additional modalities or training data. For the proposed $\mathbf{D}_t$, unless otherwise stated, we set $M = 64$ directions over the $dim = 2048$ dimensions. We report Top-1 and Mean per-class accuracy.

LTN Variants. The key module of LTN is the Time Parameterization method with three effective variants To study the impact of each variant, we start from a baseline using MoCo (He et al. 2020) with multiple positive samples $P = 4$ as (Feichtenhofer et al. 2021) and we then incorporate the time parameterization variants. The results in Tab. 1 indicate

Method	Supervision	Backbone	Mod.	Dataset	Frozen	Toyota Smarthome CS(%)	CV2(%)
From scratch	Supervised	R3D-50	V	SH	×	50.2	28.6
SimCLR (Chen et al. 2020)	Self-sup.	R3D-50	V	SH	✓	42.2	26.3
SwAV (Caron et al. 2020)	Self-sup.	R3D-50	V	SH	✓	41.4	25.6
MoCo (He et al. 2020)	Self-sup.	R3D-50	V	SH	✓	47.2	28.8
ρBYOL (Feichtenhofer et al. 2021)	Self-sup.	R3D-50	V	SH	✓	42.4	26.8
LTN (Ours)	Self-sup.	R3D-50	V	SH	✓	**53.7**	**30.1**
LTN (Ours)	Self-sup.	R3D-50	V	K400	✓	**54.5**	**35.5**
STA (Das et al. 2019)	Supervised	I3D+LSTM	V+P	K400	×	54.2	50.3
AssembleNet++ (Ryoo et al. 2020)	Supervised	R(2+1)D-50	V	K400	×	63.6	-
NPL (Piergiovanni and Ryoo 2021)	Supervised	R3D-50	V	K400	×	-	54.6
ImprovedSTA (Climent-Pérez and Florez-Revuelta 2021)	Supervised	I3D+LSTM	V+P	K400	×	63.7	53.6
VPN (Das et al. 2020)	Supervised	I3D+AGCNs	V+P	K400	×	60.8	53.5
MoCo (He et al. 2020)	Self-sup.	R3D-50	V	K400	×	61.8	52.7
LTN (Ours)	Self-sup.	R3D-50	V	K400	×	**65.9**	**54.6**

Table 5: Comparison of LTN to state-of-the-art methods on the Toyota Smarthome dataset (SH) with Cross-Subject (CS) and Cross-View2 (CV2) evaluation protocols. Mod: Modalities, V: RGB frames only, P: pre-extracted Pose data (skeleton keypoints coordinates), K400: the Kinetics-400 dataset. We classify methods *w.r.t.*supervision in the second column.

that leveraging time information is pertinent in improving the accuracy of fine-grained action classification. Specifically, in Variant 1, joint linear addition and visual representation related to time encoding without using $\mathbf{D}_t$ slightly boosts the Top-1 performance. We argue that the learned representation should code spatio-temporal data augmentation. If the entire representation is biased by time in the absence of $\mathbf{D}_t$, the static information that should be invariant is also shifted. This motivates us to use latent space decomposition to disentangle the 'time-encoded component' $\mathbf{D}_t$ coded in the learned representation. Using $\mathbf{D}_t$ to parameterize time encoding can significantly improve the performance (+1.9% by Variant 2 based on attention), especially by means of linear transformation (+4.0% by Variant 3).

Impact of LTN for Different Contrastive Models. We compare two state-of-the-art momentum-based contrastive models (Grill et al. 2020; He et al. 2020), a pair of positive samples (P=2) and the improved versions (Feichtenhofer et al. 2021) by leveraging multiple positive Keys (P=4) on the Smarthome dataset. Then, we incorporate the proposed LTN (Variant 3 with $M = 64$) into all models. The results in Tab. 2 demonstrate that LTN improves all three models and performs the best with ρ MoCo (Feichtenhofer et al. 2021) for our target downstream action classification task.

Design of Time Encoder. We explore how many directions are required in $\mathbf{D}_t$. We empirically test six different values for M from 64 to 2000. Quantitative results in Tab. 4 show that when using 64 directions (out of all dim=2048 directions), the model achieves the best action classification results. Hence, we set $M = 64$ for the other experiments. For the design of the proposed time encoder, we investigate the effect of different numbers of hidden layers and dimensions for the time encoder across five architectures. The results shown in Tab. 3 suggest that 2-layer MLP with 2048 dimensions in the hidden layer is the most effective.

Comparison with State-of-the-art

We first compare our method on Smarthome. As we are the firsts to conduct the self-supervised action classification task on this dataset using only RGB data, we re-implement state-of-the-art models (Caron et al. 2020; Chen et al. 2020; Feichtenhofer et al. 2021; Grill et al. 2020; He et al. 2020) and we compare the linear evaluation results without extra training data. We find that our proposed LTN, jointly with MoCo (He et al. 2020) achieves state-of-the-art performance, see Tab. 5. To further compare the results with skeleton-based methods (Climent-Pérez and Florez-Revuelta 2021; Das et al. 2020) trained with additional stream (Yang et al. 2021a,c), we conduct a self-supervised pre-training on Kinetics-400 and we transfer the model on Smarthome by linear evaluation and fine-tuning, see Tab. 5 bottom. In both settings, our model outperforms self-supervised state-of-the-art accuracy and many supervised approaches (Climent-Pérez and Florez-Revuelta 2021; Das et al. 2019, 2020; Piergiovanni and Ryoo 2021; Ryoo et al. 2020; Shi et al. 2019).

Method	Backbone	Mod.	K400 (%)
VTHCL (Yang et al. 2020)	R3D-50	V	37.8
CVRL (Qian et al. 2021)	R3D-50	V	66.1
SeCo (Yao et al. 2021)	R3D-50	V	61.9
MoCo (He et al. 2020)	R3D-50	V	66.6
ρBYOL (Feichtenhofer et al. 2021)	R3D-50	V	70.0
MCL (Li et al. 2021b)	R3D-50	V+F	66.6
LTN (Ours)	R3D-50	V	**71.3**

Table 6: Comparison with state-of-the-art methods on Kinetics-400 by *Linear evaluation*. Mod: Modalities, V: RGB frames only, F: pre-extracted optical flow.

We then compare our method to state-of-the-art approaches by linear evaluation on the general video understanding benchmark, Kinetics-400. For fair comparison, we mainly focus on the methods using R3D-50 and $T = 8$ sampled frames for training. The results are shown in Tab. 6 and demonstrate that, our LTN can improved upon previous methods (Feichtenhofer et al. 2021; He et al. 2020; Li et al. 2021b; Qian et al. 2021; Yang et al. 2020; Yao et al. 2021).

We also compare our LTN to state-of-the-art on HMDB51

Method	Backbone	Mod.	Data	Frozen	UCF (%)	HMDB (%)	Data	Frozen	UCF (%)	HMDB (%)
OPN (Lee et al. 2017)	VGG-M	V	-	✓	-	-	UCF	×	59.6	23.8
ClipOrder (Xu et al. 2019)	R(2+1)D	V	-	✓	-	-	UCF	×	72.4	30.9
CoCLR (Han, Xie, and Zisserman 2020)	S3D	V	UCF	✓	70.2	39.1	UCF	×	81.4	52.1
LTN (Ours)	R3D-50	V	UCF	✓	**71.8**	**40.3**	UCF	×	**81.6**	**52.8**
SpeedNet (Benaim et al. 2020)	S3D-G	V	-	✓	-	-	K400	×	81.1	48.8
VTHCL (Yang et al. 2020)	R3D-50	V	-	✓	-	-	K400	×	82.1	49.2
TaCo (Bai et al. 2020)	R3D-50	V	K400	✓	59.6	26.7	K400	×	85.1	51.6
MoCo (He et al. 2020)	R3D-50	V	-	✓	-	-	K400	×	92.8	67.5
CVRL (Qian et al. 2021)	R3D-50	V	-	✓	-	-	K400	×	92.2	66.7
ρBYOL (Feichtenhofer et al. 2021)	R3D-50	V	-	✓	-	-	K400	×	94.2	72.1
SeCo (Yao et al. 2021)	R3D-50	V	K400	✓	-	-	K400	×	88.3	55.6
CATE (Sun et al. 2021)	R3D-50	V	K400	✓	84.3	53.6	K400	×	88.4	61.9
CORP (Hu et al. 2021)	R3D-50	V	K400	✓	90.2	58.7	K400	×	93.5	68.0
FAME (Ding et al. 2022)	I3D	V	K400	✓	-	-	K400	×	88.6	61.1
LTN (Ours)	R3D-50	V	K400	✓	**90.6**	**58.9**	K400	×	**94.5**	**72.3**
CoCLR (Han, Xie, and Zisserman 2020)	S3D	V+F	K400	✓	77.8	52.4	K400	×	90.6	62.9
MCL (Li et al. 2021b)	R(2+1)D-50	V+F	-	✓	-	-	K400	×	93.4	69.1
BraVe (Recasens et al. 2021)	TSM-50x2	V+F+A	AudioS	✓	92.8	70.6	AudioS	×	96.5	79.3

Table 7: Comparison with state-of-the-art methods on UCF101 and HMDB51 with pre-training on Kinetics-400 (K400). Mod: Modalities, V: RGB frames only, F: pre-extracted optical flow, A: Audio.

and UCF101 (see Tab. 7). For fair comparison, we mainly focus on the model trained with the R3D-50 backbone used in our work with training frames $T = 8$. Using frozen features, our model outperforms all other works and even outperforms a number of works that adopt fine-tuning. For fine-tuning, the improvements are slight as the duration of these videos is small and the action is not as sensitive as Smarthome to time variance. However, we still outperform all previously single RGB-based models and our model performs competitively with current multi-modal methods (Han, Xie, and Zisserman 2020; Li et al. 2021b; Recasens et al. 2021) combining information from pre-extracted optical flow and audio.

Further Analysis

Per-class Comparison with State-of-the-art. We list the Smarthome classes that benefit the most and the least from LTN (see Tab. 8) compared to the state-of-the-art model (MoCo). We find that our method is able to effectively classify the fine-grained actions (*e.g.*, 'Cook.Usestove' +47.1%, 'Makecoffee.Boilwater' +31.8%, 'Laydown' +25.9%, 'Leave' +22.4%) while being challenged in distinguishing some object-oriented activities (*e.g.*, 'Drink.Fromglass' -28.3%, 'Drink.Fromcan': -14.2%). We believe that this is due to the fact that we focus on temporal modeling using time encoding, which only places emphasis on humans and ignores object information. To tackle this challenge and to further improve classification performance, future work will extend our method to latent spatial information (Sun et al. 2021) in order to capture the object information, while maintaining time awareness, which is still an open problem.

Representation Analysis. To demonstrate that the learned presentations are aware of temporal augmentations, we randomly select 2 videos ('Leave' and 'Enter') that are correctly classified by our model and uniformly sample 20 segments for each video. Then, we visualize their time-aware (learned by the proposed LTN) and time-invariant (learned by MoCo) representations respectively with t-SNE (see Appendix for visualization). We find that, unlike the time-invariant representations of uniformly sampled segments learned by previous model (He et al. 2020) that are only regrouped together, the time-blended representations learned by our LTN are well aligned over the time order. Hence we conclude that LTN can learn the consistent amount of temporal changes with the video segments on their time-aware representations to benefit fine-grained motion-focused action classification.

Activity	Gain from LTN (%)
Cook.Usestove	+47.08
Maketea.Boilwater	+31.78
Laydown	+25.88
Cutbread	+25.42
Leave	+22.43
Mean Accuracy	+6.97
Walk	-5.07
Usetablet	-11.30
Cook.Cleandishes	-12.74
Drink.Fromcan	-14.24
Drink.Fromglass	-28.25

Table 8: Activities that benefit the most and the least from LTN, and Mean per-class accuracy gain on Smarthome CS.

Conclusions

In this work, we present LTN, a temporal parameterization approach that learns time-aware action representation. We show that embedding time information of each video segment into the contrastive model by time navigation through a time encoder and an orthogonal basis can significantly improve the representation capability for videos. Experimental analysis confirms that a visual encoder extracting such representation can boost downstream action recognition. Future work will extend our time parameterization approach to spatial dimension, in order to better capture the object information that may also be crucial for fine-grained action recognition.

Acknowledgements

This work was supported by Toyota Motor Europe (TME) and the French government, through the 3IA Cote d'Azur Investments In the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002. This work was granted access to the HPC resources of IDRIS under the allocation AD011011627R1.

References

Bachman, P.; Hjelm, R. D.; and Buchwalter, W. 2019. Learning Representations by Maximizing Mutual Information Across Views. In *NeurIPS*.

Bai, Y.; Fan, H.; Misra, I.; Venkatesh, G.; Lu, Y.; Zhou, Y.; Yu, Q.; Chandra, V.; and Yuille, A. 2020. Can Temporal Information Help with Contrastive Self-Supervised Learning? In *arXiv:2011.13046*.

Benaim, S.; Ephrat, A.; Lang, O.; Mosseri, I.; Freeman, W. T.; Rubinstein, M.; Irani, M.; and Dekel, T. 2020. SpeedNet: Learning the Speediness in Videos. In *CVPR*.

Bruno, K.; Du, T.; and Lorenzo, T. 2019. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*.

Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2020. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *NeurIPS*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*.

Climent-Pérez, P.; and Florez-Revuelta, F. 2021. Improved Action Recognition with Separable Spatio-Temporal Attention Using Alternative Skeletal and Video Pre-Processing. *Sensors*.

Das, S.; Dai, R.; Koperski, M.; Minciullo, L.; Garattoni, L.; Bremond, F.; and Francesca, G. 2019. Toyota Smarthome: Real-World Activities of Daily Living. In *ICCV*.

Das, S.; Dai, R.; Yang, D.; and Bremond, F. 2021. VPN++: Rethinking Video-Pose embeddings for understanding Activities of Daily Living. *IEEE TPAMI*.

Das, S.; Sharma, S.; Dai, R.; Bremond, F.; and Thonnat, M. 2020. VPN: Learning Video-Pose Embedding for Activities of Daily Living. In *ECCV*.

Diba, A.; Sharma, V.; Van Gool, L.; and Stiefelhagen, R. 2019. DynamoNet: Dynamic Action and Motion Network. In *ICCV*.

Ding, S.; Li, M.; Yang, T.; Qian, R.; Xu, H.; Chen, Q.; Wang, J.; and Xiong, H. 2022. Motion-aware Contrastive Video Representation Learning via Foreground-background Merging. In *CVPR*.

Dwibedi, D.; Aytar, Y.; Tompson, J.; Sermanet, P.; and Zisserman, A. 2019. Temporal Cycle-Consistency Learning. In *CVPR*.

Feichtenhofer, C.; Fan, H.; Xiong, B.; Girshick, R.; and He, K. 2021. A Large-Scale Study on Unsupervised Spatiotemporal Representation Learning. In *CVPR*.

Goyal, R.; Kahou, S. E.; Michalski, V.; Materzynska, J.; Westphal, S.; Kim, H.; Haenel, V.; Fruend, I.; Yianilos, P.; Mueller-Freitag, M.; Hoppe, F.; Thurau, C.; Bax, I.; and Memisevic, R. 2017. The "Something Something" Video Database for Learning and Evaluating Visual Common Sense. In *ICCV*.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. A.; Guo, Z. D.; Azar, M. G.; Piot, B.; Kavukcuoglu, K.; Munos, R.; and Valko, M. 2020. Bootstrap your own latent: A new approach to self-supervised Learning. In *NeurIPS*.

Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *CVPR*.

Han, T.; Xie, W.; and Zisserman, A. 2020. Self-supervised Co-training for Video Representation Learning. In *NeurIPS*.

Hara, K.; Kataoka, H.; and Satoh, Y. 2017. Learning Spatio-Temporal Features with 3D Residual Networks for Actio Recognition. In *ICCVW*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*.

Hjelm, R. D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; and Bengio, Y. 2019. Learning deep representations by mutual information estimation and maximization. In *ICLR*.

Hu, K.; Shao, J.; Liu, Y.; Raj, B.; Savvides, M.; and Shen, Z. 2021. Contrast and Order Representations for Video Self-Supervised Learning. In *ICCV*.

Huang, J.; Dong, Q.; Gong, S.; and Zhu, X. 2019. Unsupervised Deep Learning by Neighbourhood Discovery. In *ICML*.

Huang, L.; Liu, Y.; Wang, B.; Pan, P.; Xu, Y.; and Jin, R. 2021. Self-Supervised Video Representation Learning by Context and Motion Decoupling. In *CVPR*.

Jiao, Y.; Xiong, Y.; Zhang, J.; Zhang, Y.; Zhang, T.; and Zhu, Y. 2020. Sub-graph Contrast for Scalable Self-Supervised Graph Representation Learning. In *ICDM*.

Kim, D.; Cho, D.; and Kweon, I. S. 2019. Self-Supervised Video Representation Learning with Space-Time Cubic Puzzles. In *AAAI*.

Kong, Q.; Wei, W.; Deng, Z.; Yoshinaga, T.; and Murakami, T. 2020. Cycle-Contrast for Self-Supervised Video Representation Learning. In *NeurIPS*.

Lee, H.-Y.; Huang, J.-B.; Singh, M.; and Yang, M.-H. 2017. Unsupervised Representation Learning by Sorting Sequences. In *ICCV*.

Li, L.; Wang, M.; Ni, B.; Wang, H.; Yang, J.; and Zhang, W. 2021a. 3D Human Action Representation Learning via Cross-View Consistency Pursuit. In *CVPR*.

Li, R.; Zhang, Y.; Qiu, Z.; Yao, T.; Liu, D.; and Mei, T. 2021b. Motion-Focused Contrastive Learning of Video Representations. In *ICCV*.

Li, T.; Liu, J.; Zhang, W.; Ni, Y.; Wang, W.; and Li, Z. 2021c. UAV-Human: A Large Benchmark for Human Behavior Understanding With Unmanned Aerial Vehicles. In *CVPR*.

Mathieu, M.; Couprie, C.; and LeCun, Y. 2016. Deep multi-scale video prediction beyond mean square error. In *ICLR*.

Misra, I.; and van der Maaten, L. 2020. Self-Supervised Learning of Pretext-Invariant Representations. In *CVPR*.

Misra, I.; Zitnick, C. L.; and Hebert, M. 2016. Shuffle and Learn: Unsupervised Learning using Temporal Order Verification. In *ECCV*.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation Learning with Contrastive Predictive Coding. In *arXiv:1807.03748*.

Park, J.; Lee, J.; Kim, I.-J.; and Sohn, K. 2022. Probabilistic Representations for Video Contrastive Learning. In *CVPR*.

Patrick, M.; Asano, Y. M.; Huang, B.; Misra, I.; Metze, F.; Henriques, J.; and Vedaldi, A. 2021. Space-Time Crop & Attend: Improving Cross-modal Video Representation Learning. In *ICCV*.

Piergiovanni, A.; and Ryoo, M. S. 2021. Recognizing Actions in Videos From Unseen Viewpoints. In *CVPR*.

Qian, R.; Meng, T.; Gong, B.; Yang, M.-H.; Wang, H.; Belongie, S.; and Cui, Y. 2021. Spatiotemporal Contrastive Video Representation Learning. In *CVPR*.

Ranasinghe, K.; Naseer, M.; Khan, S.; Khan, F. S.; and Ryoo, M. 2022. Self-supervised Video Transformer. In *CVPR*.

Recasens, A.; Luc, P.; Alayrac, J.-B.; Wang, L.; Strub, F.; Tallec, C.; Malinowski, M.; Pătrăucean, V.; Altché, F.; Valko, M.; Grill, J.-B.; van den Oord, A.; and Zisserman, A. 2021. Broaden Your Views for Self-Supervised Video Learning. In *ICCV*.

Ryoo, M.; Piergiovanni, A.; Kangaspunta, J.; and Angelova, A. 2020. AssembleNet++: Assembling Modality Representations via Attention Connections. In *ECCV*.

Shi, L.; Zhang, Y.; Cheng, J.; and Lu, H. 2019. Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition. In *CVPR*.

Shuang, M.; Zhaoyang, Z.; Daniel, M.; and Yale, S. 2021. Active contrastive learning of audio-visual video representations. In *ICLR*.

Srivastava, N.; Mansimov, E.; and Salakhutdinov, R. 2015. Unsupervised Learning of Video Representations using LSTMs. In *ICML*.

Sun, C.; Nagrani, A.; Tian, Y.; and Schmid, C. 2021. Composable Augmentation Encoding for Video Representation Learning. In *ICCV*.

Tian, Y.; Krishnan, D.; and Isola, P. 2020. Contrastive Multiview Coding. In *ECCV*.

Vondrick, C.; Pirsiavash, H.; and Torralba, A. 2016a. Anticipating Visual Representations from Unlabeled Video. In *CVPR*.

Vondrick, C.; Pirsiavash, H.; and Torralba, A. 2016b. Generating Videos with Scene Dynamics. In *NeurIPS*.

Vondrick, C.; Shrivastava, A.; Fathi, A.; Guadarrama, S.; and Murphy, K. 2018. Tracking Emerges by Colorizing Videos. In *ECCV*.

Wang, Y.; Yang, D.; Bremond, F.; and Dantcheva, A. 2022. Latent Image Animator: Learning to Animate Images via Latent Space Navigation. In *ICLR*.

Wei, D.; Lim, J.; Zisserman, A.; and Freeman, W. T. 2018. Learning and Using the Arrow of Time. In *CVPR*.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised Feature Learning via Non-parametric Instance Discrimination. In *CVPR*.

Xiaolong, W.; Allan, J.; and Alexei A, E. 2019. Learning correspondence from the cycle-consistency of time. In *CVPR*.

Xu, D.; Xiao, J.; Zhao, Z.; Shao, J.; Xie, D.; and Zhuang, Y. 2019. Self-Supervised Spatiotemporal Learning via Video Clip Order Prediction. In *CVPR*.

Yang, C.; Xu, Y.; Dai, B.; and Zhou, B. 2020. Video Representation Learning with Visual Tempo Consistency. In *arXiv:2006.15489*.

Yang, D.; Dai, R.; Wang, Y.; Mallick, R.; Minciullo, L.; Francesca, G.; and Bremond, F. 2021a. Selective Spatio-Temporal Aggregation Based Pose Refinement System: Towards Understanding Human Activities in Real-World Videos. In *WACV*.

Yang, D.; Wang, Y.; Dantcheva, A.; Garattoni, L.; Francesca, G.; and Bremond, F. 2021b. Self-Supervised Video Pose Representation Learning for Occlusion-Robust Action Recognition. In *FG*.

Yang, D.; Wang, Y.; Dantcheva, A.; Garattoni, L.; Francesca, G.; and Bremond, F. 2021c. UNIK: A Unified Framework for Real-world Skeleton-based Action Recognition. In *BMVC*.

Yang, D.; Wang, Y.; Dantcheva, A.; Garattoni, L.; Francesca, G.; and Bremond, F. 2022. ViA: View-invariant Skeleton Action Representation Learning via Motion Retargeting. *arXiv:2209.00065*.

Yao, T.; Zhang, Y.; Qiu, Z.; Pan, Y.; and Mei, T. 2021. SeCo: Exploring Sequence Supervision for Unsupervised Representation Learning. In *AAAI*.