Self-Supervised Video Pose Representation Learning for Occlusion-Robust Action Recognition

Di Yang, Yaohui Wang, Antitza Dantcheva, Lorenzo Garattoni, Gianpiero Francesca, Francois F Bremond

Self-Supervised Video Pose Representation Learning for Occlusion-Robust Action Recognition

Di Yang[1,2] Yaohui Wang[1,2] Antitza Dantcheva[1,2] Lorenzo Garattoni[3]
Gianpiero Francesca[3] François Brémond[1,2]

[1]Inria [2]Université Côte d'Azur [3]Toyota Motor Europe
{di.yang, yaohui.wang, antitza.dantcheva, françois.brémond }@inria.fr
{lorenzo.garattoni, gianpiero.francesca}@toyota-europe.com

***Abstract*— Action recognition based on human pose has witnessed increasing attention due to its robustness to changes in appearances, environments, and view-points. Despite associated progress, one remaining challenge has to do with occlusion in real-world videos that hinders the visibility of all joints. Such occlusion impedes representation of such scenes by models that have been trained on full-body pose data, obtained in laboratory conditions with specific sensors. To address this, as a first contribution, we introduce OR-VPE, a novel video pose embedding network that is streamlined to learn an occlusion-robust representation for pose sequences in videos. In order to enable our embedding network to handle partially visible joints, we propose to incorporate a sub-graph data augmentation mechanism during training, which simulates occlusions, into a video pose encoder based on Graph Convolutional Networks (GCNs). As a second contribution, we apply a contrastive learning module to train the video pose representation in a self-supervised manner without the necessity of action annotations. This is achieved by minimizing the mutual information of the same pose sequence pruned into different spatio-temporal sub-graphs. Experimental analyses show that compared to training the same encoder from scratch, our proposed OR-VPE, with pre-training on a large-scale dataset, NTU-RGB+D 120, improves the performance of the downstream action classification on Toyota Smarthome, N-UCLA and Penn Action datasets.**

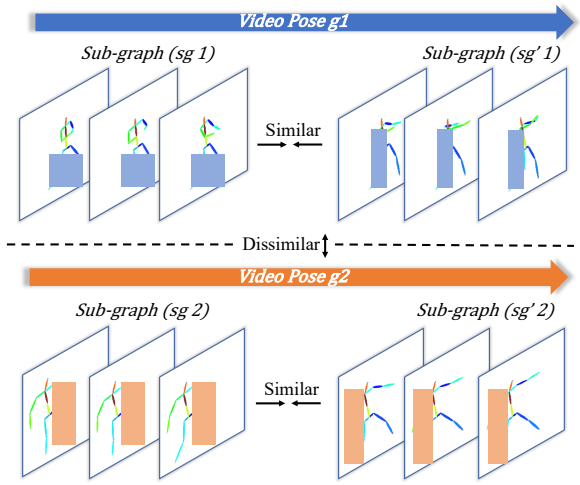


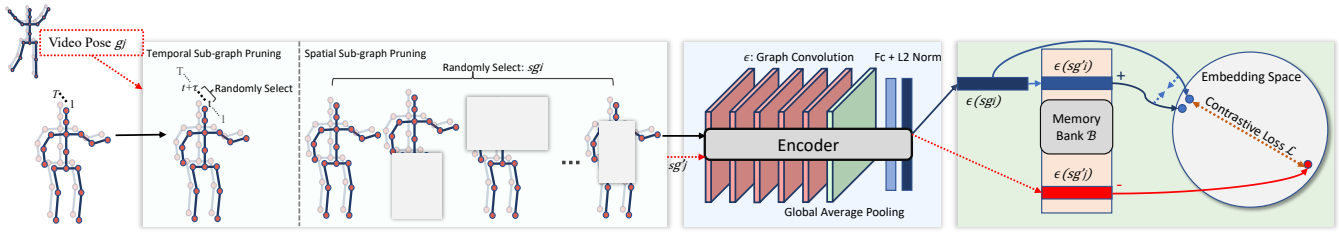
Fig. 1. **Self-supervised Video Pose Embedding.** The proposed approach enforces the network to learn the high representational similarity between the same instance (*e.g.,* video pose g_1) pruned into different sub-graphs (*e.g.,* sg_1, sg_1'). Meanwhile, it also follows the same mechanism as previous instance discrimination task [40] which distinguishes individual instances according to the visual cues.

I. Introduction

Recently, action recognition based on human pose in videos (*i.e.* video pose) has shown promising classification accuracy using high-quality human pose data obtained from Kinect sensors [46]. Such approaches are able to filter out noise caused by background clutter and changing light conditions, while maintaining the focus on the performed action [35], [33], [47], [41], [17], [2], [42], [18], [30], [21], [44]. However, we note that in named works the sensor pose data has generally been captured in lab environments, and hence may not contain occlusions. Therefore, accuracy of named approaches significantly decreases, when the pre-trained models are tested with real-world videos where, pose data encounters a number of occlusions and truncations of the body. Models, generalizing onto real-world videos, necessitate the ability to represent partially visible body joints. At the same time, it is infeasible in practice to train an individual embedding model for each possible occlusion pattern, as there exists large occlusion diversity in natural human poses.

Motivated by the above, in this paper, we propose to pre-train a single occlusion-robust video pose embedding model, namely OR-VPE, by simulating joint occlusions during training. Specifically, we apply Graph Convolutional Networks (GCNs) [30] as the embedding backbone that models human joints, as well as their natural connections (*i.e.* bones) in skeleton (*i.e.* pose) spatio-temporal graphs. In each training epoch, a novel spatio-temporal sub-graph data augmentation strategy, namely SubG-DA, is performed by randomly selecting a sub-graph for the same instance (*e.g.,* lower-body, upper-body, etc.) that simulates different occlusion patterns. This mechanism endows our model with robustness to common patterns of missing joints in the real-world. Nevertheless, learning a supervised video pose embedding demands a huge number of annotations, which in turn encourages researchers to investigate self-supervised learning schemes to leverage the massive amount of unlabeled videos [9], [39]. Hence in this paper, we additionally propose a self-supervised learning module using sub-graph contrast (see Fig. 1) to pre-train the embedding model.

Contributions: (i) We propose OR-VPE, a novel and generic video pose embedding model that embeds different pose visibility patterns with a GCN-based video pose encoder



(a) Sub-graph Data Augmentation (SubG-DA) (b) Video Pose Encoder - AGCNs (c) Contrastive Module

Fig. 2. **Overview of OR-VPE.** Our framework includes three main components: Given an input video pose g_i represented by a spatio-temporal graph, the (a) SubG-DA randomly prunes g_i into a sub-graph sg_i in both spatial and temporal dimensions, to make the input pose sequence have a large occlusion diversity. Then the (b) AGCN-based encoder embeds sg_i to $\epsilon(sg_i)$ in a low-dimensional latent space by graph convolutions. Finally, to make the embedding space occlusion-invariant, the (c) Contrastive Module builds different sub-graph embeddings $\epsilon(sg'_i)$ with the same pose sequence (*i.e.* instance) in the previous training epoch sharing high similarity in terms of their semantics while being dissimilar to other embeddings $\epsilon(sg'_j)$. This is achieved by the contrastive loss $\mathcal{L}$. Note that all the embeddings in the previous epoch are stored in a memory bank $\mathcal{B}$.

and a novel sub-graph data augmentation strategy to improve the robustness of video pose representation to occlusions. (ii) We apply a contrastive learning module to learn OR-VPE in a self-supervised manner without using action labels. (iii) We demonstrate that our embedding model pre-trained on NTU-RGB+D 120 dataset generalizes well onto unseen real-world videos with additional fine-tuning, and boosts the accuracy compared to the same model specifically trained only on the given dataset.

II. Related Work

a) Self-supervised Human Pose Embedding: To explore the embedding (*i.e.* representation) ability of human pose for action recognition, a recent method named Pr-ViPE [32] incorporated probabilistic embedding to address inherent ambiguities in 2D pose due to 3D-to-2D projection based on triplet loss [29]. The pre-training was performed on the Human3.6M [14] dataset, which has multi-view poses from a motion capture system. However, Pr-ViPE [32] is for single 2D pose embedding, it necessitates multi-view 2D poses or 3D pose for training. In contrast, our work focuses on occlusion diversity in real-world videos and proposed OR-VPE can be beneficial in both, 2D and 3D pose embedding in sequence level, in the absence of multi-view data.

b) Graph Convolutional Networks for Video Pose: ST-GCN [42] involves spatial graph convolutions along with interleaving temporal convolutions for pose-based action recognition. In this context, fixed topology of the human pose was considered, however ignored the important long-range dependencies between unconnected joints. In contrast, recent approaches based on Adaptive Graph Convolutional Networks (AGCNs) [18], [30], [11], [31], [21] have seen significant performance boost, by improving the representation of topology of human poses to process long-range dependencies between joints for action recognition. In particular, 2s-AGCN [30] introduced an adaptive graph convolutional network to adaptively learn the topology of the graph with self-attention, which was shown beneficial in action recognition and hierarchical structure of GCNs. In addition, 2s-AGCN used a two-stream ensemble with pose bone and joint features to boost performance. Previous GCN-based work only focused on the performance in a given dataset by a supervised training manner. In this work, we are the first to train an AGCN-based embedding in a self-supervised manner to analyze the transfer behavior of the learned representation across datasets.

c) Contrastive Learning: Owing to their promising performances, contrastive learning and its variants [40], [13], [1], [34], [12], [6], [16] has established itself as an important direction for self-supervised representation learning. Particularly, related work [16] is predominantly based on sub-graph contrastive learning to learn graph representations by utilizing the strong correlation between central nodes and their pruned sub-graphs, in order to capture regional structure information. In our work, we apply the graph representation learning method on human pose sequence in videos for real-world pose-based action recognition. With the proposed data augmentation strategy, the graph convolutional networks learn the graph representations through a contrastive loss defined based on sub-graphs pruned from the original graph.

III. Proposed Approach

In this section, we introduce the proposed framework including three main components (see Fig. 2): (a) Sub-graph Data Augmentation (SubG-DA), (b) Video Pose Encoder and (c) Contrastive Module. In the pre-training phase, given an input video, we can obtain the human pose sequence through the off-the-shelf pose estimators [27], [4], [10], [43], which can be represented as a spatio-temporal graph [42]. The graph is fed into SubG-DA to have a large occlusion diversity by randomly pruning the spatio-temporal structure. Then the encoder embeds the poses into a low-dimensional latent space. Finally, a Contrastive Module is leveraged to render the latent space occlusion-invariant through contrastive learning, minimizing the mutual information between positive and negative samples. This pre-trained pose-based video representation can be transferred to other real-world video datasets containing occlusions to further improve the action recognition performance.

A. Sub-graph Based Data Augmentation

a) Graph Modeling for Pose Sequence: For each video v, we first estimate human 2D or 3D pose sequences (*i.e.* video pose) as the input of the model. As shown

in Fig. 2, the input video pose is modeled by a spatio-temporal graph [42], referred to as g, where the joints are represented as vertices and their natural connections in the human body are represented as spatial edges. For the temporal dimension, the corresponding joints in two consecutive frames are connected with temporal edges. T, V, and C represent the length of the video, the number of joints of the pose in one frame, as well as the input channels, respectively. The input video pose graph is represented by the matrix $g \in \mathbb{R}^{C\times T\times V}$, and an adjacency matrix $\mathbf{A} \in \mathbb{R}^{V\times V}$, respectively. More specifically, $\mathbf{A}(p,q) = 1$, if joint p and joint q are connected and $\mathbf{A}(p,q) = 0$, otherwise. The final adjacency matrix $\mathbf{A^{norm}} \in \mathbb{R}^{V\times V}$ is normalized using a degree matrix $\mathbf{\Lambda} \in \mathbb{R}^{V\times V}$ as:

$$\mathbf{\Lambda}(p,q) = \sum_{r=1}^{V} \mathbf{A}(p,r); \quad \mathbf{A^{norm}} = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{A}\mathbf{\Lambda}^{\frac{1}{2}} \tag{1}$$

b) Spatio-temporal Sub-graph Pruning: In the spatial dimension, we use a binary mask vector $\mathbf{O}$ to represent the visibility of each joint for g, where each entry, the embedding (*i.e.* OR-VPE) of g is 1, if its corresponding joint is visible and 0 otherwise. This visibility indicator can represent whether a joint is not visible due to occlusion. Different joint masks can be used for simulating diverse realistic occlusion patterns. We multiply the mask with normalized pose, which then constitutes the model input. In order to enable the model to be robust to masked pose joints, we generate various patterns of occlusion during training. It is ideal to train our model on various realistic joint occlusion patterns. However, datasets in practice may not contain diverse occlusion patterns, so we address this by simulating occlusion patterns and using the simulated patterns for training. In each training epoch, we randomly select a segment of τ frames from the input video pose g as a temporal sub-graph, and then we use different pre-defined occlusion masks $\mathbf{O}$ to prune the body into multiple spatial sub-graphs (*e.g.,* upper-body, lower-body, left-body), and the one fed into the video pose encoder (see III-B) is randomly selected as the augmented input data.

B. Video Pose Encoder

a) Adaptive Graph Convolutional Networks: we adopt AGCNs [30] as the backbone. $\epsilon(g) \in \mathbb{R}^{e_{out}}$, denotes the embedding features where e_{out} is the representation dimension. The graph convolutional layers can be formulated as:

$$\epsilon(g) = \sigma\Big(\mathbf{E}\big((\mathbf{A^{norm}} \odot \mathbf{M})g\big)\Big), \tag{2}$$

where $\mathbf{M}$ denotes a self-attention mask for the adaptive re-weighting of the $\mathbf{A^{norm}}$ to different actions, and $\odot$ denotes the Hadamard product. The operation of $\big((\mathbf{A^{norm}} \odot \mathbf{M})g\big)$ enforces the features of the body joints to be extracted following the adaptive skeleton topology learned from $\mathbf{A}$. $\mathbf{E}$ denotes a 1×1 convolutional layer to expand the feature dimension and σ is a non-linear activation layer (*e.g.,* ReLU).

C. Contrastive Module

a) Sub-graph Contrastive Learning: We apply contrastive learning to train our encoder ϵ. Specifically, in each training iteration, given a set of video poses $g = \{g_1, ..., g_n\}$, the i-th instance is pruned into a spatio-temporal sub-graph $sg_i = \mathbf{O} \odot g_i$ by a randomly selected occlusion mask $\mathbf{O}$ and we denote its sub-graph in the previous iteration that $sg'_i = \mathbf{O'} \odot g_i$. We can obtain their corresponding representations $\mathcal{E} = \{\epsilon(sg_1), ..., \epsilon(sg_n)\}$ and $\mathcal{E}' = \{\epsilon(sg'_1), ..., \epsilon(sg'_n)\}$, where we refer to $\epsilon(sg_i)$ and $\epsilon(sg'_i)$ as the sub-graph representations of the i-th instance. Learning the embedding based on the sub-graph contrast involves two mechanisms. For each $\epsilon(sg_i)$, we encourage the similarity between $\epsilon(sg_i)$ and its another sub-graph representation counterpart pruned from the same instance $\epsilon(sg'_i)$, while decreasing the similarities between $\epsilon(sg_i)$ and sub-graph representations from other instances $\epsilon(sg'_j)$ (*i.e.* negative samples). Subsequently, we can obtain the contrastive loss function:

$$\mathcal{L} = -\sum_{i=1}^{N}\left[\log \frac{\mathrm{sim}\left(\epsilon(sg_i), \epsilon(sg'_i)\right)}{\sum_{j=1}^{K}\mathrm{sim}\left(\epsilon(sg_i), \epsilon(sg'_j)\right)}\right], \tag{3}$$

where N represents the number of the instances, K denotes the number of the negative samples, and the similarity is computed as:

$$\mathrm{sim}(\epsilon_1, \epsilon_2) = \exp\big(\frac{\phi(\epsilon_1)\cdot\phi(\epsilon_2)}{\|\phi(\epsilon_1)\|\cdot\|\phi(\epsilon_2)\|}\cdot\frac{1}{Temp}\big), \tag{4}$$

where $Temp$ refers to the temperature hyper-parameter [40], and ϕ is a learnable mapping. As suggested by Chen et al. [6], applying a non-linear mapping function can substantially improve the learned representations.

b) Memory Bank: As it is non-trivial to extract all video pose features in a single batch at each iteration, we maintain a memory bank $\mathcal{B}$ of size $N \times e_{out}$ to reduce the computation overhead as [40]. $\mathcal{B}$ stores the approximated representations of video poses, which are accumulated over iterations as:

$$\epsilon_{bank} = m\epsilon_{bank} + (1-m)\epsilon_{current}, \tag{5}$$

where ϵ can be any $\epsilon(sg)$ or $\epsilon(sg')$, and $m \in [0, 1]$ is the momentum coefficient to ensure smoothness and stability. Based on $\mathcal{B}$, the learning process thus comprises of taking a mini-batch of video pose representations in current training epoch $\epsilon(sg)$ as queries, computing the loss function $\mathcal{L}$ based on their positive representations $\epsilon(sg')$ and N other representations stored in $\mathcal{B}$. Note that one can further reduce the computation overhead by sampling K representations from each bank rather than using the entire bank, when computing $\mathcal{L}$, or adopting noise contrastive estimation as in the works of Wu et al. [40].

IV. Experiments and Analysis

A. Experimental Settings

We conduct experiments to evaluate the representation ability of video pose embedding. Firstly, we train a single OR-VPE on a large-scale dataset, NTU-RGB+D 120 [19] (NTU-120) without using action labels, then we transfer the

TABLE I

RESULTS ON SMARTHOME, PENN ACTION AND N-UCLA WITH (TRANSFER LEARNING) AND WITHOUT EMBEDDING PRE-TRAINED ON NTU-120.

Methods	**Toyota Smarthome**				**Penn Action**		**N-UCLA**	
	#Params	CS(%)	CV1(%)	CV2(%)	#Params	Top-1 Accuracy (%)	#Params	CV (%)
Baseline (Linear classification w/o Embedding)	7.97K	23.1	15.8	19.3	3.85K	28.5	2.57K	35.6
Linear classification with Self-supervised Embedding (Ours)	7.97K	**42.7**	**18.1**	**32.4**	3.85K	**78.5**	2.57K	**56.7**
Baseline (AGCNs w/o Embedding)	3.45M	55.7	21.6	53.3	3.45M	77.2	3.45M	78.2
Fine-tuned with Self-supervised Embedding (Ours)								
with temporal SubG-DA only	3.45M	55.8	22.1	54.4	3.45M	78.8	3.45M	78.9
with spatial-temporal SubG-DA	3.45M	**56.3**	**24.6**	**59.0**	3.45M	**93.3**	3.45M	**84.5**
Fine-tuned with Supervised Embedding	3.45M	**58.2**	**27.3**	58.5	3.45M	90.7	3.45M	**87.6**

pre-trained embedding onto three downstream action classification datasets, namely Toyota Smarthome (Smarthome) [7], Northwestern-UCLA [37] (N-UCLA) and Penn Action [45]. For the downstream action classification task, we incorporate additional linear classifiers for different datasets. We demonstrate that the pre-trained Embedding can boost the performance of the video pose encoder (AGCNs) compared to training from scratch. See SM for more datasets details and the implementation details.

B. Pre-training Dataset

a) NTU-RGB+D 120: NTU-120 [19] is a large-scale multi-modality dataset, which consists of 114,480 sequences of high-quality 2D and 3D poses with 25 joints, associated with depth maps, RGB and IR frames captured by the Microsoft Kinect v2 sensor. There are 120 action classes performed in the laboratory. We only use sequences of 2D and 3D poses in this work for pre-training two embeddings for downstream tasks using 2D or 3D data.

C. Evaluation Datasets

a) Toyota Smarthome: Smarthome [7] is a real-world daily living dataset for action classification, recorded in an apartment, where 18 older subjects carry out tasks of daily living during a day. The dataset contains 16,115 videos of 31 action classes, and the videos are taken from 7 different camera viewpoints. All actions are performed in a natural way without strong prior instructions. It provides RGB videos and pose data, which is extracted from SSTA-PRS [43] (*i.e.* skeleton-v2). In this work, we use the 2D pose data only for all experiments and comparisons. We follow the cross-subject (CS) and cross-view (CV1 and CV2) protocols.

b) Penn Action: Penn Action dataset [45] contains 2,326 real-world video sequences of 15 different actions and human joint annotations for each sequence. Given that annotated 2D poses have a large number of missing joints due to occlusions and truncations. We report Top-1 accuracy following the standard train-test split.

c) Northwestern-UCLA: N-UCLA [37] (N-UCLA) is a multi-view activity 3D dataset acquired simultaneously by 3 Kinect v1 sensors. It consists of 1,194 video samples with 10 activity classes. The activities were performed by 10 subjects, and recorded from three viewpoints. We perform experiments on N-UCLA using the cross-view (CV) protocol: we train our model on samples from camera view1 and view2, then test on the samples from the remaining camera view3.

D. Implementation Details

a) Settings of SubG-DA: We pre-define 6 spatial sub-graph patterns (*i.e.* masks $\mathbf{O}$ in III-A) to simulate the different kinds of occlusions: (1) upper-body invisible, (2) lower-body invisible, (3) right-body invisible, (4) left-body invisible, (5) center-body invisible and (6) full-body visible. In the temporal domain, we select $\tau = 150$ (see III-A) for NTU-120. Subsequently, we randomly prune a sub-graph for each instance in different training epochs to have a data augmentation.

b) Settings of AGCNs: At the end of the AGCNs, an global average pooling layer and a fully connected layer with L2-normalization [40] are placed to embed the features into the lower dimension $e_{out} = 128$ (see III-B). The fully connected layer can be changed when transferred to downstream tasks. Unless otherwise stated in the ablation study, for all OR-VPE models, We use SGD for training with momentum 0.9, an initial learning rate of 0.1 for $60, 50, 30$, and 50 epochs with step LR decay with a factor of 0.1 at epochs $\{30, 50\}, \{30, 40\}, \{10, 20\}$, and $\{30, 40\}$ for NTU-120, Smarthome, Penn Action, and N-UCLA, respectively. Weight decay is set to 0.0001 for final models. 2D (*e.g.,* Smarthome and Penn Action) and 3D (*e.g.,* NTU-120 and N-UCLA) inputs are pre-processed following [25] and [30] respectively. Note that we convert the human structures on different datasets into a unified graph with unified joint number and order for model transferring.

c) Settings of Contrastive Module: For the Contrastive module, we set $K = 4096$ and $Temp = 0.07$ for computing the contrastive loss $\mathcal{L}$ and $m = 0.5$ for the memory bank (see III-C).

E. Ablation Study

a) Impact of Video Pose Embedding: We conduct the ablation study on all three evaluation datasets using joint data only for the analysis of impact of the pre-trained Embedding. We compare classification accuracy with the same video pose encoder (AGCNs) with and without pre-training the representation on NTU-120 dataset. Results in Tab. I demonstrate the impact of the embedding (OR-VPE) in both (i) linear classification (*i.e.* unsupervised domain adaptation) without additional fine-tuning for the backbone encoder and (ii) fine-tuning with additional re-training for the backbone encoder. From the training curves (see Fig. 3), we deduce that at the beginning of training steps, the embedding has a significant

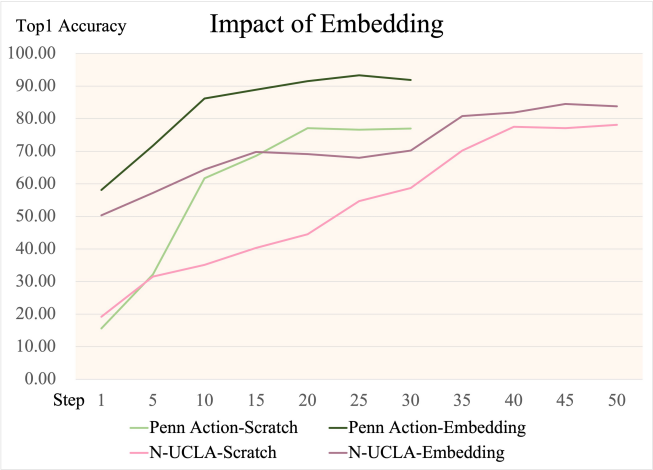


Fig. 3. Validation accuracy with the training steps on Penn Action and N-UCLA datasets without (Scratch) and with Embedding pre-trained on NTU-120. It demonstrates that the learned video pose representation can improve the downstream action classification on targeting benchmarks.

TABLE II

TOP-1 CLASSIFICATION ACCURACY COMPARISON WITH STATE-OF-THE-ART METHODS ON THE PENN ACTION DATASET.

Methods	**Data**	**Penn Action** Accuracy(%)
Nie et al. [24]	RGB+Pose	85.5
Cao et al. [3]	RGB+Pose	98.1
Liu et al. [20]	RGB+Pose	91.4
Luvizon et al. [22]	RGB+Pose	98.7
Late fusion: I3D [5]+**OR-VPE (Ours)**	RGB+Pose	96.3
Iqbal et al. [15]	Pose	79.0
MS-G3D Net [21]	Pose	92.5
2s-AGCN [30] (Baseline)	Pose	93.1*
OR-VPE (Ours)	Pose	**94.0**

boost. This suggests that the video pose representation is well pre-trained on NTU-120, providing a strong transfer ability. See SM for comparison with supervised pre-training.

b) Impact of Sub-graph Data Augmentation: We pre-train our video pose encoder with and without the SubG-DA, the result in Tab. I shows its effectiveness particularly in the spatial dimension, that can simulate the realistic occlusions to make the pose embedding occlusion-robust.

F. Comparison with State-of-the-art

We compare our embedding model with pre-training to other state-of-the-art methods, reporting results in Tab. IV (Smarthome), Tab. II (Penn Action) and Tab. III (N-UCLA). "*" indicates that without pre-training, we reduce the number of blocks due to the paucity of training data to report the best result we can achieve. Note that for fair comparison, we use the same pose data (2D on Smarthome, Penn Action and 3D on N-UCLA) and we also perform joint-bone 2-stream fusion as 2s-AGCN [30]. We demonstrate that our model with pre-training outperforms the Baseline model (*i.e.* 2s-AGCN [30] without embedding) and other state-of-the-art models [23], [42], [30], [31], [21], [15] in pose stream that use only pose data and performs competitively compared to methods [7], [8], [24], [3], [20], [22] also using RGB data. These results suggest that OR-VPE pre-trained on sensor pose data is able to significantly boost the accuracy not only on a similar benchmark using sensor 3D pose data (*e.g.,* $+6.7\%$ on N-UCLA) but also on real-world benchmarks using estimated 2D poses (*e.g.,* $+6.6\%$ on Smarthome CV2). Compared to methods [7], [8], [24], [3], [20], [22] also using RGB data, our proposed method performs competitively on smaller benchmarks (*e.g.,* , Smarthome CV1 and Penn Action). We argue that, (i) with fewer training data, the RGB-based methods can take advantages of pre-training on Kinetics [5] that contains a larger number of real-world videos and action diversity compared to NTU-RGB+D [19] dataset. (ii) The RGB networks can better distinguish the fine-grained and object-oriented activities which might me the failure cases with our proposed methods (*e.g.,* "Usetablet": 0.16%, "Use-laptop": 0.10%, "Pour.Frombottle": 0.09%, "Pour.Fromcan": 0.14% on Smarthome CV1). To demonstrate that our proposed method can also leverage RGB information, we simply fuse the classification scores obtained from OR-VPE and an RGB-based model, I3D [5] on Penn Action dataset. The result is reported in Tab. II. One of the future directions to further improve the accuracy could be the combination of our method and RGB-based methods together by applying careful multi-modal designs [7], [8], [22], [3].

TABLE III

TOP-1 CLASSIFICATION ACCURACY COMPARISON WITH STATE-OF-THE-ART METHODS ON THE N-UCLA DATASET.

Methods	**Data**	**N-UCLA** CV(%)
NKTM [26]	RGB	85.6
I3D [5]	RGB	86.0
ST-GCN [42]	Pose	75.8
2s-AGCN [30] (Baseline)	Pose	80.2
OR-VPE (Ours)	Pose	**86.9**

TABLE IV

MEAN PER-CLASS ACCURACY COMPARISON WITH STATE-OF-THE-ART METHODS ON THE TOYOTA SMARTHOME DATASET.

Methods	**Data**	**Toyota Smarthome** CS(%)	CV1(%)	CV2(%)
DT [36]	RGB	41.9	20.9	23.7
I3D [5]	RGB	53.4	34.9	45.1
I3D+NL [38]	RGB	53.6	34.3	43.9
AssembleNet++ [28] (+object)	RGB	63.6	-	-
Separable STA [7]	RGB+Pose	54.2	35.2	50.3
VPN [8]	RGB+Pose	60.8	43.8	53.5
LSTM [23]	Pose	42.5	13.4	17.2
MS-AAGCN [31]	Pose	60.4	-	-
MS-G3D Net [21]	Pose	61.1	17.5	59.4
2s-AGCN [30] (Baseline)	Pose	60.9	22.5	53.5
OR-VPE (Ours)	Pose	**62.2**	**25.4**	**60.1**

G. Comparison with Supervised Embedding

In this section, we show the proposed self-supervised embedding compared with supervised embedding. The results in Tab I suggest that without manual action annotations,

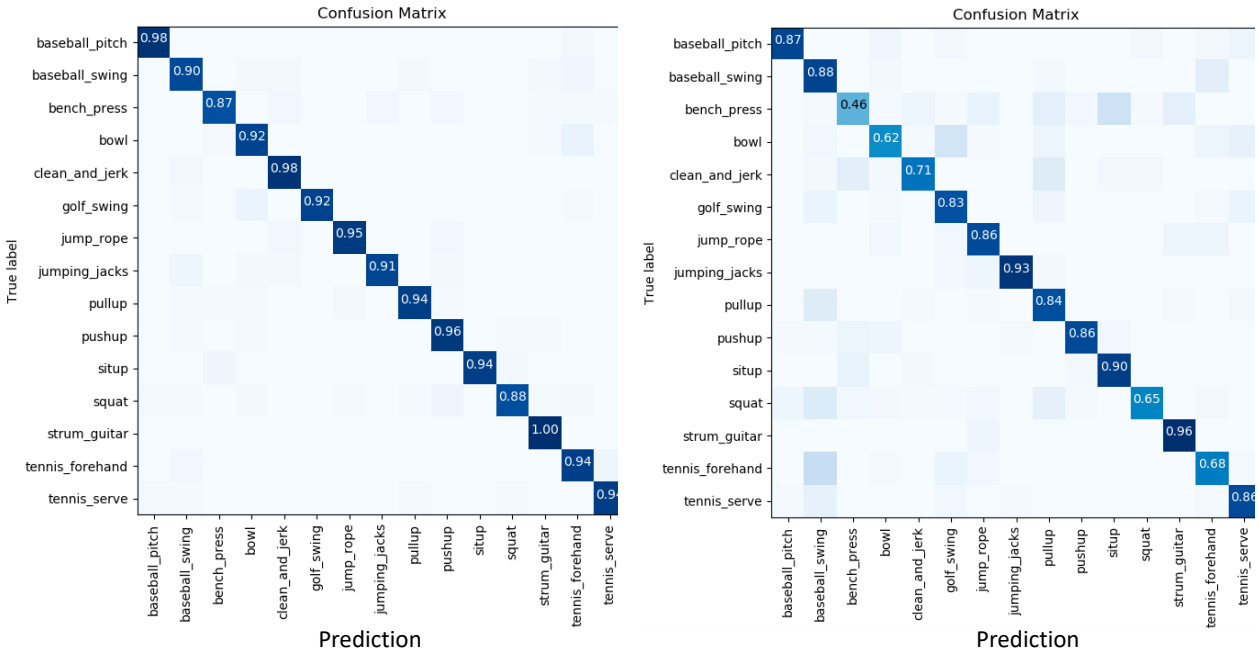


Fig. 4. **Confusion matrices of action classification on Penn Action.** Impact of OR-VPE with pre-trained Embedding (**Ours-left**) compared with AGCNs (**Baseline-right**).

we can still achieve performance similar (even better) to the supervised framework. So we can estimate pose for any unlabeled videos to learn the video pose representation for downstream action recognition tasks, which simplifies the practical applications in the real-world.

H. Classification Visualization

We visualize the confusion matrices on Penn Action (see Fig. 4) which contains the action classification accuracy for each action class. It further demonstrates the impact of our OR-VPE.

V. Conclusions

In this paper, we propose a self-supervised video pose embedding framework that renders video pose representation robust to occlusion, as well as is able to generalize onto real-world action datasets. Our experimental analysis shows that the proposed embedding networks (OR-VPE) pre-trained on a single pose dataset from RGBD sensors can also have a strong generalization ability across datasets with real-world estimated poses containing occlusions. OR-VPE allows us to extract poses in real-world videos and learn a good representation, without need of action categories that can boost the performance of downstream targeting action recognition datasets. We plan to make OR-VPE publicly available for the research community.

Acknowledgement

This work has been supported by Toyota Motor Europe (TME) and the French government, through the 3IA Cote d'Azur Investmentsin the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002. We are grateful to Inria-Sophia Antipolis "NEF" computation cluster for providing resources and support.

References

[1] P. Bachman, R. D. Hjelm, and W. Buchwalter. Learning representations by maximizing mutual information across views. In *NeurIPS*, 2019.

[2] C. Caetano, F. Brémond, and W. Schwartz. Skeleton image representation for 3D action recognition based on tree structure and reference joints. *SIBGRAPI*, 2019.

[3] C. Cao, Y. Zhang, C. Zhang, and H. Lu. Body joint guided 3-d deep convolutional descriptors for action recognition. *IEEE Transactions on Cybernetics*, 2018.

[4] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2D pose estimation using part affinity fields. *IEEE TPAMI*, 2019.

[5] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[6] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[7] S. Das, R. Dai, M. Koperski, L. Minciullo, L. Garattoni, F. Bremond, and G. Francesca. Toyota smarthome: Real-world activities of daily living. In *ICCV*, 2019.

[8] S. Das, S. Sharma, R. Dai, F. Bremond, and M. Thonnat. Vpn: Learning video-pose embedding for activities of daily living. *ECCV*, 2020.

[9] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman. Temporal cycle-consistency learning. In *CVPR*, 2019.

[10] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu. RMPE: Regional multi-person pose estimation. In *ICCV*, 2017.

[11] X. Gao, W. Hu, J. Tang, J. Liu, and Z. Guo. Optimized skeleton-based action recognition via sparsified graph regression. *ACM MM*, 2019.

[12] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

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

[14] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE TPAMI*, 2014.

[15] U. Iqbal, M. Garbade, and J. Gall. Pose for action - action for pose. In *FG*, 2017.

[16] Y. Jiao, Y. Xiong, J. Zhang, Y. Zhang, T. Zhang, and Y. Zhu. Sub-graph contrast for scalable self-supervised graph representation learning. In *ICDM*, 2020.

[17] C. Li, Q. Zhong, D. Xie, and S. Pu. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. In *IJCAI*, 2018.

[18] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *CVPR*, 2019.

[19] J. Liu, A. Shahroudy, M. Perez, G. Wang, L. Y. Duan, and A. C. Kot. Ntu rgb+d 120: A large-scale benchmark for 3D human activity understanding. *IEEE TPAMI*, 2020.

[20] M. Liu and J. Yuan. Recognizing human actions as the evolution of pose estimation maps. In *CVPR*, 2018.

[21] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *CVPR*, 2020.

[22] D. Luvizon, D. Picard, and H. Tabia. Multi-task deep learning for real-time 3D human pose estimation and action recognition. *IEEE TPAMI*, 2020.

[23] B. Mahasseni and S. Todorovic. Regularizing long short term memory with 3D human-skeleton sequences for action recognition. *CVPR*, 2016.

[24] B. X. Nie, C. Xiong, and S. Zhu. Joint action recognition and pose estimation from video. In *CVPR*, 2015.

[25] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli. 3D human pose estimation in video with temporal convolutions and semi-supervised training. In *CVPR*, 2019.

[26] H. Rahmani and A. Mian. Learning a non-linear knowledge transfer model for cross-view action recognition. In *CVPR*, 2015.

[27] G. Rogez, P. Weinzaepfel, and C. Schmid. LCR-Net++: Multi-person 2D and 3D Pose Detection in Natural Images. *IEEE TPAMI*, 2019.

[28] M. Ryoo, A. Piergiovanni, J. Kangaspunta, and A. Angelova. Assemblenet++: Assembling modality representations via attention connections. *ECCV*, 2020.

[29] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[30] L. Shi, Y. Zhang, J. Cheng, and H. Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*, 2019.

[31] L. Shi, Y. Zhang, J. Cheng, and H. LU. Skeleton-Based Action Recognition with Multi-Stream Adaptive Graph Convolutional Networks. *IEEE TIP*, 2020.

[32] J. J. Sun, J. Zhao, L.-C. Chen, F. Schroff, H. Adam, and T. Liu. View-invariant probabilistic embedding for human pose. In *ECCV*, 2020.

[33] A. B. Tanfous, H. Drira, and B. B. Amor. Sparse coding of shape trajectories for facial expression and action recognition. *IEEE TPAMI*, 2019.

[34] Y. Tian, D. Krishnan, and P. Isola. Contrastive multiview coding. In *ECCV*, 2020.

[35] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3D skeletons as points in a lie group. *CVPR*, 2014.

[36] H. Wang, A. Kläser, C. Schmid, and C. Liu. Action recognition by dense trajectories. *CVPR*, 2011.

[37] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu. Cross-view action modeling, learning and recognition. In *CVPR*, 2014.

[38] X. Wang, R. B. Girshick, A. Gupta, and K. He. Non-local neural networks. *CVPR*, 2018.

[39] X. Wang, A. Jabri, and A. A. Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019.

[40] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

[41] C. Xie, C. Li, B. Zhang, C. Chen, J. Han, C. Zou, and J. Liu. Memory attention networks for skeleton-based action recognition. *IJCAI*, 2018.

[42] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *AAAI*, 2018.

[43] D. Yang, R. Dai, Y. Wang, R. Mallick, L. Minciullo, G. Francesca, and F. Brémond. Selective spatio-temporal aggregation based pose refinement system: Towards understanding human activities in real-world videos. In *WACV*, 2021.

[44] D. Yang, Y. Wang, A. Dantcheva, L. Garattoni, G. Francesca, and F. Bremond. Unik: A unified framework for real-world skeleton-based action recognition. *BMVC*, 2021.

[45] W. Zhang, M. Zhu, and K. G. Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *ICCV*, 2013.

[46] Z. Zhang. Microsoft kinect sensor and its effect. *IEEE MM*, 2012.

[47] W. Zheng, L. Li, Z. Zhang, Y. Huang, and L. Wang. Relational network for skeleton-based action recognition. In *ICME*, 2019.