International Journal of Computer Vision
https://doi.org/

Meet JEANIE: a Similarity Measure for 3D Skeleton Sequences via Temporal-Viewpoint Alignment

Lei Wang* · **Jun Liu** · **Liang Zheng** · **Tom Gedeon** · **Piotr Koniusz***

Received: 08.30.2023 / Accepted: date

Abstract Video sequences exhibit significant nuisance variations (undesired geometric variability) due to varying speed of actions, temporal locations, and poses. When evaluating the similarity of two sequences, such variations result in temporal-viewpoint misalignments that impair comparison of two sets of frames. Thus, we propose Joint tEmporal and cAmera viewpoiNt alIgnmEnt (JEANIE) for sequence pairs. In particular, we focus on 3D skeleton sequences whose camera and subject's poses can be easily manipulated in 3D. We evaluate JEANIE on skeletal few-shot action recognition, where matching correctly frames of support and query sequence pairs (by factoring out nuisance variations) is essential due to limited samples representing novel classes. Given a query sequence, we create its several views by simulating several camera locations. For a support sequence, we match it with view-simulated query sequences as in the popular Dynamic Time Warping (DTW). Specifically, with the goal of computing optimal distance, each support frame can be matched to the query frame with the same frame index or neighbouring frame index to allow locally time warping. However, we simultaneously also let each support frame match across adjacent camera views of query frame to achieve camera viewpoint warping. Multiple alignment patterns of query and support frames are possible, resulting in multiple paths. Thus, each path is a matching plan describing between which support-query frame pairs the feature distances are evaluated and aggregated. Finally, the path with the minimum aggregated distance is selected as output of JEANIE. Through viewpoint simulation and matching across adjacent views, JEANIE achieves viewpoint alignment, an advantage over DTW which only performs temporal alignment. JEANIE achieves state-of-the-art results on NTU-60, NTU-120, Kinetics-skeleton and UWA3D Multiview Activity II on supervised and unsupervised setups.

· L. Wang is a Research Fellow at the School of Computing, the Australian National University (ANU). E-mail: lei.w@anu.edu.au.
· J. Liu is an Assistant Professor at the Singapore University of Technology and Design.
· L. Zheng is an Associate Professor at ANU.
· T. Gedeon is an Optus Chair for Artificial Intelligence at the Curtin University, Australia.
· P. Koniusz (*: equal contribution, PK: corresponding author) is a Senior Research Scientist at Data61♥CSIRO & Senior Honorary Lecturer at the ANU. E-mail: piotr.koniusz@data61.csiro.au. Code is available: https://github.com/LeiWangR/JEANIE.

1 Introduction

Action recognition is a key topic in computer vision due to its applications in video surveillance [87, 91], human-computer interaction, sport analysis and robotics. Many pipelines [82, 24, 23, 9, 90, 43, 96, 98, 88] perform action classification given the large amount of labeled training data. However, manually collecting and labeling videos for 3D skeleton sequences is laborious, and such pipelines need to be retrained or fine-tuned for new class concepts. Popular action recognition networks such as the two-stream neural network [24, 23, 104] and 3D Convolutioal Neural Network (3D CNN) [82, 9] aggregate frame-wise and temporal block representations, respectively. However, such networks are trained on large-scale datasets such as Kinetics [9, 97, 92, 99] under a fixed set of training class concepts.

Thus, there exists a growing interest in devising effective Few-shot Learning (FSL) models for action recognition, termed Few-shot Action Recognition (FSAR), that rapidly adapt to novel classes given few training samples [64, 108, 31, 19, 116, 7, 94]. FSAR models are scarce due to the volumetric nature of videos and large intra-class variations.

In contrast, FSL for image recognition has been widely studied [63, 48, 25, 4, 22, 46] including contemporary CNN-based FSL methods [41, 85, 75, 26, 79, 113] which use meta-learning, prototype-based learning or feature represen-

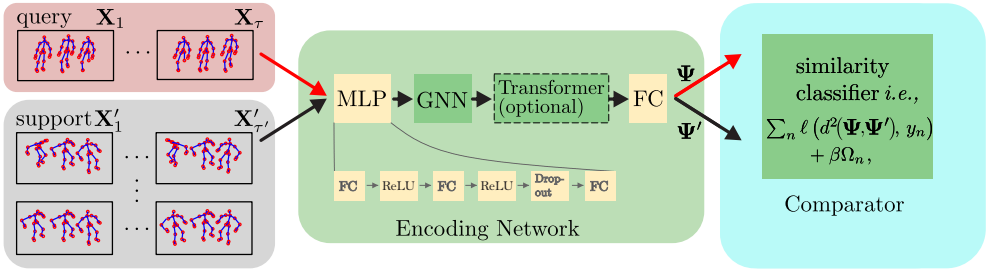


Fig. 1: Skeletal FSAR (simplified overview) takes episodes of query and support sequences, splits them into temporal blocks ($\mathbf{X}_1, ..., \mathbf{X}_\tau$ and $\mathbf{X}'_1, ..., \mathbf{X}'_\tau$), passes them to Encoding Network to obtain features $\mathbf{\Psi} = [\psi_1, ..., \psi_\tau]$ and $\mathbf{\Psi}' = [\psi'_1, ..., \psi'_{\tau'}]$, and Comparator which typically uses some distance measure $d(\cdot,\cdot)$, regularization Ω and the similarity classifier $\ell(\cdot,\cdot)$.

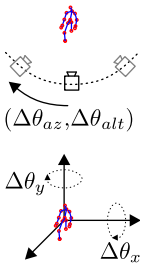


Fig. 2: One may use (*top*) stereo projections to simulate different camera views or simply use (*bottom*) Euler angles to rotate 3D scene.

tation learning. Just in 2020–2023, many FSL methods [32, 18, 101, 51, 58, 21, 29, 49, 20, 7, 80, 44, 73, 115, 130, 57, 131] have been dedicated to image classification or detection [112, 114, 121, 123, 122]. In contrast, in this paper, we aim at advancing few-shot action recognition of articulated set of connected 3D body joints, simply put, skeletal FSAR.

With the exception of very recent models [53, 52, 62, 61, 59, 94, 128], FSAR approaches that learn from skeleton-based 3D body joints are scarce. The above situation prevails despite action recognition from articulated sets of connected body joints, expressed as 3D coordinates, does offer a number of advantages over videos such as (i) the lack of the background clutter, (ii) the volume of data being several orders of magnitude smaller, and (iii) the 3D geometric manipulations of skeletal sequences being algorithm-friendly.

Video sequences may be captured under varying camera poses where subjects may follow different trajectories resulting in subjects' pose variations. Variations of action speed, location, and motion dynamics are also common. Yet, FSAR has to learn and infer similarity between support-query sequence pairs under the limited number of samples of novel classes. Thus, a good measure of similarity between support-query sequence pairs has to factor out the above variations. To this end, we propose a FSAR model that learns on skeleton-based 3D body joints via Joint tEmporal and cAmera viewpoiNt alIgnmEnt (JEANIE). We focus on 3D skeleton sequences as camera/subject's pose can be easily altered in 3D by the use of projective camera geometry.

JEANIE achieves good matching of queries with support sequences by simultaneously modeling the optimal (i) temporal and (ii) viewpoint alignments. To this end, we build on soft-DTW [16], a differentiable variant of Dynamic Time Warping (DTW) [15] (Fig. 5 is an overview how DTW differs from the Euclidean distance). Given a query sequence, we create its several views by simulating several camera locations. For a support sequence, we can match it with view-simulated query sequences as in DTW. Specifically, with the goal of computing optimal distance, each support temporal block[1] can be matched to the query temporal block with the same temporal block index or neighbouring temporal block index to allow locally time warping. However, we simultaneously also let each support temporal block match across adjacent camera views of query temporal block to achieve camera viewpoint warping. Multiple alignment patterns of query and support blocks result in multiple paths across temporal and viewpoint modes. Thus, each path represents a matching plan describing between which support-query block pairs the feature distances are evaluated and aggregated. By the use of soft-minimum, the path with the minimum aggregated distance is selected as the output of JEANIE. Thus, while DTW provides optimal temporal alignment of support-query sequence pairs, JEANIE simultaneously provides the optimal joint temporal-viewpoint alignment.

To facilitate the viewpoint alignment in JEANIE, we use easy 3D geometric operations. Specifically, we obtain skeletons under several viewpoints by rotating skeletons (zero-centered by hip) via Euler angles [1],or generating skeleton locations given simulated camera positions, according to the algebra of stereo projections [2].

We note that view-adaptive models for action recognition do exist. View Adaptive Recurrent Neural Network [117, 118] is a classification model equipped with a view-adaptive subnetwork that contains the rotation/translation switches within its RNN backbone and the main LSTM-based network. Temporal Segment Network [100] models long-range temporal structures with a new segment-based sampling and aggregation module. However, such pipelines require a large number of training samples with varying viewpoints and temporal shifts to learn a robust model. Their limitations become evident when a network trained under a fixed set of

[1] In fact, we bundle several neighboring frames into a temporal block, and perform alignment between support-query sequence pairs by temporally aligning temporal blocks rather than individual frames.

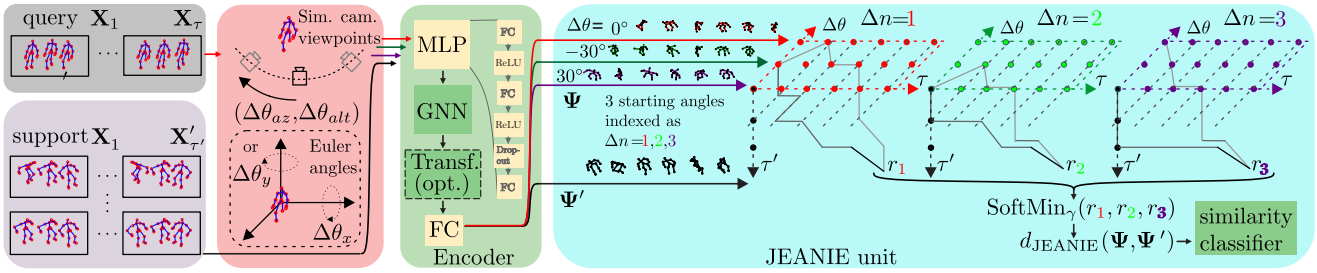


Fig. 3: Our 3D skeleton-based FSAR with JEANIE. Frames from a query sequence and a support sequence are split into short-term temporal blocks $\mathbf{X}_1, ..., \mathbf{X}_\tau$ and $\mathbf{X}'_1, ..., \mathbf{X}'_{\tau'}$ of length M given stride S. Subsequently, we generate (i) multiple rotations by $(\Delta\theta_x, \Delta\theta_y)$ of each query skeleton by either Euler angles (baseline approach) or (ii) simulated camera views (gray cameras) by camera shifts $(\Delta\theta_{az}, \Delta\theta_{alt})$ *w.r.t.* the assumed average camera location (black camera). We pass all skeletons via Encoding Network (with an optional transformer) to obtain feature tensors $\boldsymbol{\Psi}$ and $\boldsymbol{\Psi}'$, which are directed to JEANIE. We note that the temporal-viewpoint alignment takes place in 4D space (we show a 3D case with three views: $-30°, 0°, 30°$). Temporally-wise, JEANIE starts from the same $t=(1,1)$ and finishes at $t=(\tau, \tau')$ (as in DTW). Viewpoint-wise, JEANIE starts from every possible camera shift $\Delta\theta \in \{-30°, 0°, 30°\}$ (we do not know the true correct pose) and finishes at one of possible camera shifts. At each step, the path may move by no more than $(\pm\Delta\theta_{az}, \pm\Delta\theta_{alt})$ to prevent erroneous alignments. Finally, SoftMin picks up the smallest distance.

action classes has to be adapted to samples of novel classes. Our JEANIE does not suffer from such a limitation.

Figure 1 is a simplified overview of our pipeline which can serve as a template for baseline FSAR. It shows that our pipeline consists of an MLP which takes neighboring frames forming a temporal block. Each sequence consists of several such temporal blocks. However, as in Figure 2, we sample desired Euler rotations or simulated camera viewpoints, generate multiple skeleton views, and pass them to the MLP to get block-wise feature maps, next forwarded to a Graph Neural Network (GNN), *e.g.*, GCN [39, 78, 106, 40, 129], We mainly use a linear S^{2}GC [129, 132], followed by an optional transformer [17], and an FC layer to obtain block feature vectors passed to JEANIE whose output distance measurement are passed to our similarity classifier. Figure 3 is a detailed overview of our supervised FSAR pipeline.

Note that JEANIE can be thought of as a kernel in Reproducing Kernel Hilbert Spaces (RKHS) [74] based on Optimal Transport [84] with a specific temporal-viewpoint transportation plan. As kernels capture the similarity of sample pairs instead of modeling class labels, they are a natural choice for FSL and FSAR problems.

In this paper, we extend our supervised FSAR model [93] by introducing an unsupervised FSAR model, and a fusion of both supervised and unsupervised models. Our rationale for an unsupervised FSAR extension is to demonstrate that the invariance properties of JEANIE (dealing with temporal and viewpoint variations) help naturally match sequences of the same class without the use of additional knowledge (class labels). Such a setting demonstrates that JEANIE is able to limit intra-class variations (temporal and viewpoint variations) facilitating unsupervised matching of sequences.

For unsupervised FSAR, JEANIE is used as a distance measure in the feature reconstruction term of dictionary learning and feature coding steps. Features of the temporal blocks are projected into such a dictionary space and the projection codes representing sequences are used for similarity measure between support-query sequences. This idea is similar to clustering training sequences into k-means clusters [14] to form a dictionary. Then the assignments of test query sequences to such a dictionary can reveal their class labels based on labeled test support sequence falling into the same cluster. However, even with JEANIE used as a distance measure, one-hot assignments resulting from k-means are sub-optimal. Thus, we investigate more recent soft assignment [6, 27, 42, 54] and sparse coding approaches [47, 111].

Finally, we also introduce a simple fusion of supervised and unsupervised FSAR by alignment of supervised and unsupervised FSAR features or by MAML-inspired [26] fusion of unsupervised and supervised FSAR losses in the so-called inner and outer loop, respectively.

Below are our contributions:

i. We propose JEANIE that performs the joint alignment of temporal blocks and simulated camera viewpoints of 3D skeletons between support-query sequences to select the optimal alignment path which realizes joint temporal (time) and viewpoint warping. We evaluate JEANIE on skeletal few-shot action recognition, where matching correctly support and query sequence pairs (by factoring out nuisance variations) is essential due to limited samples representing novel classes.
ii. To simulate different camera locations for 3D skeleton sequences, we consider rotating them (1) by Euler angles within a specified range along axes, or (2) towards

the simulated camera locations based on the algebra of stereo projection.

iii. We propose unsupervised FSAR where JEANIE is used as a distance measure in the feature reconstruction term of dictionary learning and coding steps (we investigate several such coders). We use projection codes to represent sequences. Moreover, we also introduce an effective fusion of both supervised and unsupervised FSAR models by unsupervised and supervised feature alignment term or MAML-inspired fusion of unsupervised and supervised FSAR losses.

iv. As minor contributions, we investigate different GNN backbones (combined with an optional transformer), as well as the optimal temporal size and stride for temporal blocks encoded by a simple 3-layer MLP unit before forwarding them to GNN. We also propose a simple similarity-based loss encouraging the alignment of within-class sequences and preventing the alignment of between-class sequences.

We achieve the state of the art on large-scale NTU-60 [70], NTU-120 [52], Kinetics-skeleton [109] and UWA3D Multiview Activity II [68].

2 Related Works

Below, we describe 3D skeleton-based AR, FSAR approaches, and Graph Neural Networks.

Action recognition (3D skeletons). 3D skeleton-based action recognition pipelines often use GCNs [39], *e.g.*, spatio-temporal GCN (ST-GCN) [109], Attention enhanced Graph Convolutional LSTM network (AGC-LSTM) [72], Actional-Structural GCN (AS-GCN) [50], Dynamic Directed GCN (DDGCN) [45], Decoupling GCN with DropGraph module [12], Shift-GCN [13], Semantics-Guided Neural Networks (SGN) [119], AdaSGN [71], Context Aware GCN (CA-GCN) [124], Channel-wise Topology Refinement Graph Convolution Network (CTR-GCN) [11], a family of Efficient GCN (EfficientGCN-Bx) [76] and Disentangling and Unifying Graph Convolutions [56]. As ST-GCN applies convolution along structural connections (links between body joints), structurally distant joints, which may cover key patterns of actions, are largely ignored. While GCN can be applied to a fully-connected graph to capture complex interactions of body joints, groups of nodes across space and/or time can be captured with tensors [43], semi-dynamic hypergraph neural networks [55], hypergraph GNN [34], angular features [67], Higher-order Transformer (HoT) [38] and Multi-order Multi-mode Transformer (3Mformer) [95]. Most recently, a Koopman pooling framework [102], an auxiliary feature refinement head [127] and a Spatial-Temporal Mesh Transformer (STMT) [134] have been proposed for 3D skeleton-based AR.

However, such models rely on large-scale datasets to train large numbers of parameters, and cannot be adapted with ease to novel class concepts whereas FSAR can.

FSAR (videos). Approaches [64, 31, 108] use a generative model, graph matching on 3D coordinates and dilated networks, respectively. Approach [133] uses a compound memory network. ProtoGAN [19] generates action prototypes. Recent FSAR model [116] uses permutation-invariant attention and second-order aggregation of temporal video blocks, whereas approach [7] proposes a modified temporal alignment for query-support pairs via DTW. Recent video FSAR models include a mixed-supervised hierarchical contrastive learning (HCL) [126], Compound Prototype Matching [36], Spatio-temporal Relation Modeling [81], motion-augmented long-short contrastive learning (MoLo) [103] and Active Multimodal Few-shot Action Recognition (AMFAR) framework [105].

FSAR (3D skeletons). Few FSAR models use 3D skeletons [53, 52, 62, 61]. Global Context-Aware Attention LSTM [53] selectively focuses on informative joints. Action-Part Semantic Relevance-aware (APSR) model [52] uses the semantic relevance between each body part and action class at the distributed word embedding level. Signal Level Deep Metric Learning (DML) [62] and Skeleton-DML [61] encode signals as images, extract CNN features and use multi-similarity miner loss. New skeletal FSAR includes Disentangled and Adaptive Spatial-Temporal Matching (DASTM) [59], uncertainty-DTW [94] and Adaptive Local-Component-Aware Graph Convolutional Network (ALCA-GCN) [128].

In contrast, we use temporal blocks of skeleton sequences encoded by GNNs under multiple simulated camera viewpoints to jointly apply temporal and viewpoint alignment of query-support sequences to factor out nuisance variability.

Graph Neural Networks. GNNs modified to the specific structure of 3D skeletal data are very popular in action recognition, as detailed in "Action recognition (3D skeletons)" section. In this paper, we leverage standard GNNs due to their good ability to represent graph-structured data. GCN [39] applies graph convolution in the spectral domain, and enjoys the depth-efficiency when stacking multiple layers due to non-linearities. However, depth-efficiency extends the runtime due to backpropagation through consecutive layers. In contrast, a very recent family of so-called spectral filters do not require depth-efficiency but apply filters based on heat diffusion on graph adjacency matrix. As a result, these are fast linear models as learnable weights act on filtered node representations. Unlike general GNNs, SGC [106], APPNP [40] and S^2GC [129] are three such linear models which we investigate for the backbone, followed by an optional transformer, and an FC layer.

Transformers in action recognition. Transformers have become popular in many tasks including action recognitoin

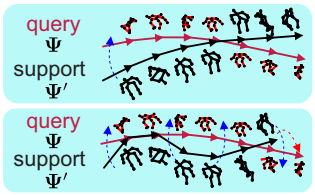


Fig. 4: (*top*) In viewpoint-invariant learning, the distance between query features $\boldsymbol{\Psi}$ and support features $\boldsymbol{\Psi}'$ has to be computed. The blue arrow indicates that trajectories of both actions need alignment. (*bottom*) In real life, subject's 3D body joints deviate from one ideal trajectory, and so advanced viewpoint alignment strategy is needed.

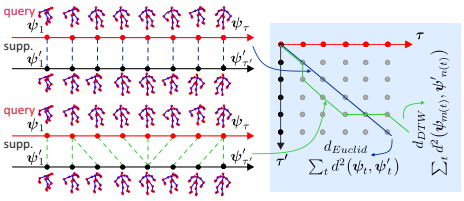


Fig. 5: Euclidean dist. *vs.* DTW. (*top*) Feature vectors $\boldsymbol{\psi}_t$ and $\boldsymbol{\psi}'_t$ of query and support frames (or temp. blocks) are matched along time t: $d_{Euclid}(\boldsymbol{\Psi}, \boldsymbol{\Psi}') = \sum_t d^2(\boldsymbol{\psi}_t, \boldsymbol{\psi}'_t)$. (*bottom*) For DTW, a path with minimum aggregated distance is selected as $d_{DTW}(\boldsymbol{\Psi}, \boldsymbol{\Psi}') = \sum_t d^2(\boldsymbol{\psi}_{m(t)}, \boldsymbol{\psi}'_{n(t)})$, and $m(t)$ and $n(t)$ parameterize query and support indexes. One is permitted steps $\downarrow$, $\searrow$, $\rightarrow$ in the graph. We expect $d_{DTW} \leq d_{Euclid}$.

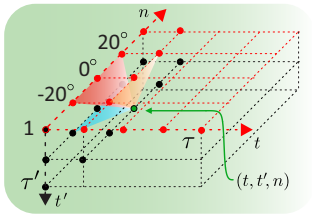


Fig. 6: JEANIE (1-max shift). We loop over all points. At (t, t', n) (green point) we add its base distance to the minimum of accumulated distances at $(t, t'-1, n-1)$, $(t, t'-1, n)$, $(t, t'-1, n+1)$ (orange plane), $(t-1, t'-1, n-1)$, $(t-1, t'-1, n)$, $(t-1, t'-1, n+1)$ (red plane) and $(t-1, t', n-1)$, $(t-1, t', n)$, $(t-1, t', n+1)$ (blue plane).

[66, 120, 125, 28, 65]. Vision Transformer (ViT) [17] is the first transformer model for image classification. The success of transformers relies on their ability to establish long-range attention among sequences in contrast to shorter relations in RNNs. Recent transformer-based AR models include Uncertainty-Guided Probabilistic Transformer (UGPT) [30], Recurrent Vision Transformer (RViT) [110], Spatio-TemporAl cRoss (STAR)-transformer [3], DirecFormer [83], Spatial-Temporal Mesh Transformer (STMT) [134], Semi-Supervised Video Transformer (SVFormer) [107] and Multi-order Multi-mode Transformer (3Mformer) [103].

In this work, we apply a simple optional transformer block with few layers following GNN to capture better block-level dependencies of 3D human body joints.

Multi-view action recognition. Multi-modal sensors enable multi-view action recognition [90, 117]. A Generative Multi-View Action Recognition framework [89] integrates complementary information from RGB and depth sensors by View Correlation Discovery Network. Some works exploit multiple views of the subject [70, 52, 118, 89] to overcome the viewpoint variations for action recognition on large training datasets. Recently, a supervised contrastive learning framework [69] for multi-view was introduced.

In contrast, our JEANIE performs jointly the temporal and simulated viewpoint alignment in an end-to-end FSAR setting. This is a novel paradigm based on improving the notion of similarity between sequences of support-query pair rather than learning class concepts.

3 Approach

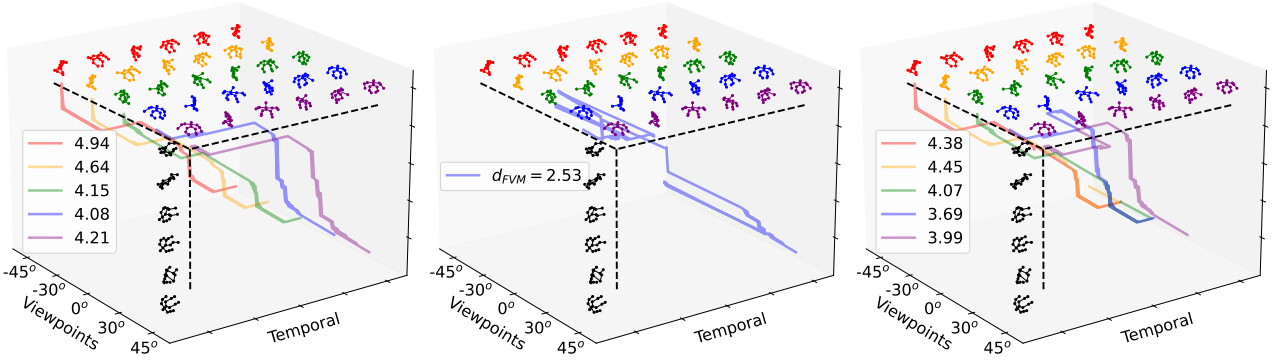
To learn similarity and dissimilarity between pairs of sequences of 3D body joints representing query and support samples from episodes, our goal is to find a joint viewpoint-temporal alignment of query and support, and minimize or maximize the matching distance d_{JEANIE} (end-to-end setting) for same or different support-query labels, respectively. Fig. 4 (top) shows that sometimes matching of query and support may be as easy as rotating one trajectory onto another, in order to achieve viewpoint invariance. A viewpoint invariant distance [33] can be defined as:

$$d_{\text{inv}}(\boldsymbol{\Psi}, \boldsymbol{\Psi}') = \inf_{\gamma, \gamma' \in T} d\big(\gamma(\boldsymbol{\Psi}), \gamma'(\boldsymbol{\Psi}')\big), \tag{1}$$

where T is a set of transformations required to achieve a viewpoint invariance, $d(\cdot, \cdot)$ is some base distance, *e.g.*, the Euclidean distance, and $\boldsymbol{\Psi}$ and $\boldsymbol{\Psi}'$ are features describing query and support pair of sequences. Typically, T may include 3D rotations to rotate one trajectory onto the other. However, a global viewpoint alignment of two sequences is suboptimal. Trajectories are unlikely to be straight 2D lines in the 3D space so one may not be able to rotate the query trajectory to align with the support trajectory. Fig. 4 (bottom) shows that the subjects' poses locally follow complicated non-linear paths.

Thus, we propose JEANIE that aligns and warps query / support sequences based on the feature similarity. One can think of JEANIE as performing Eq. (1) with T containing all possible combinations of locally time-warping augmentations of sequences and camera pose augmentations for each frame (or temporal block). JEANIE unit in Fig. 3 realizes such a strategy. Figure 6 (discussed later in the text) shows one step of the temporal-viewpoint computations of JEANIE in search for optimal temporal-viewpoint alignment path between query and support sequences. Soft-minimum across all such possible alignment paths can be equivalently written as an infinimum over a set of specific tansformations in Eq. (1).

Below, we detail our pipeline, and explain the proposed JEANIE, Encoding Network (EN), feature coding and dic-



(a) soft-DTW (view-wise) (b) FVM (c) JEANIE(1-max shift)

Fig. 7: A comparison of paths in 3D for soft-DTW, Free Viewpoint Matching (FVM) and our JEANIE. For a given support skeleton sequence (green color), we choose viewing angles between -45^0 and 45^0 for the camera viewpoint simulation. The support skeleton sequence is shown in black color. (a) soft-DTW finds each individual alignment per viewpoint fixed throughout alignment: $d_{\text{shortest}} = 4.08$. (b) FVM is a greedy matching algorithm that in each time step seeks the best alignment pose from all viewpoints which leads to unrealistic zigzag path (person cannot jump from front to back view suddenly): $d_{\text{FVM}} = 2.53$. (c) Our JEANIE (1-max shift) is able to find smooth joint viewpoint-temporal alignment between support and query sequences. We show each optimal path for each possible starting position: $d_{\text{JEANIE}} = 3.69$. While $d_{\text{FVM}} = 2.53$ for FVM is overoptimistic, $d_{\text{shortest}} = 4.08$ for fixed-view matching is too pessimistic, whereas JEANIE strikes the right matching balance with $d_{\text{JEANIE}} = 3.69$.

tionary learning, and our loss function. Firstly, we present our notations.

Notations. $\mathcal{I}_K$ stands for the index set $\{1, 2, ..., K\}$. Concatenation of α_i is denoted by $[\alpha_i]_{i \in \mathcal{I}_I}$, whereas $\mathbf{X}_{:,i}$ means we extract/access column i of matrix $\boldsymbol{D}$. Calligraphic mathcal fonts denote tensors (*e.g.*, $\boldsymbol{\mathcal{D}}$), capitalized bold symbols are matrices (*e.g.*, $\boldsymbol{D}$), lowercase bold symbols are vectors (*e.g.*, $\boldsymbol{\psi}$), and regular fonts denote scalars.

Prerequisites. Below we refer to prerequisites used in the subsequent chapters. Appendix A explains how Euler angles and stereo projections are used in simulating different skeleton viewpoints. Appendix B explains several GNN approaches that we use in our Encoding Network. Appendix C explains several feature coding and dictionary learning strategies which we use for unsupervised FSAR.

3.1 Encoding Network (EN)

We start by generating $K \times K'$ Euler rotations or $K \times K'$ simulated camera views (moved gradually from the estimated camera location) of query skeletons. Our EN contains a simple 3-layer MLP unit (FC, ReLU, FC, ReLU, Dropout, FC), GNN, optional Transformer [17] and FC. The MLP unit takes M neighboring frames, each with J 3D skeleton body joints, forming one temporal block $\mathbf{X} \in \mathbb{R}^{3 \times J \times M}$, where 3 indicates 3D Cartesian coordinates. In total, depending on stride S, we obtain some τ temporal blocks which capture the short temporal dependency, whereas the long temporal dependency is modeled with our JEANIE. Each temporal block is encoded by the MLP into a $d \times J$ dimensional feature map:

$$\widehat{\mathbf{X}} = (\text{MLP}(\mathbf{X}; \mathcal{F}_{MLP}))^T \in \mathbb{R}^{J \times d}. \tag{2}$$

We obtain $K \times K' \times \tau$ query and τ' support feature maps, each of size $J \times d$. Each maps is forwarded to a GNN. For S^{2}GC [129] (default GNN in our work) with L layers, we have:

$$\widehat{\widehat{\mathbf{X}}} = \frac{1}{L} \sum_{l=1}^{L} \left((1-\alpha) \mathbf{S}^l \widehat{\mathbf{X}} + \alpha \widehat{\mathbf{X}} \right) \in \mathbb{R}^{J \times d}, \tag{3}$$

where $\mathbf{S}$ is the adjacency matrix capturing connectivity of body joints, whereas $0 \leq \alpha \leq 1$ controls the self-importance of each body joint. Appendix B describe several GNN variants we experimented with: GCN [39], SGC [106], APPNP [40] and S^{2}GC [129].

Optionally, a transformer[2] (described below in "Transformer Encoder") may be used. Finally, an FC layer returns $\boldsymbol{\Psi} \in \mathbb{R}^{d' \times K \times K' \times \tau}$ query feature maps and $\boldsymbol{\Psi'} \in \mathbb{R}^{d' \times \tau'}$ support feature maps. Feature maps are passed to JEANIE whose output is passed into the similarity classifier. The whole Encoding Network is summarized as follows. Let support maps $\boldsymbol{\Psi'} \equiv [f(\boldsymbol{X}'_1; \mathcal{F}), ..., f(\boldsymbol{X}'_{\tau'}; \mathcal{F})] \in \mathbb{R}^{d' \times \tau'}$ and query maps $\boldsymbol{\Psi} \equiv [f(\boldsymbol{X}_{1,1,1}; \mathcal{F}), ..., f(\boldsymbol{X}_{K,K',\tau}; \mathcal{F})] \in \mathbb{R}^{d' \times K \times K' \times \tau}$. For M query and M support frames per

[2]Our transformer is similar to ViT [17] but instead of using image patches, we feed each body joint encoded by GNN into the transformer.

block, $\mathbf{X} \in \mathbb{R}^{3\times J\times M}$ and $\mathbf{X}' \in \mathbb{R}^{3\times J\times M}$. We also define:

$$f(\mathbf{X};\mathcal{F}) = \tag{4}$$
$$\text{FC}(\text{Transf}(\text{GNN}(\text{MLP}(\mathbf{X};\mathcal{F}_{MLP});\mathcal{F}_{GNN});\mathcal{F}_{Tr});\mathcal{F}_{FC}),$$

where $\mathcal{F} \equiv [\mathcal{F}_{MLP}, \mathcal{F}_{GNN}, \mathcal{F}_{Tr}, \mathcal{F}_{FC}]$ is the set of parameters of EN (including an optional transformer).

Transformer Encoder. Vision transformer [17] consists of alternating layers of Multi-Head Self-Attention (MHSA) and a feed-forward MLP (2 FC layers with a GELU non-linearity intertwined). LayerNorm (LN) is applied before every block, and residual connections after every block. If transformer is used, each feature matrix $\widehat{\mathbf{X}} \in \mathbb{R}^{J\times d}$ per temporal block is encoded by a GNN into $\widehat{\widehat{\mathbf{X}}} \in \mathbb{R}^{J\times d}$ and then passed to the transformer. Similarly to the standard transformer, we prepend a learnable vector $\mathbf{y}_{\text{token}} \in \mathbb{R}^{1\times d}$ to the sequence of block features $\widehat{\mathbf{X}}$ obtained from GNN, and we also add the positional embeddings $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{(1+J)\times d}$ based on the standard sine and cosine functions so that token $\mathbf{y}_{\text{token}}$ and each body joint enjoy their own unique positional encoding. One can think of our GNN block as replacing the tokenizer linear projection layer of a standard transformer. Compared to the use of FC layer as linear projection layer, our GNN tokenizer in Eq. (5) enjoys (i) better embeddings of human body joints based on the graph structure (ii) no learnable parameters. From the tokenizer, we obtain $\mathbf{Z}_0 \in \mathbb{R}^{(1+J)\times d}$:

$$\mathbf{Z}_0 = [\mathbf{y}_{\text{token}}; \text{GNN}(\widehat{\mathbf{X}})] + \mathbf{E}_{\text{pos}}, \tag{5}$$

and feed it into in the following transformer backbone:

$$\mathbf{Z}'_k = \text{MHSA}(\text{LN}(\mathbf{Z}_{k-1})) + \mathbf{Z}_{k-1}, \quad k = 1, ..., L_{\text{tr}} \tag{6}$$
$$\mathbf{Z}_k = \text{MLP}(\text{LN}(\mathbf{Z}'_k)) + \mathbf{Z}'_k, \quad k = 1, ..., L_{\text{tr}} \tag{7}$$
$$\mathbf{y}' = \text{LN}(\mathbf{Z}^{(0)}_{L_{\text{tr}}}) \quad \text{where} \quad \mathbf{y}' \in \mathbb{R}^{1\times d} \tag{8}$$
$$f(\mathbf{X};\mathcal{F}) = \text{FC}(\mathbf{y}'^T;\mathcal{F}_{FC}) \quad \in \mathbb{R}^{d'}, \tag{9}$$

where $\mathbf{Z}^{(0)}_{L_{\text{tr}}}$ is the first d-dimensional row vector extracted from the output matrix $\mathbf{Z}_{L_{\text{tr}}} \in \mathbb{R}^{(J+1)\times d}$, and L_{tr} controls the depth of the transformer (the number of layers), whereas $\mathcal{F} \equiv [\mathcal{F}_{MLP}, \mathcal{F}_{GNN}, \mathcal{F}_{Tr}, \mathcal{F}_{FC}]$ is the set of parameters of EN. Finally, $f(\mathbf{X};\mathcal{F})$ from Eq. (9) becomes equivalent of Eq. (4) with the transformer.

3.2 JEANIE

Prior to explaining the details of the JEANIE measure, we briefly explain details of soft-DTW.

Soft-DTW [15, 16]. Dynamic Time Warping can be seen as a specialized "metric" with a matching transportation plan[3] acting on the temporal mode of sequences. Soft-DTW is defined as:

$$d_{\text{DTW}}(\boldsymbol{\Psi},\boldsymbol{\Psi}') = \underset{\mathbf{A}\in\mathcal{A}_{\tau,\tau'}}{\text{SoftMin}_\gamma} \left\langle \mathbf{A}, \boldsymbol{D}(\boldsymbol{\Psi},\boldsymbol{\Psi}') \right\rangle, \tag{10}$$

$$\text{where } \text{SoftMin}_\gamma(\boldsymbol{\alpha}) = -\gamma \log \sum_i \exp(-\alpha_i/\gamma). \tag{11}$$

The binary $\mathbf{A} \in \mathcal{A}_{\tau,\tau'}$ encodes a path within the transportation plan $\mathcal{A}_{\tau,\tau'}$ which depends on lengths τ and τ' of sequences $\boldsymbol{\Psi} \equiv [\boldsymbol{\psi}_1, ..., \boldsymbol{\psi}_\tau] \in \mathbb{R}^{d'\times\tau}$, $\boldsymbol{\Psi}' \equiv [\boldsymbol{\psi}'_1, ..., \boldsymbol{\psi}'_{\tau'}] \in \mathbb{R}^{d'\times\tau'}$. $\boldsymbol{D} \in \mathbb{R}_+^{\tau\times\tau'} \equiv [d_{\text{base}}(\boldsymbol{\psi}_m, \boldsymbol{\psi}'_n)]_{(m,n)\in\mathcal{I}_\tau\times\mathcal{I}_{\tau'}}$ is the matrix of distances, evaluated for $\tau\times\tau'$ frames (or temporal blocks) according to some base distance $d_{\text{base}}(\cdot,\cdot)$, *i.e.*, the Euclidean distance.

In what follows, we make use of principles of soft-DTW, *i.e.*, the property of time-warping. However, we design a joint alignment between temporal skeleton sequences and simulated skeleton viewpoints, which means we achieve joint time-viewpoint warping (a novel idea never done before).

JEANIE. Matching query-support pairs requires temporal alignment due to potential offset in locations of discriminative parts of actions, and due to potentially different dynamics/speed of actions taking place. The same concerns the direction of actor's pose, *i.e.*, consider the pose trajectory *w.r.t.* the camera. Thus, the JEANIE measure is equipped with an extended transportation plan $\mathcal{A}' \equiv \mathcal{A}_{\tau,\tau',K,K'}$, where apart from temporal block counts τ and τ', for query sequences, we have possible η_{az} left and η_{az} right steps from the initial camera azimuth, and η_{alt} up and η_{alt} down steps from the initial camera altitude. Thus, $K = 2\eta_{az}+1$, $K' = 2\eta_{alt}+1$. For the variant with Euler angles, we simply have $\mathcal{A}'' \equiv \mathcal{A}_{\tau,\tau',K,K'}$ where $K = 2\eta_x+1$, $K' = 2\eta_y+1$ instead. The JEANIE formulation is given as:

$$d_{\text{JEANIE}}(\boldsymbol{\Psi},\boldsymbol{\Psi}') = \underset{\mathbf{A}\in\mathcal{A}'}{\text{SoftMin}_\gamma} \left\langle \mathbf{A}, \boldsymbol{\mathcal{D}}(\boldsymbol{\Psi},\boldsymbol{\Psi}') \right\rangle, \tag{12}$$

where $\boldsymbol{\mathcal{D}} \in \mathbb{R}_+^{K\times K'\times\tau\times\tau'} \equiv [d_{\text{base}}(\boldsymbol{\psi}_{m,k,k'}, \boldsymbol{\psi}'_n)]_{\substack{(m,n)\in\mathcal{I}_\tau\times\mathcal{I}_{\tau'} \\ (k,k')\in\mathcal{I}_K\times\mathcal{I}_{K'}}}$, and tensor $\boldsymbol{\mathcal{D}}$ contains distances evaluated between all possible temporal blocks.

Figure 6 illustrates one step of JEANIE. Suppose the given viewing angle set is $\{-40°, -20°, 0°, 20°, 40°\}$. For the current node at (t, t', n) we evaluate, we have to aggregate its base distance with the smallest aggregated distance of its predecessor nodes. The "1-max shift" means that the predecessor node must be a direct neighbor of the current node (imagine that dots on a 3D grid are nodes connected by links). Thus, for 1-max shift, at location (t, t', n), we extract the node's base distance and add it together with the minimum of aggregated distances at the shown 9 predecessor nodes. We store that aggregated distance at (t, t', n), and we move to the next node. Note that for viewpoint index n,

[3] In analogy to terminology used in Optimal Transport (*e.g.*, the Wasserstein distance), we call it a transportation plan. Also, notice that Soft-DTW may violate some of the metric axioms.

Algorithm 1 Joint tEmporal and cAmera viewpoiNt alIgnmEnt (JEANIE).

Input (forward pass): $\boldsymbol{\Psi}, \boldsymbol{\Psi}', \gamma > 0, d_{\text{base}}(\cdot,\cdot)$, ι-max shift.

$r_{:,:,:} = \infty$, $r_{n,1,1} = d_{\text{base}}(\boldsymbol{\psi}_{n,1}, \boldsymbol{\psi}'_1)$, $\forall n \in \{-\eta, ..., \eta\}$
$\mathit{\Pi} \equiv \{-\iota, ..., 0, ..., \iota\} \times \{(0,1), (1,0), (1,1)\}$
for $t \in \mathcal{I}_\tau$:
  for $t' \in \mathcal{I}_{\tau'}$:
    if $t \neq 1$ or $t' \neq 1$:
      for $n \in \{-\eta, ..., \eta\}$:
        $r_{n,t,t'} = d_{\text{base}}(\boldsymbol{\psi}_{n,t}, \boldsymbol{\psi}'_{t'})$
        $+ \text{SoftMin}_\gamma \left([r_{n-i,t-j,t'-k}]_{(i,j,k)\in\mathit{\Pi}} \right)$

Output: $\text{SoftMin}_\gamma \left([r_{n,\tau,\tau'}]_{n\in\{-\eta,...,\eta\}} \right)$

we look up $(n-1, n, n+1)$ neighbors. Extension to the ι-max shift is straightforward. The importance of low value of ι-max shift, *e.g.*, $\iota = 1$ is that low value of ι promotes the so-called smoothness of alignment. That is, while time or viewpoint may be warped, they are not warped abruptly (*e.g.*, the subject's pose is not allowed to suddenly rotate by 90° in one step then rotate back by -90°. This smoothness is the key preventing greedy matching that would result in an overoptimistic distance between two sequences.

Algorithm 1 illustrates JEANIE. For brevity, let us tackle the camera viewpoint alignment along the azimuth, *e.g.*, for some shifting steps $-\eta, ..., \eta$, each with size $\Delta\theta_{az}$. The maximum viewpoint change from block to block is ι-max shift (smoothness). As we have no way to know the initial optimal camera shift, we initialize all possible origins of shifts in accumulator $r_{n,1,1} = d_{\text{base}}(\boldsymbol{\psi}_{n,1}, \boldsymbol{\psi}'_1)$ for all $n \in \{-\eta, ..., \eta\}$. Subsequently, steps related to soft-DTW (temporal-viewpoint matching) take place. Finally, we choose the path with the smallest distance over all possible viewpoint ends by selecting a soft-minimum over $[r_{n,\tau,\tau'}]_{n\in\{-\eta,...,\eta\}}$. Notice that elements of the accumulator tensor $\boldsymbol{\mathcal{R}} \in \mathbb{R}^{(2\iota+1)\times\tau\times\tau'}$ are accessed by writing $r_{n,t,t'}$. Moreover, whenever either index $n-i$, $t-j$ or $t'-k$ in $r_{n-i,t-j,t'-k}$ (see algorithm) is out of bounds, we define $r_{n-i,t-j,t'-k} = \infty$.

Free Viewpoint Matching (FVM). To ascertain whether JEANIE is better than performing separately the temporal and simulated viewpoint alignments, we introduce an important and plausible baseline called Free Viewpoint Matching. FVM, for every step of DTW, seeks the best local viewpoint alignment, thus realizing a non-smooth temporal-viewpoint path in contrast to JEANIE. To this end, we apply soft-DTW in Eq. (12) with the base distance replaced by:

$$d_{\text{FVM}(\boldsymbol{\psi}_t, \boldsymbol{\psi}'_{t'})} = \underset{m,n\in\{-\eta,...,\eta\}}{\text{SoftMin}_\gamma} \; d_{\text{base}}\left(\boldsymbol{\psi}_{m,n,t}, \boldsymbol{\psi}'_{m',n',t'}\right), \quad (13)$$

where $\boldsymbol{\Psi} \in \mathbb{R}^{d'\times K\times K'\times\tau}$ and $\boldsymbol{\Psi}' \in \mathbb{R}^{d'\times K\times K'\times\tau'}$ are query and support feature maps. We abuse slightly the notation by writing $d_{\text{FVM}(\boldsymbol{\psi}_t, \boldsymbol{\psi}'_{t'})}$ as we minimize over viewpoint indexes inside of Eq. (13). Thus, we calculate the distance matrix $\boldsymbol{D} \in \mathbb{R}_+^{\tau\times\tau'} \equiv [d_{\text{FVM}}(\boldsymbol{\psi}_t, \boldsymbol{\psi}'_{t'})]_{(t,t')\in\mathcal{I}_\tau\times\mathcal{I}_{\tau'}}$ for soft-DTW.

Fig. 7 shows the comparison between soft-DTW (viewwise), FVM and our JEANIE. FVM is a greedy matching method which leads to complex zigzag path in 3D space (we illustrate the camera viewpoint in a single mode, *e.g.*, the azimuth for $\boldsymbol{\psi}_{n,t}$, and no viewpoint mode for $\boldsymbol{\psi}'_{t'}$). Although FVM is able to produce the path with a smaller aggregated distance compared to soft-DTW and JEANIE, it suffers from obvious limitations: (i) It is unreasonable for poses in a given sequence to match under extreme sudden changes of viewpoints. (ii) Even if two sequences are from two different classes, FVM still yields the smallest distance (decreased inter-class variance).

3.3 Loss Function for Supervised FSAR

For the N-way Z-shot problem, we have one query feature map and $N \times Z$ support feature maps per episode. We form a mini-batch containing B episodes. Thus, we have query feature maps $\{\boldsymbol{\Psi}_b\}_{b\in\mathcal{I}_B}$ and support feature maps $\{\boldsymbol{\Psi}'_{b,n,z}\}_{\substack{b\in\mathcal{I}_B \\ n\in\mathcal{I}_N \\ z\in\mathcal{I}_Z}}$. Moreover, $\boldsymbol{\Psi}_b$ and $\boldsymbol{\Psi}'_{b,1,:}$ share the same class, one of N classes drawn per episode, forming the subset $C^\ddagger \equiv \{c_1, ..., c_N\} \subset \mathcal{I}_C \equiv \mathcal{C}$.

Specifically, labels $y(\boldsymbol{\Psi}_b) = y(\boldsymbol{\Psi}'_{b,1,z}), \forall b \in \mathcal{I}_B, z \in \mathcal{I}_Z$ while $y(\boldsymbol{\Psi}_b) \neq y(\boldsymbol{\Psi}'_{b,n,z}), \forall b \in \mathcal{I}_B, n \in \mathcal{I}_N \setminus \{1\}, z \in \mathcal{I}_Z$. In most cases, $y(\boldsymbol{\Psi}_b) \neq y(\boldsymbol{\Psi}_{b'})$ if $b \neq b'$ and $b, b' \in \mathcal{I}_B$. Selection of $C^\ddagger$ per episode is random. For the N-way Z-shot protocol, we minimize:

$$l(\boldsymbol{d}^+, \boldsymbol{d}^-) = \left(\mu(\boldsymbol{d}^+) - \{\mu(\text{TopMin}_\beta(\boldsymbol{d}^+))\}\right)^2 \quad (14)$$

$$+ \left(\mu(\boldsymbol{d}^-) - \{\mu(\text{TopMax}_{\text{NZ}\beta}(\boldsymbol{d}^-))\}\right)^2, \quad (15)$$

$$\text{where} \quad \begin{cases} \boldsymbol{d}^+ = [d_{\text{JEANIE}}(\boldsymbol{\Psi}_b, \boldsymbol{\Psi}'_{b,1,z})]_{\substack{b\in\mathcal{I}_B \\ z\in\mathcal{I}_Z}} \\ \boldsymbol{d}^- = [d_{\text{JEANIE}}(\boldsymbol{\Psi}_b, \boldsymbol{\Psi}'_{b,n,z})]_{\substack{b\in\mathcal{I}_B \\ n\in\mathcal{I}_N\setminus\{1\} \\ z\in\mathcal{I}_Z}} \end{cases},$$

and $\boldsymbol{d}^+$ is a set of within-class distances for the mini-batch of size B given N-way Z-shot learning protocol. By analogy, $\boldsymbol{d}^-$ is a set of between-class distances. Function $\mu(\cdot)$ is simply the mean over coefficients of the input vector, $\{\cdot\}$ detaches the graph during the backpropagation step, whereas $\text{TopMin}_\beta(\cdot)$ and $\text{TopMax}_{\text{NZ}\beta}(\cdot)$ return β smallest and $NZ\beta$ largest coefficients from the input vectors, respectively. Thus, Eq. (14) promotes the within-class similarity while Eq. (15) reduces the between-class similarity. Integer $\beta \geq 0$ controls the focus on difficult examples, *e.g.*, $\beta = 1$ encourages all within-class distances in Eq. (14) to be close to the positive target $\mu(\text{TopMin}_\beta(\cdot))$, the smallest observed within-class distance in the mini-batch. If $\beta > 1$, this means we relax our positive target. By analogy, if $\beta = 1$, we encourage all between-class distances in Eq. (15) to approach

Algorithm 2 Unsupervised FSAR (one training iteration by alternating over variables).

Input: $\Upsilon \equiv \{\boldsymbol{\Psi}_b\}_{b\in\mathcal{I}_B} \cup \{\boldsymbol{\Psi}'_{b,n,z}\}_{\substack{b\in\mathcal{I}_B \\ n\in\mathcal{I}_N \\ z\in\mathcal{I}_Z}}$: query/support seq. in batch; $\mathcal{F}$: EN parameters; $\boldsymbol{M}$ and $\boldsymbol{A}$; `alpha_iter` and `dic_iter`: numbers of iterations for updating $\boldsymbol{A}$ and $\boldsymbol{M}$; $\omega, \omega_{\text{DL}}$ and ω_{EN}: the learning rate for $\boldsymbol{A}$, $\boldsymbol{M}$ and $\mathcal{F}$ respectively; B: size of the mini-batch.

1: **for** i = 1, ..., `alpha_iter`: (fix $\boldsymbol{M}$ and update $\boldsymbol{A}$)
2: $\quad \boldsymbol{A} := \boldsymbol{A} - \omega\nabla_{\boldsymbol{A}}\mathcal{L}_{\text{unsup}}(\Upsilon; \boldsymbol{A}, \boldsymbol{M}, \mathcal{F})$
3: **for** i = 1, ..., `dic_iter`: (fix $\boldsymbol{A}$ and update $\boldsymbol{M}$)
4: $\quad \boldsymbol{M} := \boldsymbol{M} - \omega_{\text{DL}}\nabla_{\boldsymbol{M}}\mathcal{L}_{\text{unsup}}(\Upsilon; \boldsymbol{A}, \boldsymbol{M}, \mathcal{F})$
5: $\mathcal{F} := \mathcal{F} - \omega_{\text{EN}}\nabla_{\mathcal{F}}\mathcal{L}_{\text{unsup}}(\Upsilon; \boldsymbol{A}, \boldsymbol{M}, \mathcal{F})$ (fix $\boldsymbol{M}$ & $\boldsymbol{A}$, update $\mathcal{F}$)

Output: $\mathcal{F}$ and $\boldsymbol{M}$

the negative target $\mu(\text{TopMax}_{\text{NZ}\beta}(\cdot))$, the average over the largest NZ between-class distances. If $\beta > 1$, the negative target is relaxed.

3.4 Feature Coding and Dictionary Learning for Unsupervised FSAR

Recall from Section 1 that unsupervised FSAR forms a dictionary from the training data without the use of labels. Assigning labeled test support samples and test query into cells of a dictionary lets infer the query label by associating query with the support sample (to paraphrase, if they share the same dictionary cell, they share the class label).

In this setting, we also use a mini-batch with B episodes. Thus, B query samples and BNZ support samples give the total of $N' = B(NZ+1)$ samples per batch for feature coding and dictionary learning. Let dictionary $\boldsymbol{M} \in \mathbb{R}^{d'\cdot\tau^*\times k}$ and dictionary-coded matrix $\boldsymbol{A} \equiv [\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_{N'}] \in \mathbb{R}^{k\times N'}$. Let τ^* be set as the average number of temporal blocks over training sequences. For dictionary $\boldsymbol{M}$ and some codes $\boldsymbol{A}$, the reconstructed feature map is given as $\boldsymbol{MA} \in \mathbb{R}^{d'\cdot\tau^*\times N'}$. In what follows we reshape the reconstructed feature map so that $\boldsymbol{MA} \in \mathbb{R}^{d'\times\tau^*\times N'}$. The feature map per sequence is given as $\boldsymbol{\Psi} \in \mathbb{R}^{d'\times K\times K'\times\tau\times N'}$. All query and support sequences per batch form a set $\Upsilon \equiv \{\boldsymbol{\Psi}_b\}_{b\in\mathcal{I}_B} \cup \{\boldsymbol{\Psi}'_{b,n,z}\}_{\substack{b\in\mathcal{I}_B \\ n\in\mathcal{I}_N \\ z\in\mathcal{I}_Z}}$ with N' feature maps which we select by writing $\boldsymbol{\Psi}_i \in \Upsilon$ where $i = 1, ..., N'$. They are obtained from Encoding Network the same way as for supervised FSAR except that both query and support sequences now are equipped with $K \times K'$ viewpoints. Algorithm 2 and Figure 8 illustrate unsupervised FSAR learning with JEANIE. In short, we minimize the following loss *w.r.t.* $\mathcal{F}$, $\boldsymbol{M}$ and $\boldsymbol{A}$ by alternating over these variables:

$$\mathcal{L}_{\text{unsup}}(\Upsilon; \boldsymbol{A}, \boldsymbol{M}, \mathcal{F}) = \sum_{i=1}^{N'} d^2_{\text{JEANIE}}(\boldsymbol{\Psi}_i(\mathcal{F}), \boldsymbol{M}\boldsymbol{\alpha}_i) + \kappa\boldsymbol{\Omega}(\boldsymbol{\alpha}_i(\mathcal{F}), \boldsymbol{M}, \boldsymbol{\Psi}_i), \quad (16)$$

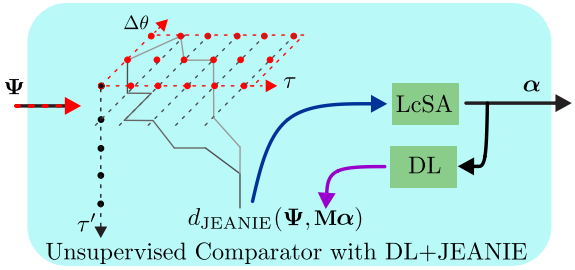


Fig. 8: Unsupervised FSAR uses the JEANIE measure as a distance between feature map $\boldsymbol{\Phi}$ of a sequence and its dictionary-based reconstruction $\boldsymbol{M\alpha}$. LcSA performs feature coding to obtain dictionary-coded $\boldsymbol{\alpha}$. DL learns the dictionary $\boldsymbol{M}$.

where $\mathcal{F} \equiv [\mathcal{F}_{MLP}, \mathcal{F}_{GNN}, \mathcal{F}_{Tr}, \mathcal{F}_{FC}]$ is the set of parameters of EN associated with $\boldsymbol{\Psi}$, that is, feature maps depend on these parameters, *i.e.*, we work with a function $\boldsymbol{\Psi}(\mathcal{F})$ not a constant.

Similarly to the Euclidean distance, $d_{\text{JEANIE}}(\cdot,\cdot)$ in Eq. (16) pursues the reconstruction of the feature map $\boldsymbol{\Psi}_i$ by the linear combination of dictionary codewords, given as $\boldsymbol{M\alpha}_i$. The reconstruction error $d^2_{\text{JEANIE}}(\boldsymbol{\Psi}_i, \boldsymbol{M\alpha}_i)$ is encouraged to be small. However, unlike the Euclidean distance, JEANIE ensures temporal and viewpoint alignment of sequences $\boldsymbol{\Psi}_i$ with the dictionary-based reconstruction $\boldsymbol{M\alpha}_i$. Constraint $\boldsymbol{\Omega}(\boldsymbol{\alpha}_i, \boldsymbol{M}, \boldsymbol{\Psi}_i)$ is a regularization term depending on the selection of feature coding method. Such a regularization encourages discriminative description, *i.e.*, similar and different feature vectors obtain similar and different dictionary-coded representations, respectively. Appendix C provides details of several feature coding and dictionary learning strategies which determine $\boldsymbol{\Omega}$. In our work, the default choice is Soft Assignment from Appendix C.1 due to its simplicity and good performance. Dictionary learning is described in Appendix C.2.

During testing, we use the trained model $\mathcal{F}$ and the learnt dictionary $\boldsymbol{M}$, pass test support and query sequences via Eq. (16) but solve only *w.r.t.* $\boldsymbol{A}$ by till $\boldsymbol{A}$ converges. Subsequently, we compare the dictionary-coded vectors of query sequences with the corresponding dictionary-coded vectors of support sequences by using some distance measure, *e.g.*, the ℓ_1 or ℓ_2 norm. We also explore the use of kernel-based distances, *e.g.*, Histogram Intersection Kernel (HIK) distance and Chi-Square Kernel (CSK) distance. The construction of the kernel distance involves a transformation from similarities to distances.

Let $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$ be some dictionary-coded vectors. Then for a kernel function $k(\boldsymbol{\alpha}, \boldsymbol{\alpha}')$, the induced distance between $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$ is given by $d(\boldsymbol{\alpha}, \boldsymbol{\alpha}') = k(\boldsymbol{\alpha}, \boldsymbol{\alpha}) + k(\boldsymbol{\alpha}', \boldsymbol{\alpha}') - 2k(\boldsymbol{\alpha}, \boldsymbol{\alpha}')$. Let $\|\boldsymbol{\alpha}\|_2 = \|\boldsymbol{\alpha}'\|_2 = 1$. The HIK distance for $k_{\text{HIK}}(\boldsymbol{\alpha}, \boldsymbol{\alpha}') = \sum_{i=1}^{d'} \min(\alpha_i, \alpha'_i)$ is given as $d_{\text{HIK}}(\boldsymbol{\alpha}, \boldsymbol{\alpha}') =$

$2-2k_{\text{HIK}}(\boldsymbol{\alpha},\boldsymbol{\alpha}')$. The CSK distance for kernel $k_{\text{CSK}}(\boldsymbol{\alpha},\boldsymbol{\alpha}')=\sum_{i=1}^{d'}\frac{2\alpha_i\alpha_i'}{\alpha_i+\alpha_i'}$ is $d_{\text{CSK}}(\boldsymbol{\alpha},\boldsymbol{\alpha}')=2-2k_{\text{CSK}}(\boldsymbol{\alpha},\boldsymbol{\alpha}')$.

The closest nearest neighbor match of test query to elements of the test support set determines the test label of the query sequence.

3.5 Fusion of Supervised and Unsupervised FSAR

Our final contribution is to introduce four simple strategies for fusing our supervised and unsupervised FSAR approaches to boost the performance. As supervised learning is label-driven and unsupervised learning is reconstruction-driven, we expect both such strategies produce complementary feature spaces amenable to fusion.

In what follows, we make use of both support and query feature maps defined over multiple viewpoints ($\boldsymbol{\Psi},\boldsymbol{\Psi}'\in\mathbb{R}^{d'\times K\times K'\times\tau}$):

$$\boldsymbol{\Psi}'\equiv f^*(\boldsymbol{\mathcal{X}}';\mathcal{F})\equiv[f(\boldsymbol{X}'_{1,1,1};\mathcal{F}),...,f(\boldsymbol{X}'_{K,K',\tau'};\mathcal{F})],$$
$$\boldsymbol{\Psi}\equiv f^*(\boldsymbol{\mathcal{X}};\mathcal{F})\equiv[f(\boldsymbol{X}_{1,1,1};\mathcal{F}),...,f(\boldsymbol{X}_{K,K',\tau};\mathcal{F})].$$

A weighted fusion of supervised and unsupervised FSAR scores. The simplest strategy is to train supervised and unsupervised FSAR models separately, and combine their predictions during testing. We call such a baseline as "weighted fusion". During the testing stage, we combine the distances of supervised and unsupervised models as follows:

$$d_{\text{fused}}=\rho\, d_{\text{JEANIE}}(\boldsymbol{\Psi}_q,\boldsymbol{\Psi}'_{n,z})+(1-\rho)d_\alpha(\boldsymbol{\alpha}_q,\boldsymbol{\alpha}'_{n,z}),\quad(17)$$

where $d_\alpha(\cdot,\cdot)$ is the distance measure for dictionary-encoded vectors, *e.g.*, the ℓ_1 norm, HIK distance or CSK distance, $0\leq\rho\leq 1$ balances the impact of supervised and unsupervised models, respectively.

Finetuning unsupervised model by supervised FSAR. For this baseline strategy, we firstly train the model using unsupervised FSAR, and then we finetune the learnt unsupervised model by using supervised FSAR. During testing stage, we evaluate on supervised learning, unsupervised learning and a fusion of both based on Eq. (17). In this case, one EN is trained which results in two sets of parameters–the first set is based on unsupervised training and the second set is based on supervised finetuning. We call this method as "finetuning unsup."

MAML-inspired fusion of supervised and unsupervised FSAR. Inspired by the success of MAML [26], we introduce a fusion strategy where the inner loop uses the unsupervised FSAR (Eq. (16)) and the outer loop uses the supervised learning loss (Eq.(14) and (15)) for the model update. Algorithm 3 details our MAML-inspired fusion strategy, called "MAML-inspired fusion".

Specifically, we start by generating representations with several viewpoints. For each mini-batch of size B we form a set with N' feature maps which are passed to Algorithm 2 which updates EN parameters $\mathcal{F}$ towards $\widehat{\mathcal{F}}$ that help accommodate unsupervised reconstruction-driven learning (so-called task-specific gradient where the task is unsupervised learning). We then recompute N' feature maps based on parameters $\widehat{\mathcal{F}}$. Finally, we apply supervised loss on such feature maps but we update now parameters $\mathcal{F}$ which means that parameters $\mathcal{F}$ are tuned for the global label-driven task with help of unsupervised task.

Intuitively, it is a second-order gradient model. Specifically, one takes the gradient step in the direction pointed by the unsupervised loss to obtain task-specific EN parameters, and then given these task-specific parameters, task-specific feature maps are extracted to and passed into the supervised loss to perform the gradient descent step in the direction pointed by the unsupervised loss to obtain update of global EN parameters.

Fusion by alignment of supervised and unsupervised feature maps. Inspired by domain adaptation, Algorithm 4 in Appendix D is an easy-to-interpret simplification (called "adaptation-based") of the above MAML-inspired fusion. Instead of complex gradient interplay between unsupervised and supervised loss functions, we explicitly align "supervised" feature maps towards "unsupervised" feature maps.

Algorithm 3 Fusion of Supervised and Unsupervised FSAR by MAML-inspired Setting (one training iteration).

Input: $\Gamma\equiv\{\boldsymbol{\mathcal{X}}_b\}_{b\in\mathcal{I}_B}\cup\{\boldsymbol{\mathcal{X}}'_{b,n,z}\}_{\substack{b\in\mathcal{I}_B\\n\in\mathcal{I}_N\\z\in\mathcal{I}_Z}}$: query/support blocks in batch; $\mathcal{F}$: EN parameters; $\boldsymbol{M}$ and $\boldsymbol{A}$; `alpha_iter` and `dic_iter`: numbers of iterations for updating $\boldsymbol{A}$ and $\boldsymbol{M}$; $\omega,\omega_{\text{DL}}$ and ω_{EN}: the learning rate for $\boldsymbol{A}$, $\boldsymbol{M}$ and $\mathcal{F}$ respectively; B: size of the mini-batch.

$\Upsilon\equiv\{\boldsymbol{\Psi}_b\}_{b\in\mathcal{I}_B}\cup\{\boldsymbol{\Psi}'_{b,n,z}\}_{\substack{b\in\mathcal{I}_B\\n\in\mathcal{I}_N\\z\in\mathcal{I}_Z}}$ where $\begin{cases}\boldsymbol{\Psi}_b=f^*(\boldsymbol{\mathcal{X}}_b;\mathcal{F})\\ \boldsymbol{\Psi}'_{b,n,z}=f^*(\boldsymbol{\mathcal{X}}'_{b,n,z};\mathcal{F})\end{cases}$
(obtain feature maps for global parameters $\mathcal{F}$)

$(\widehat{\mathcal{F}},\boldsymbol{M})=\text{Algorithm2}(\Upsilon,\mathcal{F},\boldsymbol{M},\boldsymbol{A},$ `alpha_iter`, `dic_iter`, $\omega,\omega_{\text{DL}},\omega_{\text{EN}})$ (unsupervised FSAR)

$\widehat{\Upsilon}\equiv\{\widehat{\boldsymbol{\Psi}}_b\}_{b\in\mathcal{I}_B}\cup\{\widehat{\boldsymbol{\Psi}}'_{b,n,z}\}_{\substack{b\in\mathcal{I}_B\\n\in\mathcal{I}_N\\z\in\mathcal{I}_Z}}$ where $\begin{cases}\widehat{\boldsymbol{\Psi}}_b=f^*(\boldsymbol{\mathcal{X}}_b;\widehat{\mathcal{F}})\\ \widehat{\boldsymbol{\Psi}}'_{b,n,z}=f^*(\boldsymbol{\mathcal{X}}'_{b,n,z};\widehat{\mathcal{F}})\end{cases}$
(obtain feature maps for parameters $\widehat{\mathcal{F}}$ from the unsupervised step)

$\widehat{\boldsymbol{d}}^+=[d_{\text{JEANIE}}(\widehat{\boldsymbol{\Psi}}_b,\widehat{\boldsymbol{\Psi}}'_{b,1,z})]_{\substack{b\in\mathcal{I}_B\\z\in\mathcal{I}_Z}}$ (within-class distance)

$\widehat{\boldsymbol{d}}^-=[d_{\text{JEANIE}}(\widehat{\boldsymbol{\Psi}}_b,\widehat{\boldsymbol{\Psi}}'_{b,n,z})]_{\substack{b\in\mathcal{I}_B\\n\in\mathcal{I}_N\setminus\{1\}\\z\in\mathcal{I}_Z}}$ (between-class distance)

$\mathcal{F}:=\mathcal{F}-\omega_{\text{EN}}\nabla_{\mathcal{F}}l(\widehat{\boldsymbol{d}}^+,\widehat{\boldsymbol{d}}^-)$

Output: $\mathcal{F}$ and $\boldsymbol{M}$

4 Experiments

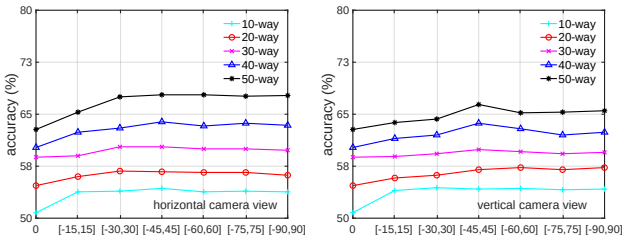
4.1 Datasets and Protocols

Below, we describe the datasets and evaluation protocols on which we validate our FSAR with JEANIE.

i. *UWA3D Multiview Activity II* [68] contains 30 actions performed by 9 people in a cluttered environment. The Kinect camera was used in 4 distinct views: front view (V_1), left view (V_2), right view (V_3), and top view (V_4).
ii. *NTU RGB+D (NTU-60)* [70] contains 56,880 video sequences and over 4 million frames. This dataset has variable sequence lengths and high intra-class variations.
iii. *NTU RGB+D 120 (NTU-120)* [52] contains 120 action classes (daily/health-related), and 114,480 RGB+D video samples captured with 106 distinct human subjects from 155 different camera viewpoints.
iv. *Kinetics* [37] is a large-scale collection of 650,000 video clips that cover 400/600/700 human action classes. It includes human-object interactions such as *playing instruments*, as well as human-human interactions such as *shaking hands* and *hugging*. As the Kinetics-400 dataset provides only the raw videos, we follow approach [109] and use the estimated joint locations in the pixel coordinate system as the input to our pipeline. To obtain the joint locations, we first resize all videos to the resolution of 340 $\times$ 256, and convert the frame rate to 30 FPS. Then we use the publicly available *OpenPose* [8] toolbox to estimate the location of 18 joints on every frame of the clips. As OpenPose produces the 2D body joint coordinates and Kinetics-400 does not offer multiview or depth data, we use a network of Martinez *et al.*[60] pre-trained on Human3.6M [10], combined with the 2D OpenPose output to estimate 3D coordinates from 2D coordinates. The 2D OpenPose and the latter network give us (x, y) and z coordinates, respectively.

Evaluation protocols. For the UWA3D Multiview Activity II, we use standard multi-view classification protocol [68, 90], but we apply it to one-shot learning as the view combinations for training and testing sets are disjoint. For NTU-120, we follow the standard one-shot protocol [52]. Based on this protocol, we create a similar one-shot protocol for NTU-60, with 50/10 action classes used for training/testing respectively. To evaluate the effectiveness of the proposed method on viewpoint alignment, we also create two new protocols on NTU-120, for which we group the whole dataset based on (i) horizontal camera views into left, center and right views, (ii) vertical camera views into top, center and bottom views. We conduct two sets of experiments on such disjoint view-wise splits (i) (*100/same 100*): using 100 action classes for training, and testing on the same 100 action classes (ii) (*100/novel 20*): training on 100 action classes but testing on the rest unseen 20 classes.

Stereo projections. For simulating different camera viewpoints, we estimate the fundamental matrix F (Eq. (19)), which relies on camera parameters. Thus, we use the Camera Calibrator from MATLAB to estimate intrinsic, extrinsic and lens distortion parameters. For a given skeleton dataset,



(a) (b)

Fig. 9: The impact of viewing angles in (a) horizontal and (b) vertical camera views on NTU-60.

Table 1: Experimental results on NTU-60 (left) and NTU-120 (right) for different camera viewpoint simulations.

# Training Classes	NTU-60					NTU-120				
	10	20	30	40	50	20	40	60	80	100
Euler simple ($K+K'$)	54.3	56.2	60.4	64.0	68.1	30.7	36.8	39.5	44.3	46.9
Euler ($K\times K'$)	**60.8**	67.4	67.5	70.3	**75.0**	32.9	39.2	43.5	48.4	50.2
CamVPC ($K\times K'$)	59.7	**68.7**	**68.4**	**70.4**	73.2	**33.1**	**40.8**	**43.7**	**48.4**	**51.4**

we compute the range of spatial coordinates x and y, respectively. We then split them into 3 equally-sized groups to form roughly left, center, right views and other 3 groups for bottom, center, top views. We choose $\sim$15 frame images from each corresponding group, upload them to the Camera Calibrator, and export camera parameters. We then compute the average distance/depth and height per group to estimate the camera position. On NTU-60 and NTU-120, we simply group the whole dataset into 3 cameras, which are left, center and right views, as provided in [52], and then we compute the average distance per camera view based on the height and distance settings given in the table in [52].

4.2 Ablation Studies

We start our experiments by investigating various architectural choices and key hyperparameters of our model.

Camera viewpoint simulations. We choose 15 degrees as the step size for the viewpoints simulation. The ranges of camera azimuth and altitude are in $[-90^0, 90^0]$. Where stated, we perform a grid search on camera azimuth and altitude with Hyperopt [5]. Below, we explore the choice of the angle ranges for both horizontal and vertical views. Fig. 9a and 9b (evaluations on the NTU-60 dataset) show that the angle range $[-45°, 45°]$ performs the best, and widening the range in both views does not increase the performance any further. Table 1 shows results for the chosen range $[-45°, 45°]$ of camera viewpoint simulations. (*Euler simple ($K+K'$)*) denotes a simple concatenation of features from both horizontal and vertical views, whereas (*Euler ($K\times K'$)*) and (*CamVPC($K\times K'$)*) represent the grid search of all possible views. The table shows that Euler angles for the viewpoint augmentation outperform (*Euler simple*), and

Table 2: The impact of the number of frames M in temporal block under stride step S on results (NTU-60). $S=pM$, where $1-p$ describes the temporal block overlap percentage. Higher p means fewer overlap frames between temporal blocks.

	$S=M$		$S=0.8M$		$S=0.6M$		$S=0.4M$		$S=0.2M$	
M	50-cls	20-cls	50-cls	20-cls	50-cls	20-cls	50-cls	20-cls	50-cls	20-cls
5	69.0	55.7	71.8	57.2	69.2	59.6	73.0	60.8	71.2	61.2
6	69.4	54.0	65.4	54.1	67.8	58.0	72.0	57.8	**73.0**	**63.0**
8	67.0	52.7	67.0	52.5	**73.8**	**61.8**	67.8	60.3	68.4	59.4
10	62.2	44.5	63.6	50.9	65.2	48.4	62.4	57.0	70.4	56.7
15	62.0	43.5	62.6	48.9	64.7	47.9	62.4	57.2	68.3	56.7
30	55.6	42.8	57.2	44.8	59.2	43.9	58.8	55.3	60.2	53.8
45	50.0	39.8	50.5	40.6	52.3	39.9	53.0	42.1	54.0	45.2

(*CamVPC*) (viewpoints of query sequences are generated by the stereo projection geometry) outperforms Euler angles in almost all the experiments on NTU-60 and NTU-120. This proves the effectiveness of using the stereo projection geometry for the viewpoint augmentation.

Block size M and stride size S. Recall from Figure 1, that each skeleton sequence is divided into short-term temporal blocks which may also partially overlap.

Table 2 shows evaluations *w.r.t.* block size M and stride S, and indicates that the best performance (both 50-class and 20-class settings) is achieved for smaller block size (frame count in the block) and smaller stride. Longer temporal blocks decrease the performance due to the temporal information not reaching the temporal alignment step of JEANIE. Our block encoder encodes each temporal block for learning the local temporal motions, and aggregate these block features finally to form the global temporal motion cues. Smaller stride helps capture more local motion patterns. Considering the accuracy-runtime trade-off, we choose $M=8$ and $S=0.6M$ for the remaining experiments.

GNN as a block of Encoding Network. Recall from Section 3.1 and Appendix B that our Encoding Network uses a GNN block. For that reason, we investigate several models with the goal of justifying our default choice.

We conduct experiments on 4 GNNs listed in Table 3. S^{2}GC performs the best on large-scale NTU-60 and NTU-120, APPNP outperforms SGC, and SGC outperforms GCN. We also notice that using GNN as a projection layer performs better than single FC layer used in standard transformer by ∼5%. We note that using the RBF-induced distance for $d_{base}(\cdot,\cdot)$ of JEANIE outperforms the Euclidean distance. We choose S^{2}GC as a block of our Encoding Network and we use the RBF-induced base distance for JEANIE and other DTW-based models.

ι-max shift. Recall from Section 3.2 that the ι-max controls the smoothness of alignment.

Table 4 shows the evaluations of ι for the maximum shift. We notice that $\iota=2$ yields the best results for all the experimental settings on both NTU-60 and NTU-120. Increasing ι does not help improve the performance. We think

Table 3: Evaluations of GNN (block of Encoding Network).

	FC layer	GCN	SGC	APPNP	S^{2}GC (Eucl.)	S^{2}GC (RBF)
NTU-60 (50-class)	51.2	56.0	68.1	68.5	75.6	**78.1**
NTU-120 (20-class)	23.3	27.9	30.7	30.8	34.5	**37.2**

Table 4: Experimental results on NTU-60 (left) and NTU-120 (right) for ι-max shift.

	NTU-60					NTU-120				
	10	20	30	40	50	20	40	60	80	100
$\iota=1$	60.8	70.7	72.5	72.9	75.2	36.3	42.5	48.7	**50.0**	54.8
$\iota=2$	**63.8**	**72.9**	**74.0**	**73.4**	**78.1**	**37.2**	**43.0**	**49.2**	**50.0**	**55.2**
$\iota=3$	55.2	58.9	65.7	67.1	72.5	36.7	**43.0**	48.5	49.0	54.9
$\iota=4$	54.5	57.8	63.5	65.2	70.4	36.5	42.9	48.3	48.9	54.3

ι relies on (i) the speeds of action execution (ii) the temporal block size M and the stride S.

4.3 Implementation Details

Before we discuss our main experimental results, below we provide network configurations and training details.

Network configurations. Given the temporal block size M (the number of frames in a block) and desired output size d, the configuration of the 3-layer MLP unit is: FC ($3M \rightarrow 6M$), LayerNorm (LN) as in [17], ReLU, FC ($6M \rightarrow 9M$), LN, ReLU, Dropout (for smaller datasets, the dropout rate is 0.5; for large-scale datasets, the dropout rate is 0.1), FC ($9M \rightarrow d$), LN. Note that M is the temporal block size and d is the output feature dimension per body joint.

Transformer block. The hidden size of our transformer (the output size of the first FC layer of the MLP in Eq. (7)) depends on the dataset. For smaller scale datasets, the depth of the transformer is $L_{\text{tr}}=6$ with 64 as the hidden size, and the MLP output size is $d=32$ (note that the MLP which provides $\widehat{\mathbf{X}}$ and the MLP in the transformer must both have the same output size). For NTU-60, the depth of the transformer is $L_{\text{tr}}=6$, the hidden size is 128 and the MLP output size is $d=64$. For NTU-120, the depth of the transformer is $L_{\text{tr}}=6$, the hidden size is 256 and the MLP size is $d=128$. For Kinetics-skeleton, the depth for the transformer is $L_{\text{tr}}=12$, hidden size is 512 and the MLP output size is $d=256$. The number of heads for the transformer of UWA3D Multiview Activity II, NTU-60, NTU-120 and Kinetics-skeleton is set as 6, 12, 12 and 12, respectively. The output size d' of the final FC layer in Eq. (9) are 50, 100, 200, and 500 for UWA3D Multiview Activity II, NTU-60, NTU-120 and Kinetics-skeleton, respectively.

Training details. The parameters (weights) of the pipeline are initialized with the normal distribution (zero mean and unit standard deviation). We use 1e-3 as the learning rate, and the weight decay is set to 1e-6. We use the SGD optimizer. We set the number of training episodes to 100K

Table 5: Results on NTU-60 (all use S^{2}GC). All methods enjoy temporal alignment by soft-DTW or JEANIE (joint temporal and viewpoint alignment) except where indicated otherwise. We use the ℓ_2 norm for comparing the codes in unsupervised setting with soft-DTW. For unsupervised JEANIE, the distance for comparing the codes is indicated.

		viewpoint simulation	align.	10	20	30	40	50
Sup.	Matching Nets [85]			46.1	48.6	53.3	56.2	58.8
	Matching Nets [85]		2V	47.2	50.7	55.4	57.7	60.2
	ProtoNet [75]			47.2	51.1	54.3	58.9	63.0
	ProtoNet [75]		2V	49.8	53.1	56.7	60.9	64.3
	TAP [77]			54.2	57.3	61.7	64.7	68.3
	Each frame to frontal view	-	-	52.9	53.3	54.6	54.2	58.3
	Each block to frontal view	-	-	53.9	56.1	60.1	63.8	68.0
	Traj. aligned (video-level)	-	-	36.1	40.3	44.5	48.0	50.2
	Traj. aligned (block-level)	-	-	52.9	55.8	59.4	63.6	66.7
	No soft-DTW (S^{2}GC)	-	-	50.8	54.7	58.8	60.2	62.8
	soft-DTW	-	T	53.7	56.2	60.0	63.9	67.8
	JEANIE	Euler	T+V	54.0	56.0	60.2	63.8	67.8
	JEANIE (simple concat.)	Euler	T+2V	54.3	56.2	60.4	64.0	68.1
	JEANIE	Euler	T+2V	60.8	67.4	67.5	70.3	75.0
	JEANIE	CamVPC	T+2V	59.7	68.7	68.4	70.4	73.2
	JEANIE (+crossval.)	CamVPC	T+2V	63.4	72.4	73.5	73.2	78.1
	JEANIE (+crossval. +Transf.)	CamVPC	T+2V	**65.0**	**75.2**	**76.7**	**78.9**	**80.0**
Unsup. +Transf.	soft-DTW (HA)	-	T	16.3	23.7	28.3	31.8	33.1
	soft-DTW (SC)	-	T	18.7	26.0	31.6	34.2	38.1
	soft-DTW (SC$_+$)	-	T	18.5	25.7	30.0	33.9	37.9
	soft-DTW (LLC)	-	T	23.1	30.1	33.0	36.4	40.9
	soft-DTW (SA)	-	T	25.4	31.7	34.6	38.0	41.7
	soft-DTW (LcSA)	-	T	25.9	32.3	35.1	38.5	42.3
	JEANIE (LLC)–ℓ_1	CamVPC	T+2V	27.5	33.6	36.0	41.6	44.5
	JEANIE (LLC)–ℓ_2	CamVPC	T+2V	27.8	33.9	36.5	41.7	44.7
	JEANIE (LLC)–HIK	CamVPC	T+2V	28.0	33.6	36.8	42.0	45.1
	JEANIE (LLC)–CSK	CamVPC	T+2V	27.8	33.9	36.8	41.7	45.0
	JEANIE (LcSA)–ℓ_1	CamVPC	T+2V	29.0	35.6	39.5	44.8	47.5
	JEANIE (LcSA)–ℓ_2	CamVPC	T+2V	**29.1**	**35.8**	39.7	45.2	**48.0**
	JEANIE (LcSA)–HIK	CamVPC	T+2V	28.8	**35.8**	39.7	**45.0**	47.7
	JEANIE (LcSA)–CSK	CamVPC	T+2V	29.0	**35.8**	**40.0**	**45.0**	**48.0**
	FVM (LcSA)–CSK	CamVPC	T+2V	27.0	33.4	36.5	42.0	45.1
Fusion +Transf.	Weighted fusion	CamVPC	T+2V	66.5	76.9	79.0	81.2	81.5
	Finetuning unsup.	CamVPC	T+2V	67.0	77.2	79.9	82.0	84.5
	MAML-inspired fusion	CamVPC	T+2V	**70.0**	**78.3**	**81.0**	**82.9**	**85.0**
	Adaptation-based	CamVPC	T+2V	69.8	78.2	80.7	82.3	84.8

for NTU-60, 200K for NTU-120, 500K for 3D Kinetics-skeleton, and 10K for UWA3D Multiview Activity II. We use Hyperopt [5] for hyperparameter search on validation sets for all the datasets.

4.4 Discussion on Supervised Few-shot Action Recognition

NTU-60. Table 5 (*Sup.*) shows that using the viewpoint alignment simultaneously in two dimensions, x and y for Euler angles, or azimuth and altitude the stereo projection geometry (*CamVPC*), improves the performance by 5–8% compared to (*Euler*) with a simple concatenation of viewpoints, a variant where the best viewpoint alignment path was chosen from the best alignment path along x and the best alignment path along y. Euler with (simple concat.) is better than Euler with y rotations only ((*V*) includes rotations along y while (*2V*) includes rotations along two axes). We indicate where temporal alignment (*T*) is also used. When we use HyperOpt [5] to search for the best angle range in which we perform the viewpoint alignment (*CamVPC*), the results improve further. Enabling the viewpoint alignment for support sequences (*CamVPC*) yields extra improvement, and our best variant of JEANIE boosts the performance by $\sim$2%.

Table 6: Experimental results on NTU-120 (S^{2}GC backbone). All methods enjoy temporal alignment by soft-DTW or JEANIE (joint temporal and viewpoint alignment) except VA [117, 118] and other cited works. For VA*, we used soft-DTW on temporal blocks while VA generated temporal blocks. For unsupervised soft-DTW and JEANIE, the best distance for comparing the codes is indicated. For brevity, we list unsupervised variants on LcSA but Table 11 in Appendix E contains all variants.

		viewpoint simulation	align.	20	40	60	80	100
Sup.	APSR [52]			29.1	34.8	39.2	42.8	45.3
	SL-DML [62]			36.7	42.4	49.0	46.4	50.9
	Skeleton-DML [61]			28.6	37.5	48.6	48.0	54.2
	ProtoNet+VA-RNN(aug.) [117]			25.3	28.6	32.5	35.2	38.0
	ProtoNet+VA-CNN(aug.) [118]			29.7	33.0	39.3	41.5	42.8
	ProtoNet+VA-fusion(aug.) [118]			29.8	33.2	39.5	41.7	43.0
	ProtoNet+VA*-fusion(aug.) [118]			33.3	38.7	45.2	46.3	49.8
	TAP [77]			31.2	37.7	40.9	44.5	47.3
	ALCA-GCN [128]			38.7	46.6	51.0	53.7	57.6
	No soft-DTW (S^{2}GC)	-	-	30.0	35.9	39.2	43.6	46.4
	soft-DTW	-	T	30.3	37.2	39.7	44.0	46.8
	JEANIE	Euler	T+V	30.6	36.7	39.2	44.0	47.0
	JEANIE (simple concat.)	Euler	T+2V	30.7	36.8	39.5	44.3	46.9
	JEANIE	Euler	T+2V	32.9	39.2	43.5	48.4	50.2
	JEANIE	CamVPC	T+2V	33.1	40.8	43.7	48.4	51.4
	JEANIE (+crossval.)	CamVPC	T+2V	37.2	43.0	49.2	50.0	55.2
	FVM (+crossval. +Transf.)	CamVPC	T+2V	34.5	41.9	44.2	48.7	52.0
	JEANIE (+crossval. +Transf.)	CamVPC	T+2V	**38.5**	**44.1**	**50.3**	**51.2**	**57.0**
Unsup. +Transf.	soft-DTW (LcSA)–ℓ_2	-	T	15.7	21.4	25.2	32.0	40.2
	JEANIE (LcSA)–CSK	CamVPC	T+2V	**18.6**	**25.2**	**32.0**	**39.6**	**48.5**
	FVM (LcSA)–CSK	CamVPC	T+2V	17.5	22.4	30.7	36.1	44.5
Fusion +Transf.	Weighted fusion	CamVPC	T+2V	44.4	48.6	50.8	52.0	58.3
	Finetuning unsup.	CamVPC	T+2V	45.6	50.8	53.0	55.0	60.2
	MAML-inspired fusion	CamVPC	T+2V	**48.2**	**53.3**	**57.0**	**60.3**	**62.1**
	Adaptation-based	CamVPC	T+2V	47.9	53.0	56.5	60.0	61.9

We also show that aligning query and support trajectories by the angle of torso 3D joint, denoted (*Traj. aligned*) are not very powerful. We note that aligning piece-wise parts (blocks) is better than trying to align entire trajectories. In fact, aligning individual frames by torso to the frontal view (*Each frame to frontal view*) and aligning block average of torso direction to the frontal view (*Each block to frontal view*) were marginally better. We note these baselines use soft-DTW.

NTU-120. Table 6 (*Sup.*) shows that our proposed method achieves the best results on NTU-120, and outperforms the recent SL-DML and Skeleton-DML by 6.1% and 2.8% respectively (100 training classes). Note that Skeleton-DML requires the pre-trained model for the weights initialization whereas our proposed model with JEANIE is fully differentiable. For comparisons, we extended the view adaptive neural networks [118] by combining them with ProtoNet [75]. VA-RNN+VA-CNN [118] uses 0.47M+24M parameters with random rotation augmentations while JEANIE uses 0.25–0.5M params. Their *rotation+translation* keys are not proven to perform smooth optimal alignment as JEANIE. In contrast, d_{JEANIE} performs jointly a smooth viewpoint-temporal alignment with smoothness by design. They also use Euler angles which are a worse option (see Table 5 and 6) than the camera projection of JEANIE. We notice that ProtoNet+VA backbones is 12% worse than our JEANIE. Even if we split skeletons into blocks to let soft-DTW per-

Table 7: Experiments on 2D and 3D Kinetics-skeleton. Note that we have no results on JEANIE or FVM for 2D coordinates as these require very different viewpoint modeling than 3D coordinates. For brevity, we list unsupervised variants on LcSA but Table 12 in Appendix E contains more variants.

		viewpoint simulation	alignment	2D skel.	3D skel.
Sup.	No soft-DTW(S^2GC)	-	-	32.8	35.9
	soft-DTW	-	T	34.7	39.6
	FVM	Euler	T+2V	-	44.1
	JEANIE	Euler	T+2V	-	50.3
	JEANIE(+Transf.)	Euler	T+2V	-	52.5
	JEANIE(+Transf.)	CamVPC	T+2V	-	**52.8**
Unsup. +Transf.	soft-DTW(LcSA)–ℓ_2	-	T	19.3	22.2
	JEANIE (LcSA)–CSK	CamVPC	T+2V	-	**28.3**
	FVM (LcSA)–ℓ_2	CamVPC	T+2V	-	25.1
Fusion +Transf.	Weighted fusion	CamVPC	T+2V	-	53.3
	Finetuning unsup.	CamVPC	T+2V	-	54.2
	MAML-inspired fusion	CamVPC	T+2V	-	**57.0**
	Adaptation-based	CamVPC	T+2V	-	56.3

form temporal alignment of prototypes & query, JEANIE is still 4–6% better. Notice also that JEANIE with transformer is between 3% and 6% better than JEANIE with no transformer, which validates the use of transformer on large datasets.

Kinetics-skeleton. We evaluate our proposed model on both 2D and 3D Kinetics-skeleton. We split the whole dataset into 200 actions for training, and the rest half for testing. Table 7 shows that using 3D skeletons outperforms the use of 2D skeletons by 3–4%. The temporal alignment only (with soft-DTW) outperforms baseline (without alignment) by ∼2% and 3% on 2D and 3D skeletons respectively, and JEANIE outperforms the temporal alignment only by around 5%. Our best variant with JEANIE further boosts results by 2%. We notice that the improvements for the use of camera viewpoint simulation (*CamVPC*) compared to the use of Euler angles are limited, around 0.3% and 0.6% for JEANIE and FVM respectively. The main reason is that the Kinetics-skeleton is a large-scale dataset collected from YouTube videos, and the camera viewpoint simulation becomes unreliable especially when videos are captured by multiple different devices, *e.g.*, camera and mobile phone.

4.5 Discussion on Unsupervised Few-shot Action Recognition

Recall from Section 3.4 that JEANIE can help train unsupervised FSAR by forming a dictionary that relies on temporal-viewpoint alignment of JEANIE which factors out nuisance temporal and pose variations in sequences.

However, the choice of feature coding and dictionary learning method can affect the performance of unsupervised learning. Thus, we investigate several variants from Appendix C.

Table 5 (*Unsup.*) and Table 11 in Appendix E (extension of Table 6 (*Unsup.*)) show on NTU-60 and NTU-120 that the LcSA coder performs better than SA by ∼0.6% and 1.5% and SA outperforms LLC by ∼1.5% and 2%. As LcSA and SA are based on the non-linear sigmoid-like reconstruction functions, we suspect they are more robust than linear reconstruction function of LLC. Since the LcSA is the best performer in our experiments followed by SA and LLC or SC, we choose LcSA for further analysis.

Table 5 (*Unsup.*) and Tables 11 and 12 in Appendix E (extensions of Tables 6 (*Unsup.*) and 7 (*Unsup.*)) also show that the choose of different distance measures for comparing the dictionary-coded vectors of sequences during the test stage do not affect the performance too much. The kernel-induced distances, *e.g.*, HIK distance and CSK distance and ℓ_2-norm perform similarly, and they outperform the ℓ_1 norm by ∼0.5% on average. We choose the CSK distance for unsupervised JEANIE with LcSA as the default distance for comparing dictionary-coded vectors as it was marginally better performer in the majority of experiments.

Tables 5 (*Unsup.*), 6 (*Unsup.*) and 7 (*Unsup.*) show that unsupervised JEANIE (temporal-viewpoint alignment) outperforms soft-DTW (temporal alignment only) by up to 5%, 9% and 6% on NTU-60, NTU-120 and Kinetics-skeleton, respectively. Table 8 (*Unsup.*) shows that the biggest improvement is obtained when using unsupervised JEANIE on UWA3D Multiview Activity II dataset, with 10% performance gain. This outlines the importance of the joint temporal-viewpoint alignment under heavy camera pose variations.

Interestingly, FVM in unsupervised learning performs worse compared to our JEANIE, *e.g.*, JEANIE suppresses FVM by ∼3%, 4% and 3% respectively on NTU-60, NTU-120 and Kinetics-skeleton in Tables 5 (*Unsup.*), 6 (*Unsup.*) and 7 (*Unsup.*). On UWA3D Multiview Activity II in Table 8 (*Unsup.*), JEANIE outperforms FVM by more than 5%. This is because FVM always seeks the best local viewpoint alignment for every step of soft-DTW which realizes a non-smooth temporal-viewpoint path in contrast to JEANIE. Without the guidance of label information, FVM fails to capture the corresponding relationships between each temporal and viewpoint alignment. Thus, FVM produces a worse dictionary than JEANIE which validates the need for factoring out jointly temporal and viewpoint nuisance variations from sequences.

Table 9 (*Unsup.*) shows that on our newly introduced multiview classification protocol on NTU-120, for the unsupervised learning experiments, JEANIE outperforms the baseline (temporal alignment only with soft-DTW) by 7% and 8% on average on (*100/same 100*) and (*100/novel 20*) respectively . Moreover, JEANIE outperforms the FVM by around 4% and 3% on (*100/same 100*) and (*100/novel 20*) respectively.

Table 8: Experiments on the UWA3D Multiview Activity II. All with S^{2}GC layer unless specified.

		align.	Train	V_1 & V_2		V_1 & V_3		V_1 & V_4		V_2 & V_3		V_2 & V_4		V_3 & V_4		Mean
			Test	V_3	V_4	V_2	V_4	V_2	V_3	V_1	V_4	V_1	V_3	V_1	V_2	
Sup.	GCN	-		36.4	26.2	20.6	30.2	33.7	22.4	43.1	26.6	16.9	12.8	26.3	36.5	27.6
	SGC	-		40.9	60.3	44.1	52.6	48.5	38.7	50.6	52.8	52.8	37.2	57.8	49.6	48.8
	+soft-DTW	T		43.9	60.8	48.1	54.6	52.6	45.7	54.0	58.2	56.7	40.2	60.2	51.1	52.2
	+JEANIE	T+2V		47.0	62.8	50.4	57.8	53.6	47.0	57.9	62.3	57.0	44.8	61.7	52.3	54.6
	APPNP	-		42.9	61.9	47.8	58.7	53.8	44.0	52.3	60.3	55.1	38.2	58.3	47.9	51.8
	+soft-DTW	T		44.3	63.2	50.7	62.3	53.9	45.0	56.9	62.8	56.4	39.3	60.1	51.9	53.9
	+JEANIE	T+2V		46.8	64.6	51.3	65.1	54.7	46.4	58.2	65.1	58.8	43.9	60.3	52.5	55.6
	S^{2}GC	-		45.5	64.4	46.8	61.6	49.5	43.2	57.3	61.2	51.0	42.9	57.0	49.2	52.5
	+soft-DTW	T		48.2	67.2	51.2	67.0	53.2	46.8	62.4	66.2	57.8	45.0	62.2	53.0	56.7
	+FVM	T+2V		50.7	68.8	56.3	69.2	55.8	47.1	63.7	68.8	62.5	51.4	63.8	55.7	59.5
	+JEANIE	T+2V		**55.3**	**70.2**	**61.4**	**72.5**	**60.9**	**50.8**	**66.4**	**73.9**	**68.8**	**57.2**	**66.7**	**60.2**	**63.7**
Unsup.	soft-DTW(LcSA)–ℓ_2	T		40.5	41.4	40.2	43.6	38.2	39.9	38.2	40.2	41.5	39.7	40.9	38.8	40.3
	JEANIE(LcSA)–CSK	T+2V		**53.0**	**52.5**	**50.1**	**51.0**	**47.6**	**49.2**	**49.5**	**52.3**	**51.3**	**49.0**	**49.2**	**47.1**	**50.2**
	FVM(LcSA)–CSK	T+2V		46.2	44.0	45.1	48.0	43.5	44.1	43.8	46.0	47.2	43.5	45.8	43.1	45.0
Fusion	Weighted fusion	T+2V		64.9	70.4	63.9	73.4	62.1	57.3	67.8	74.1	69.7	61.3	68.9	63.2	66.4
	Finetuning unsup.	T+2V		73.3	70.8	68.8	74.0	62.7	61.7	69.4	74.3	71.1	67.9	72.1	65.8	69.3
	MAML-inspired fusion	T+2V		**78.7**	**73.9**	**72.7**	**75.9**	**65.8**	**70.9**	**74.3**	**76.2**	**77.9**	**77.3**	**80.2**	**73.0**	**74.7**
	Adaptation-based	T+2V		76.3	71.5	72.0	75.0	**65.8**	69.2	72.8	75.5	75.9	76.5	78.3	71.7	73.4

Table 9: Results on NTU-120 (multiview classification). We use S^{2}GC.

	Eval. Protocol	Train	bott. cent.	bott. top	bott.& cent. top	left cent.	left right	left & cent. right
Sup.	100/same 100	soft-DTW	74.2	73.8	75.0	58.3	57.2	68.9
		FVM	79.9	78.2	80.0	65.9	63.9	75.0
		JEANIE	**81.5**	**79.2**	**83.9**	**67.7**	**66.9**	**79.2**
	100/novel 20	soft-DTW	58.2	58.2	61.3	51.3	47.2	53.7
		FVM	66.0	65.3	68.2	58.8	53.9	60.1
		JEANIE	**67.8**	**65.8**	**70.8**	**59.5**	**55.0**	**62.7**
Unsup.	100/same 100	soft-DTW	55.6	53.9	56.1	40.9	39.7	47.3
		FVM	57.8	58.0	59.7	47.9	43.1	48.8
		JEANIE	**60.3**	**61.7**	**63.2**	**51.7**	**46.9**	**52.5**
	100/novel 20	soft-DTW	40.2	39.7	40.8	33.7	32.9	45.5
		FVM	46.2	44.5	47.0	38.1	34.0	47.1
		JEANIE	**48.8**	**47.2**	**50.0**	**41.0**	**39.7**	**51.8**
Fusion	100/same 100	Weighted fusion	82.8	80.2	84.6	68.3	67.4	79.7
		Finetuning unsup.	83.2	81.0	86.0	69.7	68.9	80.5
		MAML-inspired fusion	**85.3**	**83.2**	**87.1**	**72.2**	**71.7**	**82.3**
		Adaptation-based	85.0	82.4	86.8	71.3	69.8	81.0
	100/novel 20	Weighted fusion	68.7	66.3	71.2	60.4	55.9	63.3
		Finetuning unsup.	69.2	67.3	72.8	61.1	56.8	64.6
		MAML-inspired fusion	**72.3**	**69.0**	**74.9**	**63.0**	**58.7**	**67.1**
		Adaptation-based	71.9	68.1	73.3	62.7	56.9	66.0

Table 10: Evaluation of different testing strategies, *e.g.*, with supervised learning, unsupervised learning and a combination of both on Kinetics-skeleton when the model is trained with the fusion of both supervised and unsupervised FSAR.

Train with fusion	# ENs	Different test cases: sup. only	unsup. only	sup.+unsup.
Weighted fusion	2	52.8	28.3	53.3
Finetuning unsup.	1	53.1 (↑0.6)	40.7 (↑12.4)	54.2 (↑0.9)
Adaptation-based	1	53.7 (↑1.2)	49.6 (↑21.3)	56.3 (↑3.0)
MAML-inspired fusion	1	54.0 (↑1.5)	50.3 (↑22.0)	57.0 (↑3.7)

4.6 Discussion on JEANIE and FVM

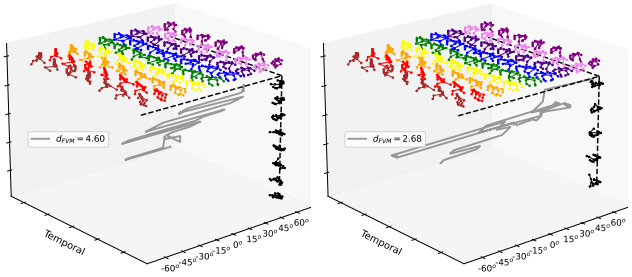
For supervised learning, JEANIE outperforms FVM by 2-4% on NTU-120, and outperforms FVM by around 6% on Kinetics-skeleton. For unsupervised learning, JEANIE improves the performance by around 3% on average on NTU-60, NTU-120 and Kinetics-skeleton. On UWA3D Multiview Activity II, JEANIE suppresses FVM by 4% and 5% respectively for supervised and unsupervised experiments. This shows that seeking jointly the best temporal-viewpoint alignment is more valuable than considering viewpoint alignment as a separate local alignment task (free range alignment per each step of soft-DTW). By and large, FVM often performs better than soft-DTW (temporal alignment only) by 3–5% on average.

To explain what makes JEANIE perform well on the task of comparing pairs of sequences, we perform some visualisations. To this end, we choose skeleton sequences from UWA3D Multiview Activity II for experiments and visualizations of FVM and JEANIE. UWA3D Multiview Activity II contains rich viewpoint configurations and so is perfect for our investigations. We verify that our JEANIE is able to find the better matching distances compared to FVM on two following scenarios.

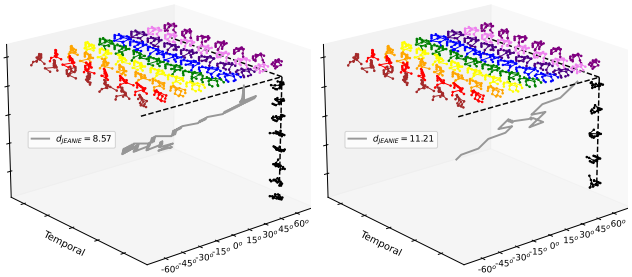
Matching similar actions. We choose a *walking* skeleton sequence (‘`a12_s01_e01_v01`’) as the query sample with more viewing angles for the camera viewpoint simulation, and we select another *walking* skeleton sequence of a different view (‘`a12_s01_e01_v03`’) and a *running* skeleton sequence (‘`a20_s01_e01_v02`’) as support samples respectively.

Matching actions with similar motion trajectories. We choose a *two hand punching* skeleton sequence (‘`a04_s01_e01_v01`’) as the query sample with more viewing angles for the camera viewpoint simulation, and we select another *two hand punching* skeleton sequence of a different view (‘`a04_s05_e01_v02`’) and a *holding head* skeleton sequence (‘`a10_s05_e01_v02`’) as support samples respectively.

Figures 10 and 11 show the visualizations. Comparing Figures 10a and 10b of FVM, we notice that for skeleton sequences from different action classes (*walking vs. running*), FVM finds the path with a very small distance $d_{\text{FVM}} = 2.68$. In contrast, for sequences from the same action class (*walking vs. walking*), FVM gives $d_{\text{FVM}} = 4.60$ which is higher

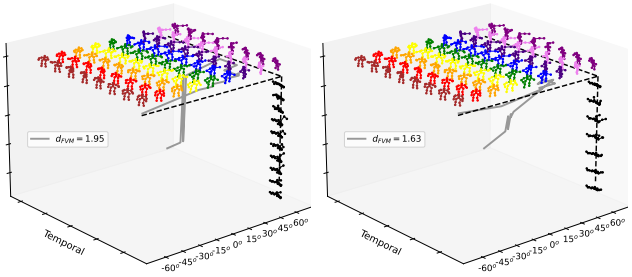


(a) *walking vs. walking* ($d_{\text{FVM}} = 4.60$) (b) *walking vs. running* ($d_{\text{FVM}} = 2.68$)

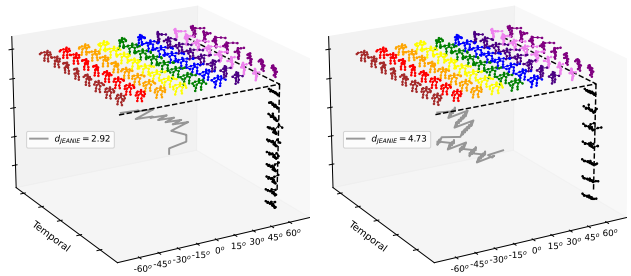


(c) *walking vs. walking* ($d_{\text{JEANIE}} = 8.57$) (d) *walking vs. running* ($d_{\text{JEANIE}} = 11.21$)

Fig. 10: Visualization of FVM and JEANIE for *walking vs. walking* (two different sequences) and *walking vs. running*. We notice that for two different action sequences in (b), the greedy FVM finds the path with a very small distance $d_{\text{FVM}} = 2.68$ but for sequences of the same action class, FVM gives $d_{\text{FVM}} = 4.60$. This is clearly suboptimal as the within-class distance is higher then the between-class distance (to counteract this issue, we propose JEANIE). In contrast, our JEANIE is able to produce a smaller distance for within-class sequences and a larger distance for between-class sequences, which is a very important property when comparing pairs of sequences.



(a) *two hand punching vs. two hand punching* ($d_{\text{FVM}} = 1.95$) (b) *two hand punching vs. holding head* ($d_{\text{FVM}} = 1.63$)



(c) *two hand punching vs. two hand punching* ($d_{\text{JEANIE}} = 2.92$) (d) *two hand punching vs. holding head* ($d_{\text{JEANIE}} = 4.73$)

Fig. 11: Visualization of FVM and JEANIE for *two hand punching vs. two hand punching* (two different sequences) and *two hand punching vs. holding head*. We notice that for two different action sequences in (b), the greedy FVM finds the path which results in $d_{\text{FVM}} = 1.63$ for sequences of different action classes, yet FVM gives $d_{\text{FVM}} = 1.95$ for two sequences of the same class. The within-class distance should be smaller than the between-class distance but greedy approaches such as FVM cannot handle this requirement well. JEANIE gives smaller distance when comparing within-class sequences compared to between-class sequences. This is very important for comparing sequences.

than in case of within-class sequences. This is an undesired effect which may result in wrong comparison decision. In contrast, in Figures 10c and 10d, our JEANIE gives $d_{\text{JEANIE}} = 8.57$ for sequences of the same action class and $d_{\text{JEANIE}} = 11.21$ for sequences from different action classes, which means that the within-class distances are smaller than between-class distances. This is a very important property when comparing pairs of sequences.

Figure 11 provides similar observations that JEANIE produces more reasonable matching distances than FVM.

4.7 Discussion on Multiview Action Recognition

As mentioned in Section 4.5, JEANIE yields good results especially in unsupervised learning, with the performance gain over 5% on UWA3D Multiview Activity II and 4% on NTU-120 multiview classification protocols. Below we discuss the multiview supervised FSAR.

Table 8 (*Sup.*) shows that adding temporal alignment (with soft-DTW) to SGC, APPNP and S^{2}GC improves results on UWA3D Multiview Activity II, and the big performance gain is obtained via further adding the viewpoint alignment by JEANIE. Despite the dataset is challenging due to novel viewpoints, JEANIE performs consistently well on all different combinations of training/testing viewpoint settings. This is expected as our method aligns both temporal and camera viewpoint which allows a robust classification. JEANIE outperforms FVM by 4.2% and the baseline (*temporal alignment only with soft-DTW*) by 7% on average.

Influence of camera views has been explored in [90] on UWA3D Multiview Activity II, and they show that when the left view V_2 and right view V_3 were used for training and front view V_1 for testing, the recognition accuracy is high since the viewing angle of the front view V_1 is between V_2 and V_3; when the left view V_2 and top view V_4 are used for training and right view V_3 is used for testing (or the

front view V_1 and right view V_3 are used for training and top view V_4 is used for testing), the recognition accuracies are slightly lower. However, as shown in Table 8 (*Sup.*), our JEANIE is able to handle the influence of viewpoints and performs almost equally well on all 12 different view combinations which highlights the importance of jointly aligning both temporal and viewpoint modes of sequences.

Table 9 (*Sup.*) shows the experimental results on the NTU-120. We notice that adding more camera viewpoints to the training process helps the multiview classification, *e.g.*, using bottom and center views for training and top view for testing, and using left and center views for training and the right view for testing, and the performance gain is more than 4% on (*100/same 100*). Notice that even though we test on 20 novel classes (*100/novel 20*) which are never used in the training set, we still achieve 62.7% and 70.8% for multiview classification in horizontal/vertical camera viewpoints.

4.8 Fusion of Supervised and Unsupervised FSAR

Recall that Section 3.5 defines two baseline and two advanced fusion strategies for supervised and unsupervised learning due to their complementary nature.

Tables 5 (*Fusion*), 6 (*Fusion*), 7 (*Fusion*) and 8 (*Fusion*) show that fusion improves the performance. The MAML-inspired fusion yields 5%, 5.1%, 4.2% and 9% improvements compared to the supervised FSAR only on NTU-60, NTU-120, Kinetics-skeleton and UWA3D Multiview Activity II, respectively. This validates our assertion that JEANIE helps design robust feature space for comparing sequences both in supervised and unsupervised scenarios.

The adaptation-based fusion (*Adaptation-based*) performs almost as well as the MAML-inspired fusion, within 1% difference across datasets. This is expected as MAML algorithms are designed to learn across multiple tasks (in our case unsupervised reconstruction-driven loss and the supervised loss) and domain adaptation inspired feature alignment has similar effect.

Training one EN with the fusion of both supervised and unsupervised FSAR outperforms a naive fusion of scores (*Weighted fusion*) from two Encoding Networks trained separately. Finetuning an unsupervised model with supervised loss (*Finetuning unsup.*) outperforms the weighted fusion.

Table 10 compares different testing strategies on fusion models. The MAML-inspired fusion achieves the best results, with 1.5%, 22.0% and 3.7% improvements when tested on supervised learning, unsupervised learning and a fusion of both. For both adaptation-based and MAML-inspired fusions, testing on unsupervised FSAR only (nearest neighbor on dictionary-encoded vectors) performs close to the results obtained from supervised FSAR only (nearest neighbor on feature maps), *i.e.*, within 5% difference. The reduced performance gap between supervised and unsupervised FSAR suggests that the feature space of EN is adapted to both unsupervised and supervised FSAR.

5 Conclusions

We have proposed tEmporal and cAmera viewpoiNt alIgnmEnt for sequence pairs (JEANIE) and evaluated it on 3D skeleton sequences whose pose/camera views are easy to manipulate in 3D. We have shown that the smooth property of alignment jointly in temporal and viewpoint modes is advantageous compared to the temporal alignment alone (soft-DTW) or models that freely align viewpoint per each temporal block without imposing the smoothness on variations of the matching path.

JEANIE can match correctly support and query sequence pairs as it factors out nuisance variations, which is essential under limited samples of novel classes. Especially, unsupervised FSAR benefits in such a scenario, *i.e.*, when nuisance variations are factored out, sequences of the same class are more likely to occupy similar/same set of atoms in the dictionary. As supervised FSAR forms the feature space driven by the similarity learning loss and the unsupervised FSAR by the dictionary reconstruction-driven loss, fusing both learning strategies has helped achieve further gains.

Our experiments have shown that using the stereo camera geometry is more efficient than simply generating multiple views by Euler angles. Finally, we have contributed unsupervised, supervised and fused FSAR approaches to the small family of FSAR for articulated 3D body joints.

Appendices

A Euler Rotations and Simulated Camera Views

Euler angles [1] are defined as successive planar rotation angles around x, y, and z axes. For 3D coordinates, we have the following rotation matrices $\mathbf{R}_x$, $\mathbf{R}_y$ and $\mathbf{R}_z$:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & \sin\theta_x \\ 0 & -\sin\theta_x & \cos\theta_x \end{bmatrix}, \begin{bmatrix} \cos\theta_y & 0 & -\sin\theta_y \\ 0 & 1 & 0 \\ \sin\theta_y & 0 & \cos\theta_y \end{bmatrix}, \begin{bmatrix} \cos\theta_z & \sin\theta_z & 0 \\ -\sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{18}$$

As the resulting composite rotation matrix depends on the order of rotation axes, *i.e.*, $\mathbf{R}_x\mathbf{R}_y\mathbf{R}_z \neq \mathbf{R}_z\mathbf{R}_y\mathbf{R}_x$, we also investigate the algebra of stereo projection.

Stereo projections [2]. Suppose we have a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t} = [t_x, t_y, t_z]^T$ between left/right cameras (imagine some non-existent stereo camera). Let $\mathbf{M}_l$ and $\mathbf{M}_r$ be the intrinsic matrices of the left/right cameras. Let $\mathbf{p}_l$ and $\mathbf{p}_r$ be coordinates of the left/right camera. As the origin of the right camera in the left camera coordinates is $\mathbf{t}$, we have: $\mathbf{p}_r = \mathbf{R}(\mathbf{p}_l - \mathbf{t})$ and $(\mathbf{p}_l - \mathbf{t})^T = (\mathbf{R}^T\mathbf{p}_r)^T$. The plane (polar surface) formed by all points passing through $\mathbf{t}$ can be expressed by $(\mathbf{p}_l - \mathbf{t})^T(\mathbf{p}_l \times \mathbf{t}) = 0$. Then, $\mathbf{p}_l \times \mathbf{t} = \mathbf{S}\mathbf{p}_l$ where $\mathbf{S} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$. Based on the above equations, we obtain $\mathbf{p}_r{}^T\mathbf{R}\mathbf{S}\mathbf{p}_l =$

0, and note that $\mathbf{RS}=\mathbf{E}$ is the Essential Matrix, and $\mathbf{p}_r^T\mathbf{E}\mathbf{p}_l=0$ describes the relationship for the same physical point under the left and right camera coordinate system. As $\mathbf{E}$ has no internal inf. about the camera, and $\mathbf{E}$ is based on the camera coordinates, we use a fundamental matrix $\mathbf{F}$ that describes the relationship for the same physical point under the camera pixel coordinate system. The relationship between the pixel and camera coordinates is: $\mathbf{p}^*=\mathbf{M}\mathbf{p}'$ and $\mathbf{p}_r'^T\mathbf{E}\mathbf{p}_l'=0$.

Now, suppose the pixel coordinates of $\mathbf{p}_l'$ and $\mathbf{p}_r'$ in the pixel coordinate system are $\mathbf{p}_l^*$ and $\mathbf{p}_r^*$, then we can write $\mathbf{p}_r^{*T}(\mathbf{M}_r^{-1})^T\mathbf{E}\mathbf{M}_l^{-1}\mathbf{p}_l^*=0$, where $\mathbf{F}=(\mathbf{M}_r^{-1})^T\mathbf{E}\mathbf{M}_l^{-1}$ is the fundamental matrix. Thus, the relationship for the same point in the pixel coordinate system of the left/right camera is:

$$\mathbf{p}_r^{*T}\mathbf{F}\mathbf{p}_l^*=0. \tag{19}$$

We treat 3D body joint coordinates as $\mathbf{p}_l^*$. Given $\mathbf{F}$, we obtain their coordinates $\mathbf{p}_r^*$ in the new view.

B Graph Neural Network as a Block of Encoding Network

GNN notations. Firstly, let $G=(\mathbf{V},\mathbf{E})$ be a graph with the vertex set $\mathbf{V}$ with nodes $\{v_1,...,v_n\}$, and $\mathbf{E}$ are edges of the graph. Let $\mathbf{A}$ and $\mathbf{D}$ be the adjacency and diagonal degree matrix, respectively. Let $\tilde{\mathbf{A}}=\mathbf{A}+\mathbf{I}$ be the adjacency matrix with self-loops (identity matrix) with the corresponding diagonal degree matrix $\tilde{\mathbf{D}}$ such that $\tilde{D}_{ii}=\sum_j(\mathbf{A}^{ij}+\mathbf{I}^{ij})$. Let $\mathbf{S}=\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$ be the normalized adjacency matrix with added self-loops. For the l-th layer, we use $\mathbf{\Theta}^{(l)}$ to denote the learnt weight matrix, and $\mathbf{\Phi}$ to denote the outputs from the graph networks. Below, we list backbones used by us.

GCN [39]. GCNs learn the feature representations for the features $\mathbf{x}_i$ of each node over multiple layers. For the l-th layer, we denote the input by $\mathbf{H}^{(l-1)}$ and the output by $\mathbf{H}^{(l)}$. Let the input (initial) node representations be $\mathbf{H}^{(0)}=\mathbf{X}$. By $\mathbf{X}$ we mean some node features for generality of explanation. For our particular case, we would be setting $\mathbf{H}^{(0)}=\widehat{\mathbf{X}}$ for each temporal block. For an L-layer GCN, the output representations are given by:

$$\mathbf{\Phi}_{\text{GCN}}=\mathbf{S}\mathbf{H}^{(L-1)}\mathbf{\Theta}^{(L)} \text{ where } \mathbf{H}^{(l)}=\text{ReLU}(\mathbf{S}\mathbf{H}^{(l-1)}\mathbf{\Theta}^{(l)}). \tag{20}$$

APPNP [40]. Personalized Propagation of Neural Predictions (PPNP) and its fast approximation, APPNP, are based on the personalized PageRank. Let $\mathbf{H}^{(0)}=f_{\mathbf{\Theta}}(\mathbf{X})$ be the input to APPNP, where $f_{\mathbf{\Theta}}(\cdot)$ can be an MLP with parameters $\mathbf{\Theta}$. Let the output of the l-th layer be $\mathbf{H}^{(l)}=(1-\alpha)\mathbf{S}\mathbf{H}^{(l-1)}+\alpha\mathbf{H}^{(0)}$, where α is the teleport (or restart) probability in range $(0,1]$. For an L-layer APPNP, we have:

$$\mathbf{\Phi}_{\text{APPNP}}=(1-\alpha)\mathbf{S}\mathbf{H}^{L}+\alpha\mathbf{H}^{(\mathbf{0})}. \tag{21}$$

SGC [106] & **S^{2}GC** [129]. SGC captures the L-hops neighborhood in the graph by the L-th power of the transition matrix used as a spectral filter. For an L-layer SGC, we obtain:

$$\mathbf{\Phi}_{\text{SGC}}=\mathbf{S}^L\mathbf{X}\mathbf{\Theta}. \tag{22}$$

Based on a modified Markov Diffusion Kernel, Simple Spectral Graph Convolution (S^{2}GC) is the summation over l-hops, $l=1,...,L$. The output of S^{2}GC is:

$$\mathbf{\Phi}_{\text{S}^2\text{GC}}=\frac{1}{L}\sum_{l=1}^{L}((1-\alpha)\mathbf{S}^l\mathbf{X}+\alpha\mathbf{X})\mathbf{\Theta}. \tag{23}$$

In case of APPNP, SGC and S^{2}GC, $|\mathcal{F}_{GNN}|=0$ because we do not use their learnable parameters $\mathbf{\Theta}$ (*i.e.*, think $\mathbf{\Theta}$ is set as the identity matrix). The GNN outputs are further passes into a Transformer and an FC layer, which returns $\mathbf{\Psi}\in\mathbb{R}^{d'\times K\times K'\times\tau}$ query feature maps and $\mathbf{\Psi}'\in\mathbb{R}^{d'\times\tau'}$ support feature maps.

C Feature Coding and Dictionary Learning

The core idea of feature coding is to reconstruct a feature vector with codewords by solving a least squares based optimization problem with constraints imposed on the codewords. The full codewords (a.k.a. elements or atoms) compose a dictionary. Atoms in the dictionary are not required to be orthogonal and the dictionary may be an over-complete (the number of atoms is larger than their dimension). For most feature coding algorithms, only a subset of codewords are chosen by the solver to represent a feature vector, and thus the coding vector $\boldsymbol{\alpha}$ may be sparse, *i.e.*, the responses are zeros on those codewords which are not chosen. In what follows, we however replace the Euclidean distance with the JEANIE measure.

The main difference among various feature coding methods lies in the constraint term. Alternatively, we obtain $\boldsymbol{\alpha}$ by defining some specific function $\boldsymbol{\alpha}(\boldsymbol{\Psi}_i;\boldsymbol{M})$ that implicitly realizes the regularization term. The choice of $\boldsymbol{\Omega}(\boldsymbol{\alpha}_i,\boldsymbol{M},\boldsymbol{\Psi}_i)$ realizes some desired constraints via regularization $\kappa>0$, *e.g.*, $\boldsymbol{\Omega}(\boldsymbol{\alpha}_i,\boldsymbol{M},\boldsymbol{\Psi}_i)=\|\boldsymbol{\alpha}_i\|_1$ encourages sparsity of $\boldsymbol{\alpha}$.

C.1 Feature Coding

Below we detail different feature coders we explore in our work, *i.e.*, Hard Assignment (HA) [14], Sparse Coding (SC) [47, 111], Non-negative Sparse Coding (SC$_+$) [35], Locality-constrained Linear Coding (LLC) [86], Soft Assignment (SA) [6, 27] and Locality-constrained Soft Assignment (LcSA) [42, 54]. LcSA is our default feature coder due to its simplicity and strong performance.

Hard Assignment (HA). This encoder assigns each $\boldsymbol{\Psi}$ to its nearest $\mathbf{m}$ by solving the following optimisation problem:

$$\begin{aligned}\boldsymbol{\alpha}(\boldsymbol{\Psi})=&\underset{\boldsymbol{\alpha}'\in\{0,1\}^k}{\arg\min}\ d^2_{\text{JEANIE}}(\boldsymbol{\Psi},\boldsymbol{M}\boldsymbol{\alpha}'),\\ &\text{s.t.}\quad\|\boldsymbol{\alpha}'\|_1=1.\end{aligned} \tag{24}$$

Sparse Coding (SC) & **Non-negative Sparse Coding (SC$_+$)**. SC encodes each $\boldsymbol{\Psi}$ as a sparse linear combination of atoms $\boldsymbol{M}$ by optimising the following objective:

$$\boldsymbol{\alpha}(\boldsymbol{\Psi})=\underset{\boldsymbol{\alpha}'}{\arg\min}\ d^2_{\text{JEANIE}}(\boldsymbol{\Psi},\boldsymbol{M}\boldsymbol{\alpha}')+\kappa\|\boldsymbol{\alpha}'\|_1, \tag{25}$$

whereas SC$_+$ additionally imposes a constraint that $\boldsymbol{\alpha}'\geq 0$. Both SC and SC$_+$ encode each $\boldsymbol{\Psi}$ on a subspace of $\boldsymbol{M}$ of size controlled by the sparsity term.

Locality-constrained Linear Coding (LLC). The LLC encoder uses the following criteria for each $\boldsymbol{\Psi}$:

$$\begin{aligned}\boldsymbol{\alpha}(\boldsymbol{\Psi})=&\underset{\boldsymbol{\alpha}'}{\arg\min}\ d^2_{\text{JEANIE}}(\boldsymbol{\Psi},\boldsymbol{M}\boldsymbol{\alpha}')+\kappa\|\mathbf{d}\odot\boldsymbol{\alpha}\|_2^2,\\ &\text{s.t.}\quad\mathbf{1}^{\text{T}}\boldsymbol{\alpha}'=1,\end{aligned} \tag{26}$$

where $\odot$ denotes the element-wise multiplication and $\mathbf{d}\in\mathbb{R}^k$ is the non-locality penalty that penalises selection of dictionary atoms that are far from $\boldsymbol{\Psi}$. Specifically,

$$\mathbf{d}=\left[\exp\frac{d^2_{\text{JEANIE}}(\boldsymbol{\Psi},\mathbf{m}_1)}{\sigma},...,\exp\frac{d^2_{\text{JEANIE}}(\boldsymbol{\Psi},\mathbf{m}_k)}{\sigma}\right]^T, \tag{27}$$

where $\sigma\geq 0$ adjusts the weight decay speed for the non-locality penalty. We further normalize $\mathbf{d}$ to be between 0 and 1. The constraint $\mathbf{1}^{\text{T}}\boldsymbol{\alpha}'=1$ follows the shift-invariant requirements of the LLC encoder.

Soft Assignment (SA) & **Locality-constrained Soft Assignment (LcSA)**. SA expresses each $\boldsymbol{\Psi}$ as the membership probability of $\boldsymbol{\Psi}$ belonging

to each $\mathbf{m}$ in $\boldsymbol{M}$, a concept known from the MLE of Gaussian Mixture Models (GMM). SA is derived under equal mixing probability and shared variance σ of GMM components. SA is a closed-form term:

$$\boldsymbol{\alpha}'(\boldsymbol{\Psi};\boldsymbol{M},\sigma)=\frac{1}{Z(\boldsymbol{\Psi};\boldsymbol{M},\sigma)}\left[\mathrm{e}^{-\frac{d^2_{\text{JEANIE}}(\boldsymbol{\Psi},\mathbf{m}_1)}{2\sigma^2}},...,\mathrm{e}^{-\frac{d^2_{\text{JEANIE}}(\boldsymbol{\Psi},\mathbf{m}_k\|)}{2\sigma^2}}\right]^{\mathrm{T}},$$
$$\text{where}\quad Z(\boldsymbol{\Psi};\boldsymbol{M},\sigma)=\sum_{k'=1,...,k}\mathrm{e}^{-\frac{1}{2\sigma^2}d^2_{\text{JEANIE}}(\boldsymbol{\Psi},\mathbf{m}_{k'})}. \tag{28}$$

The above model usually yields largest values of α'_i for anchor $\mathbf{m}_i$ in $\boldsymbol{M}$ that is a close JEANIE neighbor of $\boldsymbol{\Psi}$. However, even for $\mathbf{m}_i$ that is far from $\boldsymbol{\Psi}$, $\alpha'_i>0$. For this reason, SA is only approximately locality-constrained.

LcSA admits the locality-constrained membership probability of the form:

$$\boldsymbol{\alpha}(\boldsymbol{\Psi})=\pi(\boldsymbol{\alpha}'(\boldsymbol{\Psi};\boldsymbol{M}_{\text{NN}(\boldsymbol{\Psi};k')})), \tag{29}$$

where $\boldsymbol{M}_{\text{NN}(\boldsymbol{\Psi};k')}$ returns k' nearest neighbors of $\boldsymbol{\Psi}$ in $\boldsymbol{M}$ based on the JEANIE measure, whereas $\pi(\cdot)$ projects back coefficients of $\boldsymbol{\alpha}'$ into $\boldsymbol{\alpha}$ at positions following original indexes of nearest neighbors in dictionary $\boldsymbol{M}$. Remaining locations in $\boldsymbol{\alpha}$ are zeroed. LcSA forms subspaces of size k'.

C.2 Dictionary Learning

For all the above listed feature coding methods, we employ a simple dictionary learning objective which follows Eq. (16). We assume some evaluated/fixed dictionary-coded vectors as a coding matrix $\boldsymbol{A}\equiv[\boldsymbol{\alpha}_1,...,\boldsymbol{\alpha}_{N'}]$ (N' is the number of samples per mini-batch), and we compute:

$$\boldsymbol{M}=\arg\min_{\boldsymbol{M}'}\sum_{i=1}^{N'}d^2_{\text{JEANIE}}(\boldsymbol{\Psi}_i,\boldsymbol{M}'\boldsymbol{\alpha}_i). \tag{30}$$

Notice that for fixed $\boldsymbol{A}$ and fixed feature matrices $\boldsymbol{\Psi}$, the regularization term becomes a constant. For the dictionary learning step, we detach $\boldsymbol{\Psi}$ and $\boldsymbol{\alpha}$, and run 1–5 iterations of gradient descent per mini-batch *w.r.t.* $\boldsymbol{M}$.

D Fusion by Alignment

Fusion by alignment of supervised and unsupervised feature maps. Inspired by domain adaptaion, Algorithm 4 performs a fusion of supervised and unsupervised FSAR by alignment of feature maps obtained with supervised and unsupervised FSAR. Specifically, we start by generating representations with several viewpoints. For each mini-batch of size B we form a set with N' feature maps which are passed to Algorithm 2. Subsequently, from EN parameters $\mathcal{F}$ we obtain parameters $\widehat{\mathcal{F}}$ that help accommodate unsupervised reconstruction-driven learning. We compute "unsupervised" feature maps for such parameters and encourage "supervised" feature maps to align with them based on the JEANIE measure. Parameter $\lambda\geq 0$ controls the strength of alignment. For the supervised step, we use the supervised loss from Eq. (14) and (15). Finally, we update EN parameters $\mathcal{F}$.

E Additional results on Unsupervised FSAR

Tables 11 and 12 below show additional results on the NTU-120, the 2D and 3D Kinetics-skeleton datasets.

Algorithm 4 Fusion of Supervised and Unsupervised FSAR by Feature Maps Alignment (one training iteration).

Input: $\Gamma\equiv\{\boldsymbol{\mathcal{X}}_b\}_{b\in\mathcal{I}_B}\cup\{\boldsymbol{\mathcal{X}}'_{b,n,z}\}_{\substack{b\in\mathcal{I}_B\\n\in\mathcal{I}_N\\z\in\mathcal{I}_Z}}$: query/support blocks in batch; $\mathcal{F}$: EN parameters; $\boldsymbol{M}$ and $\boldsymbol{A}$; `alpha_iter` and `dic_iter`: numbers of iterations for updating $\boldsymbol{A}$ and $\boldsymbol{M}$; $\omega,\omega_{\text{DL}}$ and ω_{EN}: the learning rate for $\boldsymbol{A}$, $\boldsymbol{M}$ and $\mathcal{F}$ respectively; B: size of the mini-batch; λ: regularization parameter.

1: $\Upsilon\equiv\{\boldsymbol{\Psi}_b\}_{b\in\mathcal{I}_B}\cup\{\boldsymbol{\Psi}'_{b,n,z}\}_{\substack{b\in\mathcal{I}_B\\n\in\mathcal{I}_N\\z\in\mathcal{I}_Z}}$ where $\begin{cases}\boldsymbol{\Psi}_b=f^*(\boldsymbol{\mathcal{X}}_b;\mathcal{F})\\ \boldsymbol{\Psi}'_{b,n,z}=f^*(\boldsymbol{\mathcal{X}}'_{b,n,z};\mathcal{F})\end{cases}$
(obtain feature maps for global parameters $\mathcal{F}$)
2: $\widehat{\mathcal{F}}:=\mathcal{F}$ (copy parameters of EN)
3: $(\widehat{\mathcal{F}},\boldsymbol{M})=$ Algorithm2$(\Upsilon,\widehat{\mathcal{F}},\boldsymbol{M},\boldsymbol{A},$ `alpha_iter`, `dic_iter`$,\omega,\omega_{\text{DL}},\omega_{\text{EN}})$ (unsupervised FSAR)
4: $\widehat{\Upsilon}\equiv\{\widehat{\boldsymbol{\Psi}}_b\}_{b\in\mathcal{I}_B}\cup\{\widehat{\boldsymbol{\Psi}}'_{b,n,z}\}_{\substack{b\in\mathcal{I}_B\\n\in\mathcal{I}_N\\z\in\mathcal{I}_Z}}$ where $\begin{cases}\widehat{\boldsymbol{\Psi}}_b=f^*(\boldsymbol{\mathcal{X}}_b;\widehat{\mathcal{F}})\\ \widehat{\boldsymbol{\Psi}}'_{b,n,z}=f^*(\boldsymbol{\mathcal{X}}'_{b,n,z};\widehat{\mathcal{F}})\end{cases}$
(obtain feature maps for parameters $\widehat{\mathcal{F}}$ from the unsupervised step)
5: $\mathcal{L}_{\text{align}}=\sum_{i=1}^{N'}d^2_{\text{JEANIE}}(\boldsymbol{\Psi}_i,\widehat{\boldsymbol{\Psi}}_i)$ (alignment of sup. & unsup. maps) where $N'=|\Upsilon|,\boldsymbol{\Psi}\in\Upsilon,\widehat{\boldsymbol{\Psi}}\in\widehat{\Upsilon}$
6: $\boldsymbol{d}^+=[d_{\text{JEANIE}}(\boldsymbol{\Psi}_b,\boldsymbol{\Psi}'_{b,1,z})]_{\substack{b\in\mathcal{I}_B\\z\in\mathcal{I}_Z}}$ (within-class distance)
7: $\boldsymbol{d}^-=[d_{\text{JEANIE}}(\boldsymbol{\Psi}_b,\boldsymbol{\Psi}'_{b,n,z})]_{\substack{b\in\mathcal{I}_B\\n\in\mathcal{I}_N\setminus\{1\}\\z\in\mathcal{I}_Z}}$ (between-class distance)
8: $\mathcal{F}:=\mathcal{F}-\omega_{\text{EN}}\nabla_{\mathcal{F}}\big(l(\boldsymbol{d}^+,\boldsymbol{d}^-)+\lambda\mathcal{L}_{\text{align}}\big)$

Output: $\mathcal{F}$ and $\boldsymbol{M}$

Table 11: Experimental results on NTU-120 (S^{2}GC backbone). All methods enjoy temporal alignment by soft-DTW or JEANIE (joint temporal and viewpoint alignment). We use the ℓ_2 norm for comparing the codes in unsupervised setting with soft-DTW. For unsupervised JEANIE, the distance for comparing the codes is indicated.

		viewpoint simulation	align.	20	40	60	80	100
	soft-DTW (HA)	-	T	11.2	16.3	19.0	25.8	30.9
	soft-DTW (SC)	-	T	12.1	17.4	21.4	27.0	32.7
	soft-DTW (SC$_+$)	-	T	11.8	17.0	21.2	26.5	32.2
	soft-DTW (LLC)	-	T	14.0	18.7	23.1	29.3	34.1
	soft-DTW (SA)	-	T	15.0	20.1	24.3	30.5	38.3
	soft-DTW (LcSA)	-	T	15.7	21.4	25.2	32.0	40.2
Unsup.	JEANIE (LLC)–ℓ_1	CamVPC	T+2V	18.0	23.8	30.5	36.3	43.0
+Transf.	JEANIE (LLC)–ℓ_2	CamVPC	T+2V	18.3	24.2	30.8	36.0	43.3
	JEANIE (LLC)–HIK	CamVPC	T+2V	18.3	24.0	31.0	36.3	43.0
	JEANIE (LLC)–CSK	CamVPC	T+2V	17.8	24.0	30.8	36.3	43.0
	JEANIE (LcSA)–ℓ_1	CamVPC	T+2V	18.3	24.5	32.0	39.5	48.0
	JEANIE (LcSA)–ℓ_2	CamVPC	T+2V	**18.6**	25.0	32.2	**40.0**	**48.5**
	JEANIE (LcSA)–HIK	CamVPC	T+2V	18.3	24.8	**32.2**	39.6	48.0
	JEANIE (LcSA)–CSK	CamVPC	T+2V	**18.6**	**25.2**	32.0	39.6	**48.5**
	FVM (LcSA)–CSK	CamVPC	T+2V	17.5	22.4	30.7	36.1	44.5

Table 12: Experiments on 2D and 3D Kinetics-skeleton. We use the ℓ_2 norm for comparing the codes in unsupervised setting with soft-DTW. For unsupervised JEANIE, the distance for comparing the codes is indicated.

		viewpoint simulation	alignment	2D skel.	3D skel.
	soft-DTW(LLC)	-	T	18.7	21.3
	soft-DTW(SA)	-	T	18.7	21.8
	soft-DTW(LcSA)	-	T	19.3	22.2
Unsup.	JEANIE (LcSA)–ℓ_1	CamVPC	T+2V	-	28.0
+Transf.	JEANIE (LcSA)–ℓ_2	CamVPC	T+2V	-	**28.3**
	JEANIE (LcSA)–HIK	CamVPC	T+2V	-	**28.3**
	JEANIE (LcSA)–CSK	CamVPC	T+2V	-	**28.3**
	FVM (LcSA)–ℓ_2	CamVPC	T+2V	-	25.1

References

1. Euler angles. Wikipedia, https://en.wikipedia.org/wiki/Euler_angles. Accessed: 08-03-2022
2. Lecture 12: Camera projection. On-line, http://www.cse.psu.edu/~rtc12/CSE486/lecture12.pdf. Accessed: 08-03-2022
3. Ahn, D., Kim, S., Hong, H., Ko, B.C.: Star-transformer: A spatio-temporal cross attention transformer for human action recognition. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 3330–3339 (2023)
4. Bart, E., Ullman, S.: Cross-generalization: Learning novel classes from a single example by feature replacement. CVPR pp. 672–679 (2005)
5. Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., Cox, D.D.: Hyperopt: a python library for model selection and hyperparameter optimization. CSD **8**(1), 014008 (2015)
6. Bilmes, J.: A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. International Computer Science Institute **4**, 126 (1998)
7. Cao, K., Ji, J., Cao, Z., Chang, C.Y., Niebles, J.C.: Few-shot video classification via temporal alignment. In: CVPR (2020)
8. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: CVPR (2017)
9. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR (2017)
10. Catalin, Ionescu, Dragos, Papava, Vlad, Olaru, Cristian, Sminchisescu: Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE TPAMI (2014)
11. Chen, Y., Zhang, Z., Yuan, C., Li, B., Deng, Y., Hu, W.: Channel-wise topology refinement graph convolution for skeleton-based action recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 13359–13368 (2021)
12. Cheng, K., Zhang, Y., Cao, C., Shi, L., Cheng, J., Lu, H.: Decoupling gcn with dropgraph module for skeleton-based action recognition. In: A. Vedaldi, H. Bischof, T. Brox, J.M. Frahm (eds.) Computer Vision – ECCV 2020, pp. 536–553. Springer International Publishing, Cham (2020)
13. Cheng, K., Zhang, Y., He, X., Chen, W., Cheng, J., Lu, H.: Skeleton-based action recognition with shift graph convolutional network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
14. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: In Workshop on Statistical Learning in Computer Vision, ECCV, pp. 1–22 (2004)
15. Cuturi, M.: Fast global alignment kernels. In: ICML (2011)
16. Cuturi, M., Blondel, M.: Soft-dtw: a differentiable loss function for time-series. In: ICML (2017)
17. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2020)
18. Dvornik, N., Schmid, C., Mairal1, J.: Selecting relevant features from a multi-domain representation for few-shot classification. In: ECCV (2020)
19. Dwivedi, S.K., Gupta, V., Mitra, R., Ahmed, S., Jain, A.: Protogan: Towards few shot learning for action recognition. arXiv (2019)
20. Elsken, T., Staffler, B., Metzen, J.H., Hutter, F.: Meta-learning of neural architectures for few-shot learning. In: CVPR (2020)
21. Fei, N., Guan, J., Lu, Z., Gao, Y.: Few-shot zero-shot learning: Knowledge transfer with less supervision. In: ACCV (2020)
22. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE TPAMI **28**(4), 594–611 (2006)
23. Feichtenhofer, C., Pinz, A., Wildes, R.P.: Spatiotemporal multiplier networks for video action recognition. In: CVPR (2017)
24. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: CVPR (2016)
25. Fink, M.: Object classification from a single example utilizing class relevance metrics. NeurIPS pp. 449–456 (2005)
26. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: D. Precup, Y.W. Teh (eds.) Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, *Proceedings of Machine Learning Research*, vol. 70, pp. 1126–1135. PMLR (2017)
27. Gemert, J.C., Geusebroek, J.M., Veenman, C.J., Smeulders, A.W.: Kernel codebooks for scene categorization. In: Proceedings of the 10th European Conference on Computer Vision: Part III, ECCV '08, p. 696–709. Springer-Verlag, Berlin, Heidelberg (2008). DOI 10.1007/978-3-540-88690-7_52
28. Girdhar, R., João Carreira, J., Doersch, C., Zisserman, A.: Video action transformer network. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 244–253 (2019). DOI 10.1109/CVPR.2019.00033
29. Guan, J., Zhang, M., Lu, Z.: Large-scale cross-domain few-shot learning. In: ACCV (2020)
30. Guo, H., Wang, H., Ji, Q.: Uncertainty-guided probabilistic transformer for complex action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 20052–20061 (2022)
31. Guo, M., Chou, E., Huang, D.A., Song, S., Yeung, S., Fei-Fei, L.: Neural graph matching networks for fewshot 3d action recognition. In: ECCV, pp. 653–669 (2018)
32. Guo, Y., Codella, N.C., Karlinsky, L., Codella, J.V., Smith, J.R., Saenko, K., Rosing, T., Feris, R.: A broader study of cross-domain few-shot learning. In: ECCV (2020)
33. Haasdonk, B., Burkhardt, H.: Invariant kernel functions for pattern analysis and machine learning. Mach. Learn. **68**(1), 35–61 (2007)
34. Hao, X., Li, J., Guo, Y., Jiang, T., Yu, M.: Hypergraph neural network for skeleton-based action recognition. IEEE Transactions on Image Processing **30**, 2263–2275 (2021). DOI 10.1109/TIP.2021.3051495
35. Hoyer, P.: Non-negative sparse coding. In: Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing, pp. 557–565 (2002). DOI 10.1109/NNSP.2002.1030067
36. Huang, Y., Yang, L., Sato, Y.: Compound prototype matching for few-shot action recognition. In: European Conference on Computer Vision, pp. 351–368. Springer (2022)
37. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A.: The kinetics human action video dataset. arXiv (2017)
38. Kim, J., Oh, S., Hong, S.: Transformers generalize deepsets and can be extended to graphs & hypergraphs. In: A. Beygelzimer, Y. Dauphin, P. Liang, J.W. Vaughan (eds.) Advances in Neural Information Processing Systems (2021)
39. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
40. Klicpera, J., Bojchevski, A., Gunnemann, S.: Predict then propagate: Graph neural networks meet personalized pagerank. In: ICLR (2019)
41. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: ICML deep learning workshop, vol. 2 (2015)

42. Koniusz, P., Mikolajczyk, K.: Soft assignment of visual words as linear coordinate coding and optimisation of its reconstruction error. In: 2011 18th IEEE International Conference on Image Processing, pp. 2413–2416 (2011). DOI 10.1109/ICIP.2011.6116129
43. Koniusz, P., Wang, L., Cherian, A.: Tensor representations for action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
44. Koniusz, P., Zhang, H.: Power normalizations in fine-grained image, few-shot image and graph classification. IEEE Trans. Pattern Anal. Mach. Intell. **44**(2), 591–609 (2022)
45. Korban, M., Li, X.: Ddgcn: A dynamic directed graph convolutional network for action recognition. In: A. Vedaldi, H. Bischof, T. Brox, J.M. Frahm (eds.) Computer Vision – ECCV 2020, pp. 761–776. Springer International Publishing, Cham (2020)
46. Lake, B.M., Salakhutdinov, R., Gross, J., Tenenbaum, J.B.: One shot learning of simple visual concepts. CogSci (2011)
47. Lee, H., Battle, A., Raina, R., Ng, A.Y.: Efficient sparse coding algorithms. NIPS'06, p. 801–808. MIT Press, Cambridge, MA, USA (2006)
48. Li, F.F., VanRullen, R., Koch, C., Perona, P.: Rapid natural scene categorization in the near absence of attention. PNAS **99**(14), 9596–9601 (2002)
49. Li, K., Zhang, Y., Li, K., Fu, Y.: Adversarial feature hallucination networks for few-shot learning. In: CVPR (2020)
50. Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., Tian, Q.: Actional-structural graph convolutional networks for skeleton-based action recognition. In: CVPR (2019)
51. Lichtenstein, M., Sattigeri, P., Feris, R., Giryes, R., Karlinsky, L.: Tafssl: Task-adaptive feature sub-space learning for few-shot classification. In: ECCV (2020)
52. Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.Y., Kot, A.C.: Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. IEEE TPAMI (2019)
53. Liu, J., Wang, G., Hu, P., Duan, L., Kot, A.C.: Global context-aware attention lstm networks for 3d action recognition. In: CVPR, pp. 3671–3680 (2017)
54. Liu, L., Wang, L., Liu, X.: In defense of soft-assignment coding. In: 2011 International Conference on Computer Vision, pp. 2486–2493 (2011). DOI 10.1109/ICCV.2011.6126534
55. Liu, S., Lv, P., Zhang, Y., Fu, J., Cheng, J., Li, W., Zhou, B., Xu, M.: Semi-dynamic hypergraph neural network for 3d pose estimation. In: C. Bessiere (ed.) Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, pp. 782–788. International Joint Conferences on Artificial Intelligence Organization (2020). DOI 10.24963/ijcai.2020/109. Main track
56. Liu, Z., Zhang, H., Chen, Z., Wang, Z., Ouyang, W.: Disentangling and unifying graph convolutions for skeleton-based action recognition. In: CVPR (2020)
57. Lu, C., Koniusz, P.: Few-shot keypoint detection with uncertainty learning for unseen species. CVPR (2022)
58. Luo, Q., Wang, L., Lv, J., Xiang, S., Pan, C.: Few-shot learning via feature hallucination with variational inference. In: WACV (2021)
59. Ma, N., Zhang, H., Li, X., Zhou, S., Zhang, Z., Wen, J., Li, H., Gu, J., Bu, J.: Learning spatial-preserved skeleton representations for few-shot action recognition. In: European Conference on Computer Vision, pp. 174–191. Springer (2022)
60. Martinez, J., Hossain, R., Romero, J., Little, J.J.: A simple yet effective baseline for 3d human pose estimation. In: ICCV, pp. 2659–2668 (2017)
61. Memmesheimer, R., Häring, S., Theisen, N., Paulus, D.: Skeleton-dml: Deep metric learning for skeleton-based one-shot action recognition. arXiv (2021)
62. Memmesheimer, R., Theisen, N., Paulus, D.: Signal level deep metric learning for multimodal one-shot action recognition. arXiv (2020)
63. Miller, E.G., Matsakis, N.E., Viola, P.A.: Learning from one example through shared densities on transforms. CVPR **1**, 464–471 (2000)
64. Mishra, A., Verma, V.K., Reddy, M.S.K., Arulkumar, S., Rai, P., Mittal, A.: A generative approach to zero-shot and few-shot action recognition. In: WACV, pp. 372–380 (2018)
65. Plizzari, C., Cannici, M., Matteucci, M.: Spatial temporal transformer network for skeleton-based action recognition. CoRR **abs/2012.06399** (2020)
66. Plizzari, C., Cannici, M., Matteucci, M.: Skeleton-based action recognition via spatial and temporal transformer networks. Computer Vision and Image Understanding **208-209**, 103219 (2021). DOI https://doi.org/10.1016/j.cviu.2021.103219
67. Qin, Z., Liu, Y., Ji, P., Kim, D., Wang, L., McKay, B., Anwar, S., Gedeon, T.: Fusing higher-order features in graph neural networks for skeleton-based action recognition. IEEE TNNLS (2022)
68. Rahmani, H., Mahmood, A., Huynh, D.Q., Mian, A.: Histogram of Oriented Principal Components for Cross-View Action Recognition. IEEE TPAMI pp. 2430–2443 (2016)
69. Shah, K., Shah, A., Lau, C.P., de Melo, C.M., Chellappa, R.: Multi-view action recognition using contrastive learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 3381–3391 (2023)
70. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: Ntu rgb+d: A large scale dataset for 3d human activity analysis. In: CVPR (2016)
71. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Adasgn: Adapting joint number and model size for efficient skeleton-based action recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 13413–13422 (2021)
72. Si, C., Chen, W., Wang, W., Wang, L., Tan, T.: An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In: CVPR (2019)
73. Simon, C., Koniusz, P., Nock, R., Harandi, M.: On modulating the gradient for meta-learning. In: ECCV (2020)
74. Smola, A.J., Kondor, R.: Kernels and regularization on graphs. COLT (2003)
75. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. In: I. Guyon, U. von Luxburg, S. Bengio, H.M. Wallach, R. Fergus, S.V.N. Vishwanathan, R. Garnett (eds.) NeurIPS, pp. 4077–4087 (2017)
76. Song, Y.F., Zhang, Z., Shan, C., Wang, L.: Constructing stronger and faster baselines for skeleton-based action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (2022). DOI 10.1109/TPAMI.2022.3157033
77. Su, B., Wen, J.R.: Temporal alignment prediction for supervised representation learning and few-shot sequence classification. In: ICLR (2022)
78. Sun, K., Koniusz, P., Wang, Z.: Fisher-Bures adversary graph convolutional networks. UAI **115**, 465–475 (2019)
79. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H.S., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: CVPR, pp. 1199–1208 (2018)
80. Tang, L., Wertheimer, D., Hariharan, B.: Revisiting pose-normalization for fine-grained few-shot recognition. In: CVPR (2020)
81. Thatipelli, A., Narayan, S., Khan, S., Anwer, R.M., Khan, F.S., Ghanem, B.: Spatio-temporal relation modeling for few-shot action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 19958–19967 (2022)
82. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional net-

works. In: ICCV (2015)
83. Truong, T.D., Bui, Q.H., Duong, C.N., Seo, H.S., Phung, S.L., Li, X., Luu, K.: Direcformer: A directed attention in transformer approach to robust action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 20030–20040 (2022)
84. Villani, C.: Optimal Transport, Old and New. Springer (2009)
85. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D.: Matching networks for one shot learning. In: D.D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, R. Garnett (eds.) NeurIPS, pp. 3630–3638 (2016)
86. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3360–3367 (2010). DOI 10.1109/CVPR.2010.5540018
87. Wang, L.: Analysis and evaluation of Kinect-based action recognition algorithms. Master's thesis, School of the Computer Science and Software Engineering, The University of Western Australia (2017)
88. Wang, L.: Robust human action modelling. Ph.D. thesis, The Australian National University (2023)
89. Wang, L., Ding, Z., Tao, Z., Liu, Y., Fu, Y.: Generative multi-view human action recognition. In: ICCV (2019)
90. Wang, L., Huynh, D.Q., Koniusz, P.: A comparative review of recent kinect-based action recognition algorithms. IEEE TIP **29**, 15–28 (2020)
91. Wang, L., Huynh, D.Q., Mansour, M.R.: Loss switching fusion with similarity search for video classification. ICIP (2019)
92. Wang, L., Koniusz, P.: Self-supervising action recognition by statistical moment and subspace descriptors. In: ACM-MM, p. 4324–4333 (2021)
93. Wang, L., Koniusz, P.: Temporal-viewpoint transportation plan for skeletal few-shot action recognition. In: Proceedings of the Asian Conference on Computer Vision (ACCV), pp. 4176–4193 (2022)
94. Wang, L., Koniusz, P.: Uncertainty-DTW for time series and sequences. ECCV (2022)
95. Wang, L., Koniusz, P.: 3Mformer: Multi-order multi-mode transformer for skeletal action recognition. In: IEEE/CVF International Conference on Computer Vision and Pattern Recognition (2023)
96. Wang, L., Koniusz, P.: Flow dynamics correction for action recognition. ICASSP (2024)
97. Wang, L., Koniusz, P., Huynh, D.Q.: Hallucinating IDT descriptors and I3D optical flow features for action recognition with cnns. In: ICCV (2019)
98. Wang, L., Liu, J., Koniusz, P.: 3D skeleton-based few-shot action recognition with JEANIE is not so naïve. arXiv preprint arXiv: 2112.12668 (2021)
99. Wang, L., Sun, K., Koniusz, P.: High-order tensor pooling with attention for action recognition. ICASSP (2024)
100. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks for action recognition in videos. IEEE TPAMI **41**(11), 2740–2755 (2019)
101. Wang, S., Yue, J., Liu, J., Tian, Q., Wang, M.: Large-scale few-shot learning via multi-modal knowledge discovery. In: ECCV (2020)
102. Wang, X., Xu, X., Mu, Y.: Neural koopman pooling: Control-inspired temporal dynamics encoding for skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10597–10607 (2023)
103. Wang, X., Zhang, S., Qing, Z., Gao, C., Zhang, Y., Zhao, D., Sang, N.: Molo: Motion-augmented long-short contrastive learning for few-shot action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 18011–18021 (2023)
104. Wang, Y., Long, M., Wang, J., Yu, P.S.: Spatiotemporal pyramid network for video action recognition. In: CVPR (2017)
105. Wanyan, Y., Yang, X., Chen, C., Xu, C.: Active exploration of multimodal complementarity for few-shot action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6492–6502 (2023)
106. Wu, F., Zhang, T., de Souza Jr., A.H., Fifty, C., Yu, T., Weinberger, K.Q.: Simplifying graph convolutional networks. In: ICML (2019)
107. Xing, Z., Dai, Q., Hu, H., Chen, J., Wu, Z., Jiang, Y.G.: Svformer: Semi-supervised video transformer for action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 18816–18826 (2023)
108. Xu, B., Ye, H., Zheng, Y., Wang, H., Luwang, T., Jiang, Y.G.: Dense dilated network for few shot action recognition. In: ACM ICMR, pp. 379–387 (2018)
109. Yan, S., Xiong, Y., Lin, D.: Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In: AAAI (2018)
110. Yang, J., Dong, X., Liu, L., Zhang, C., Shen, J., Yu, D.: Recurring the transformer for video action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14063–14073 (2022)
111. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1794–1801 (2009). DOI 10.1109/CVPR.2009.5206757
112. Yu, X., Zhuang, Z., Koniusz, P., Li, H.: 6DoF object pose estimation via differentiable proxy voting regularizer. In: BMVC. BMVA Press (2020)
113. Zhang, H., Koniusz, P.: Power normalizing second-order similarity network for few-shot learning. In: WACV, pp. 1185–1193 (2019)
114. Zhang, H., Koniusz, P., Jian, S., Li, H., Torr, P.H.S.: Rethinking class relations: Absolute-relative supervised and unsupervised few-shot learning. In: CVPR, pp. 9432–9441 (2021)
115. Zhang, H., Li, H., Koniusz, P.: Multi-level second-order few-shot learning. IEEE Transactions on Multimedia (2022)
116. Zhang, H., Zhang, L., Qi, X., Li, H., Torr, P., Koniusz, P.: Few-shot action recognition with permutation-invariant attention. In: ECCV (2020)
117. Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., Zheng, N.: View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In: ICCV (2017)
118. Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., Zheng, N.: View adaptive neural networks for high performance skeleton-based human action recognition. IEEE TPAMI **41**(8), 1963–1978 (2019)
119. Zhang, P., Lan, C., Zeng, W., Xing, J., Xue, J., Zheng, N.: Semantics-guided neural networks for efficient skeleton-based human action recognition. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
120. Zhang, Q., Wang, T., Zhang, M., Liu, K., Shi, P., Snoussi, H.: Spatial-temporal transformer for skeleton-based action recognition. In: 2021 China Automation Congress (CAC), pp. 7029–7034 (2021). DOI 10.1109/CAC53003.2021.9728206
121. Zhang, S., Luo, D., Wang, L., Koniusz, P.: Few-shot object detection by second-order pooling. In: ACCV, *Lecture Notes in Computer Science*, vol. 12625, pp. 369–387. Springer (2020)
122. Zhang, S., Murray, N., Wang, L., Koniusz, P.: Time-rEversed diffusioN tEnsor Transformer: A new TENET of Few-Shot Object Detection. In: ECCV (2022)
123. Zhang, S., Wang, L., Murray, N., Koniusz, P.: Kernelized few-shot object detection with efficient integral aggregation. In:

CVPR, pp. 19207–19216 (2022)
124. Zhang, X., Xu, C., Tao, D.: Context aware graph convolution for skeleton-based action recognition. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
125. Zhang, Y., Wu, B., Li, W., Duan, L., Gan, C.: STST: Spatial-Temporal Specialized Transformer for Skeleton-Based Action Recognition, p. 3229–3237. Association for Computing Machinery, New York, NY, USA (2021)
126. Zheng, S., Chen, S., Jin, Q.: Few-shot action recognition with hierarchical matching and contrastive learning. In: European Conference on Computer Vision, pp. 297–313. Springer (2022)
127. Zhou, H., Liu, Q., Wang, Y.: Learning discriminative representations for skeleton based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10608–10617 (2023)
128. Zhu, A., Ke, Q., Gong, M., Bailey, J.: Adaptive local-component-aware graph convolutional network for one-shot skeleton-based action recognition. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 6038–6047 (2023)
129. Zhu, H., Koniusz, P.: Simple spectral graph convolution. In: ICLR (2021)
130. Zhu, H., Koniusz, P.: EASE: Unsupervised discriminant subspace learning for transductive few-shot learning. CVPR (2022)
131. Zhu, H., Koniusz, P.: Transductive few-shot learning with prototype-based label propagation by iterative graph refinement. CVPR (2023)
132. Zhu, H., Sun, K., Koniusz, P.: Contrastive laplacian eigenmaps. In: NeurIPS, pp. 5682–5695 (2021)
133. Zhu, L., Yang, Y.: Compound memory networks for few-shot video classification. In: ECCV (2018)
134. Zhu, X., Huang, P.Y., Liang, J., de Melo, C.M., Hauptmann, A.G.: Stmt: A spatial-temporal mesh transformer for mocap-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1526–1536 (2023)