Neural Koopman Pooling: Control-Inspired Temporal Dynamics Encoding for Skeleton-Based Action Recognition

Xinghan Wang[1], Xin Xu[1], Yadong Mu[1,2] *
[1]Peking University, [2]Peng Cheng Laboratory
{xinghan_wang, xuxin2001, myd}@pku.edu.cn

Abstract

Skeleton-based human action recognition is becoming increasingly important in a variety of fields. Most existing works train a CNN or GCN based backbone to extract spatial-temporal features, and use temporal average/max pooling to aggregate the information. However, these pooling methods fail to capture high-order dynamics information. To address the problem, we propose a plug-and-play module called Koopman pooling, which is a parameterized high-order pooling technique based on Koopman theory. The Koopman operator linearizes a non-linear dynamics system, thus providing a way to represent the complex system through the dynamics matrix, which can be used for classification. We also propose an eigenvalue normalization method to encourage the learned dynamics to be non-decaying and stable. Besides, we also show that our Koopman pooling framework can be easily extended to one-shot action recognition when combined with Dynamic Mode Decomposition. The proposed method is evaluated on three benchmark datasets, namely NTU RGB+D 60, 120 and NW-UCLA. Our experiments clearly demonstrate that Koopman pooling significantly improves the performance under both full-dataset and one-shot settings.

1. Introduction

Skeleton-based human action recognition is a crucial task in many applications, ranging from video surveillance to autonomous driving and human-robot interaction. With the prevalence of deep learning, the rising of LSTM, CNN and GCN has significantly improved the performance of action recognition. Most existing methods [9,13,28,46,70,73] use a CNN or GCN based backbone to extract complex spatial-temporal features, and use temporal average/max pooling to aggregate the information. However, vanilla temporal average/max pooling contains only first-order information and abandons higher-order statistical information.

*Corresponding author

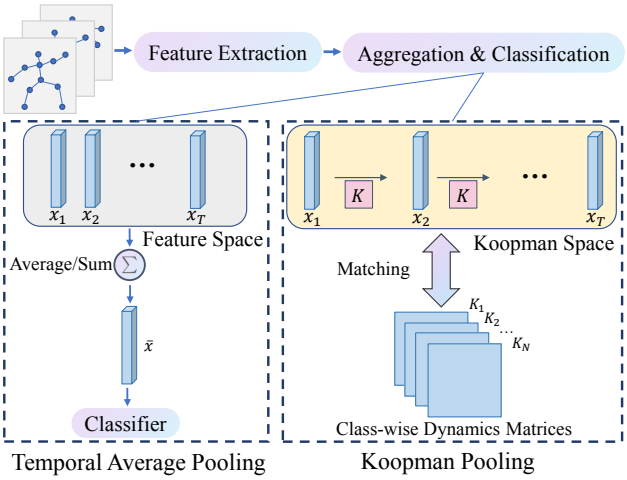

Figure 1. Most existing works use temporal average pooling to aggregate the information along the temporal dimension (left), and only first-order information is considered. Our proposed Koopman pooling (right) instead focuses on the true latent dynamics of the sequence in linear Koopman space, and learns a set of class-wise Koopman dynamics matrices to represent the dynamics of each class. The classification is achieved by dynamics matching.

To this end, recent works focus on second-order pooling to capture second-order information of the feature sequences. Specifically, bilinear and covariance pooling is widely used to extract second-order statistical information from the feature sequence. Early works [14] use simple bilinear pooling to model the interaction of features and aggregate temporal information. In recent years, researchers proposed to use covariance pooling [22] to capture second-order statistical information, as covariance matrix can model the interaction between features while maintaining good geometrical structure [31]. However, the skeleton sequence as well as the extracted feature sequence has complex underlying dynamics in nature. Existing methods such as covariance pooling only exploit the feature interaction between frames or channels, but they fail to discover the true dynamics of the sequence. Instead, we aim

to directly focus on the temporal dynamics of the sequence and conduct sequence recognition based on class-specific dynamics.

A critical motivating idea of this work is the application of Koopman theory [26]. The original Koopman theory aims to re-formulate a non-linear dynamical system to be a linear one. To this end, a Koopman operator is required to lift the original features into some possibly infinite-dimensional Hilbert space, wherein the evolution of the dynamics becomes linear. Such a treatment is favored in numerous applications, including time-series analysis, since an arsenal of spectral-analysis tools can facilitate the in-depth investigation of the model (such as the temporal stability property). In practice, identifying the optimal Koopman operator of a specific dynamic system remains a challenge. Besides conventional dynamic mode decomposition (DMD) [6], recent years also witnessed the utilization of black-box neural network for learning the Koopman operator in differentiable fashion [1, 23, 38, 42, 61].

To our best knowledge, the proposed Koopman pooling in this paper is the first work to leverage the power of Koopman theory to formulate a new high-order pooling method. Unlike existing methods like covariance pooling which use covariance matrix to model the temporal correlation of features implicitly, we instead view the temporal evolution of feature vectors as a dynamical system, and use the dynamics itself to model the temporal correlation explicitly. As shown in figure 1, the original trajectories are mapped to a new embedding space where the temporal evolution is linear. The transition matrix of this linear system can therefore be viewed as the signature of this sequence, which contains rich high-order information. For the classification task, our model learns the class-wise Koopman matrices which represent class-specific dynamics, and conducts dynamics matching to obtain the classification score. Based on our observation of the learned dynamics, this paper highlight the critical importance of the stability of learned dynamics when tackling recognition tasks, and propose an eigenvalue normalization technique to push the learned linear dynamics to be stable and non-decaying. We also combine Koopman pooling and dynamic mode decomposition(DMD) to formulate a new framework for one-shot action recognition, which uses dynamics to match the sequence instead of the common practice of computing distance in the embedding space or conducting metric learning.

To verify the effectiveness of the proposed method, we conduct extensive experiments on 3 skeleton-based action recognition datasets, namely NTU RGB+D, NTU RGB+D 120, and NW-UCLA. The results demonstrate that Koopman pooling significantly improves the performance under both full-dataset and one-shot settings.

To be summarized, the main contributions of this paper are as follows:

- We proposed Koopman pooling, which is the first work in the literature to design a plug-and-play high-order pooling method based on Koopman theory that allows eigenvalue manipulation and one-shot recognition.
- We emphasize the critical importance of learning a stable and non-decaying system for recognition tasks and accordingly design an eigenvalue normalization technique based on control theory.
- Our comprehensive experiments based on various backbones [9, 28, 68] on the commonly used benchmark datasets NTU RGB+D 60/120 and NW-UCLA shows Koopman pooling significantly improves the performance under both full-dataset and one-shot setting.

2. Related work

2.1. Temporal Pooling Methods

Recently, there is a growing research interest in improving vanilla temporal average pooling by introducing higher-order statistics. Some works [14, 25, 34, 35, 86] use bilinear pooling to capture pairwise interactions among CNN features of adjacent frames or different channels. Recently, researchers tackle second-order pooling from a covariance perspective and introduce covariance matrix as the second-order pooled features. Spatial covariance pooling is successfully applied to image classification task [18, 22, 30, 31, 58, 59, 67, 76, 83]. For temporal pooling, Girdhar et al. [19] introduce an attention module that is equivalent to a rank-1 approximation of second-order pooling. Gao et al. [17] proposes a temporal-attentive covariance pooling for generating powerful video representations. Dong et al. [15] integrates the correlation computation and covariance pooling to model high-order temporal features of videos.

2.2. Deep Koopman Model

Koopman operator [26] was originally introduced to tackle non-linear dynamical systems in physics. Recently, there is a growing research interest in the combination of deep learning and Koopman theory [2, 20, 23, 38, 42, 47, 61]. Most existing works [1, 2, 38] adopt the auto-encoder framework and use specially designed loss functions to ensure the linear evolution of the system. However, they mainly focus on the task of sequence prediction, and few works look into recognition tasks. Zhang et al. [81] design a Koopman model for gait recognition by constructing linear embeddings. Nevertheless, their model still adopts an auto-encoder structure without an end-to-end trainable classifier, which fails when generalized to other backbones or settings (such as one-shot recognition). Instead, our proposed Koopman pooling is an end-to-end trainable, plug-and-play second-order pooling module that can be inserted into any

spatial-temporal backbone, thus allowing eigenvalues manipulation and one-shot recognition. Also, our work is the first to stress the need of a stable and non-decaying dynamics for recognition and design an eigenvalue normalization based on stability theory.

2.3. Skeleton-based Action Recognition

Traditional methods [3–5, 43, 44, 48, 49, 63, 75] for skeleton action recognition mainly use hand-crafted dynamic features and geometric relationships to model the spatio-temporal action sequence. Recently, with the success of deep learning, deep neural network has been the main workhorse for action recognition. Early works [32, 33, 55, 56, 62, 77, 80, 85] use recurrent neural networks(RNN) as the backbone to extract the temporal information of the time series of human joints. However, RNN is known to suffer from gradient explosion and vanishing. Some researchers [7, 21, 29, 66, 71, 78] propose to apply convolutional neural nets(CNN) to better extract the spatial and temporal information from the skeleton sequence by performing spatial-temporal convolution. Recently, GCN-based methods are proposed to tackle the task by modeling the skeleton sequence as a spatial-temporal graph. The idea is straightforward as the skeleton itself is a graph in nature and has its own topology. A large proportion of works follow this track [9, 10, 13, 24, 28, 37, 46, 53, 54, 70, 72–74, 82]. For instance, CTR-GCN [9] proposes to dynamically model channel-wise topologies in a refinement approach, and reform graph convolution to relax the strict constraints, which leads to better representation capability.

3. The Proposed Neural Koopman Pooling

3.1. Preliminary on Classic Koopman Theory

Let us first briefly review the classic Koopman theory before diving into the details of the proposed Koopman pooling model. Suppose a discrete-time nonlinear dynamical system can be described by

$$y_{k+1} = F(y_k), \;\; Y \in \mathcal{M} \subset \mathbb{R}^d, \tag{1}$$

where $y_k, y_{k+1} \in Y$ are state vectors on the state space $\mathcal{M}$, and F represents some state-transition function. For a particular choice of feature representation y_k, the dynamics are typically with high-level non-linearity. The Koopman theory instead lifts y_k into another higher-dimensional space, wherein the evolution of the states is linear.

Formally, Koopman operator $\mathcal{K}$ acts on scalar functions $\phi : \mathcal{M} \to \mathbb{R}$, which are elements of an infinite-dimensional Hilbert space:

$$x_k \;\; = \;\; \phi(y_k), \tag{2}$$

where x_k is a new (possibly) infinite-dimensional representation induced by ϕ. And the Koopman operator can be defined as a composition:

$$\mathcal{K}\phi \;\; = \;\; \phi \circ F, \tag{3}$$

which can be equivalently expressed as below:

$$\mathcal{K}\phi(y_k) = \phi(F(y_k)) = \phi(y_{k+1}). \tag{4}$$

One can view Koopman operator $\mathcal{K}$ as an evolution of functions in the Hilbert space of all possible ϕ. However, Koopman operator $\mathcal{K}$ is not tractable because it is infinite-dimensional. We can use some finite-dimension matrix $\mathbf{K}$ as an approximation of the Koopman operator, which results in Eq. (5):

$$x_{k+1} = \mathbf{K} x_k. \tag{5}$$

Finite-dimensional approximation of Koopman operator $\mathcal{K}$ is popularly realized by Dynamic Mode Decomposition (DMD) [6, 50, 69]. Let $\mathbf{X}_{1:T-1}$ denotes $[x_1, x_2, \ldots, x_{T-1}]$ and $\mathbf{X}_{2:T}$ denotes $[x_2, x_3, \ldots, x_T]$. The DMD algorithm seeks the best-fit linear operator $\mathbf{K}$ such that

$$\mathbf{X}_{2:T} = \mathbf{K}\mathbf{X}_{1:T-1} \tag{6}$$

The best-fit mathematical solution is

$$\mathbf{K} = \arg\min_{\mathbf{K}} \|\mathbf{X}_{2:T} - \mathbf{K}\mathbf{X}_{1:T-1}\|_F = \mathbf{X}_{2:T}\mathbf{X}^{\dagger}_{1:T-1}, \tag{7}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and † denotes pseudo-inverse.

3.2. Problem Formulation

In skeleton-based action recognition, an input $\hat{\mathbf{X}}$ contains coordinates for V joints over T time-steps, namely $\hat{\mathbf{X}} = (\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_T)$. A feature extraction backbone $\mathcal{F}$ (such as CNN or GCN) is applied to extract spatial-temporal features from input $\hat{\mathbf{X}}$, resulting in feature sequence $\mathbf{X} = \mathcal{F}(\hat{\mathbf{X}}) = (x_1, x_2, \ldots, x_T)$, where $x_t \in \mathbb{R}^C$ and C denotes the feature dimension. In this paper, we focus on obtaining the classification score from features $\mathbf{X}$. Existing pooling methods can be categorized as follows:

Vanilla Average Pooling. Temporal dimension is reduced by average pooling, and a classifier such as fc-layer is then applied to derive classification score:

$$score = FC\left(\frac{1}{T}\sum_{t=1}^{T} x_t\right). \tag{8}$$

Covariance Pooling. The covariance matrix is used as the feature. However, plain covariance pooling still aggregates the features along temporal dimension orderlessly and abandons temporal dynamics information.

$$score = FC\left(\frac{1}{T}\sum_{t=1}^{T} x_t x_t^T\right). \tag{9}$$

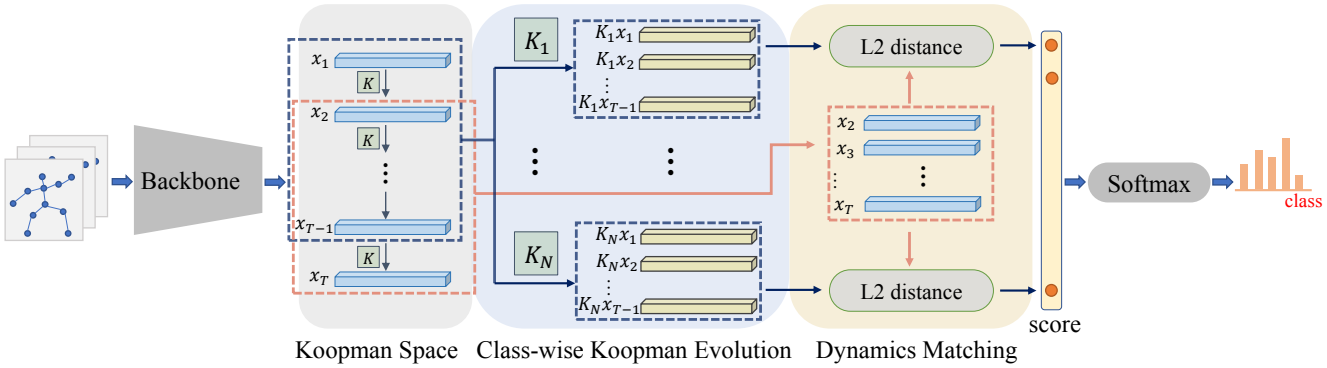

Figure 2. **Overall architecture of the proposed Koopman pooling model.** Skeleton sequence is first fed to the backbone to extract spatial-temporal information. Then the feature sequence $x_1 \sim x_{T-1}$ is evolved by each class-wise dynamics matrix $\mathbf{K}_i$. L2 linear fitting error of i-th class is calculated as $\sum_{t=1}^{T} \|x_t - \mathbf{K}_i x_{t-1}\|^2$. The opposite of linear fitting error is then used as the class activation score, therefore the best-fit $\mathbf{K}_i$ indicates the classification result i. Class-wise dynamics matrix $\mathbf{K}_i$ is further processed by eigenvalues normalization, which is described in Sec. 3.4.

Temporal Bilinear Pooling. Temporal bilinear pooling is capable of capturing the interaction between temporal features by calculating weighted inner product:

$$score = FC\left(\frac{1}{T-1}\sum_{t=1}^{T-1} x_t^T \mathbf{W} x_{t+1}\right). \tag{10}$$

3.3. Class-wise Dynamics based Koopman Pooling

The pooling methods mentioned in Sec. 3.2 either reduce the temporal dimension by simple averaging operation or just implicitly model the interaction through the inner product. They cannot fully exploit the complex dynamics of the feature sequences. Instead, we aim to directly focus on their true dynamics. As a highly non-linear dynamical system is difficult to tackle, inspired by Koopman theory, we hope that in the feature space, the temporal dynamic of the sequence is linear:

$$x_{t+1} = \mathbf{K} x_t \tag{11}$$

where $\mathbf{K}$ is the linear transition matrix. $\mathbf{K}$ can therefore be viewed as a kind of "pooled feature/signature" for the sequence, as it summarized the temporal dynamics of the whole sequence. $\mathbf{K}$ can be derived by DMD:

$$\mathbf{K} = \mathbf{X}_{2:T}\mathbf{X}^{\dagger}_{1:T-1}. \tag{12}$$

In the classification task, as each class has its specific dynamics, N learnable matrices $\mathbf{K}_i \in \mathbb{R}^{C\times C}$ (where C denotes feature dimension and N is the number of classes) is set to represent the linear dynamics of each class. During classification, when given a specific sample, its dynamics matrix derived from Eq. (12) is compared with each $\mathbf{K}_i$ and find the nearest one. A naive way of performing such matching is to calculate ℓ_2 distance between matrices:

$$c = \underset{i\in\{1,\cdots,N\}}{\arg\min} \|\mathbf{K} - \mathbf{K}_i\|_F. \tag{13}$$

However, the performance of this simple method turns out to be poor, as gradient back-propagation through pseudo-inverse in Eq. (12) is highly unstable. Instead, to compare two dynamics, we propose to use the ℓ_2 distance of 1-step linear evolution on the feature $\mathbf{X}$ by $\mathbf{K}$ and $\mathbf{K}_i$, namely

$$c = \underset{i\in\{1,\cdots,N\}}{\arg\min} \|(\mathbf{K} - \mathbf{K}_i)\mathbf{X}_{1:T-1}\|_F. \tag{14}$$

The advantage of choosing the above distance is its equivalence to the linear fitting error of $\mathbf{K}_i$ which greatly stabilized the training process:

$$\begin{aligned}\|(\mathbf{K} - \mathbf{K}_i)\mathbf{X}_{1:T-1}\|_F &= \|\mathbf{K}X_{1:T-1} - \mathbf{K}_i\mathbf{X}_{1:T-1}\|_F \\ &= \|\mathbf{X}_{2:T} - \mathbf{K}_i\mathbf{X}_{1:T-1}\|_F.\end{aligned} \tag{15}$$

The above equation is derived from Eq. (12), as $\mathbf{K}\mathbf{X}_{1:T-1} = \mathbf{X}_{2:T}\mathbf{X}^{\dagger}_{1:T-1}\mathbf{X}_{1:T-1} = \mathbf{X}_{2:T}$ holds when the dimension of feature (*i.e.*, the dimension of $\mathbf{K}$) is larger than temporal dimension T, which is true in our case. Under such formulation, the class activation score for i-th class is calculated as

$$score_i = -\|\mathbf{X}_{2:T} - \mathbf{K}_i\mathbf{X}_{1:T-1}\|_F. \tag{16}$$

During training, the class-wise linear dynamics matrices $\mathbf{K}_1, \cdots, \mathbf{K}_N$ are learned in an end-to-end fashion together with the feature extraction backbone. We use cross-entropy loss to jointly train the backbone and dynamics matrices:

$$\mathcal{L} = CE(score, y), \tag{17}$$

where y is the ground-truth label. For a specific sample from class i, the cross-entropy loss encourages the linear fitting error of $\mathbf{K}_i$ to be low and the error of $\mathbf{K}_j, j \neq i$ to be high.

3.4. Stability-assured Eigenvalue Normalization

Koopman methods enjoy better theoretical interpretability compared to other time-series models such as RNN. Specifically, the spectrum and eigenvalues of Koopman matrix $\mathbf{K}$ are critically important for system dynamics evolution. Existing Koopman methods [1, 16, 45] mainly focus on the stability of the system as it is a key issue in long-term prediction and system evolution. However, few works look into the impact of stability in recognition tasks. In this section, we emphasize the vital importance of stability in recognition tasks and propose an eigenvalue normalization technique to push the learned dynamics to be stable and non-decaying.

Suppose $\lambda_1, \lambda_2, \ldots, \lambda_C \in \mathbb{C}$ are the eigenvalues of $\mathbf{K}$ and $v_1, v_2, \ldots, v_C \in \mathbb{C}^n$ are the corresponding eigenvectors, where C is the dimension of $\mathbf{K}$. Then the Koopman mode decomposition [41] can be written as:

$$\mathbf{K}^t x_1 = \mathbf{K}^t \sum_j \alpha_j v_j = \sum_j \lambda_j^t \alpha_j v_j. \tag{18}$$

From Eq. (18), we can see the modulus of λ_j determines whether the j-th mode is decaying, static or divergent. An eigenvalue with a modulus smaller than one will cause the mode to eventually decay to zero over time, while an eigenvalue with a modulus greater than one will cause the system to be unstable and divergent.

In the above Koopman pooling formulation, since there's no explicit constraint on the norm of eigenvalues, the majority of the learned eigenvalues have a modulus smaller than one, causing the system to decay. Figure 3 shows some failure cases due to the decaying system. As shown, we visualize the trajectories in the linear Koopman space by applying PCA dimension reduction to embed the features into a 2-dimensional plane for visualization. 'original' denotes the feature sequence $\mathbf{X} = [x_1, x_2, \ldots, x_T]$, and '$i$' denotes the 1-step linear evolution of $\mathbf{X}$ by the i-th class-wise dynamics matrix $\mathbf{K}_i$, i.e. $\mathbf{K}_i\mathbf{X} = [\mathbf{K}_i x_1, \mathbf{K}_i x_2, \ldots, \mathbf{K}_i x_T]$. Ideally, we have $[x_1, x_2, \ldots, x_{T-1}] = [\mathbf{K}_i x_2, \mathbf{K}_i x_2, \ldots, \mathbf{K}_i x_T]$ when i equals to the ground-truth class of the action sequence. Notice that the 1-step linear evolution trajectories decay significantly compared to the original trajectories. The key observation here is decaying system leads to imperfect linear fitting and therefore class mismatch during classification, which greatly lowers the recognition accuracy. Likewise, an unstable system will face the same issue, as divergence leads to imperfect linear fitting, too. To overcome this problem and further boost the recognition accuracy, we propose an eigenvalue normalization technique to ensure the stability and non-decaying of the learned dynamics.

For each class-wise dynamics matrix $\mathbf{K}_i$, $i = 1 \ldots N$, suppose the eigen-decomposition of $\mathbf{K}_i$ is $\mathbf{K}_i = \mathbf{U}\mathbf{D}\mathbf{U}^{-1}$, where $\mathbf{D} = diag\{\lambda_1, \lambda_2, \ldots, \lambda_C\}$. Our proposed eigenvalue normalization is formulated as:

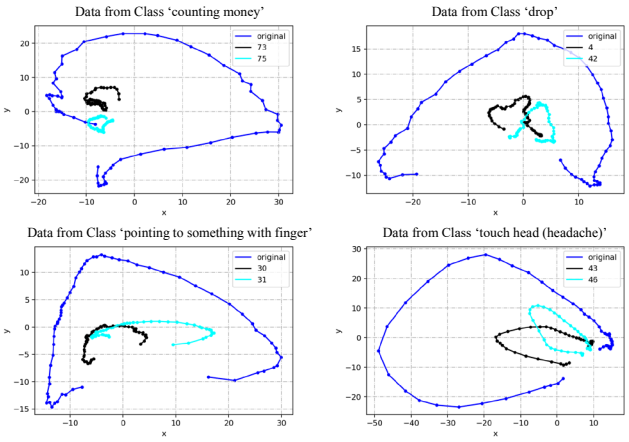

Figure 3. **Visualization of original trajectories and their 1-step linear evolution by the class-wise dynamics matrices $\mathbf{K}_i$ in failure cases.** The ground-truth class name is annotated above each figure as 'Data from Class x' and the corresponding trajectory is plotted in black. Ideally, the blue lines (original trajectories) and the black lines (evolved trajectories by dynamics matrix of ground-truth class) should coincide. As the norm of evolved trajectories $\mathbf{K}_i\mathbf{X}$ is significantly smaller than original trajectories $\mathbf{X}$ due to the decaying system, class mismatch is more likely to happen. Best viewed in color.

$$\hat{\lambda}_j = \lambda_j * \frac{|\lambda_j|^p}{|\lambda_j|}, \tag{19}$$

where $p < 1$ is the normalization factor. Then the normalized dynamics matrix $\hat{\mathbf{K}}_i$ is calculated as $\hat{\mathbf{K}}_i = \mathbf{U}\hat{\mathbf{D}}\mathbf{U}^{-1}$, where $\hat{\mathbf{D}} = diag(\hat{\lambda}_1, \hat{\lambda}_2, \cdots, \hat{\lambda}_C)$. The key motivation of designing the above normalization techniques is $|\hat{\lambda}_j| > |\lambda_j|$ for $|\lambda_j| < 1$ and $|\hat{\lambda}_j| < |\lambda_j|$ for $|\lambda_j| > 1$. This ensures that the normalization pushes the linear system to be non-decaying and stable.

As the gradients of eigen-decomposition are numerically unstable and computationally expensive, we proposed a 2-stage training protocol. First, the class-wise Koopman dynamics matrices are optimized jointly with feature extraction backbone. Next, the learned dynamics matrices are normalized and frozen. Only the feature extraction backbone is optimized to map the sequences to the linear space governed by the normalized dynamics matrices.

3.5. Koopman Pooling for One-shot Recognition

Koopman pooling can be viewed as a kind of temporal dynamics matching. With only one exemplar sample provided under one-shot setting, fully exploiting the information of it for further matching becomes critical. Therefore, it is straightforward to extend Koopman pooling to the task of one-shot action recognition. In one-shot action recognition, the testing set consists of novel classes that didn't appear in the training stage. One sample from each novel class

Methods	NTU-RGB+D 60		NTU-RGB+D 120		Northwestern UCLA (%)
	X-Sub (%)	X-View (%)	X-Sub (%)	X-Set (%)	
ST-GCN [72]	81.5	88.3	70.7	73.2	-
2s-AGCN [53]	88.5	95.1	82.5	84.2	-
SGN [79]	89.0	94.5	79.2	81.5	92.5
AGC-LSTM [55]	89.2	95.0	-	-	93.3
DGNN [52]	89.9	96.1	-	-	-
Shift-GCN [12]	90.7	96.5	85.9	87.6	94.6
DC-GCN+ADG [11]	90.8	96.6	86.5	88.1	95.3
DDGCN [27]	91.1	97.1	-	-	-
MS-G3D [37]	91.5	96.2	86.9	88.4	-
MST-GCN [10]	91.5	96.6	87.5	88.8	-
EfficientGCN-B4 [60]	91.7	95.7	88.3	89.1	-
Dynamic GCN [74]	91.5	96.0	87.3	88.6	-
CTR-GCN [9]	92.4	**96.8**	88.9	90.6	96.5
CTR-GCN* [9]	92.4	96.4	88.9	90.4	96.5
CTR-GCN w/ Koompan Pooling	**92.9**	**96.8**	**90.0**	**91.3**	**97.0**

Table 1. **Classification accuracy comparison of our method with CTR-GCN and some other existing models on the NTU-RGB+D 60, 120, and NW-UCLA datasets.** X-Sub denotes cross-subject, X-View denotes cross-view and X-Set denotes cross-setup. CTR-GCN denotes the reported performance in [9] and CTR-GCN* denotes the reproduced results by re-running the official codes from [9]. Our CTR-GCN + Koopman Pooling model outperforms the original CTR-GCN model by a non-trivial margin.

is provided as the exemplar, and the model is required to yield classification results on other samples from the novel classes. Existing methods [8, 36, 65] mainly train a feature extraction backbone on seen classes, and then use some kind of matching techniques in the embedding space during testing, such as ℓ_2-matching. However, these matching techniques still rely on temporal average pooling, which ignores dynamical information of sequences. Instead, we aim to fully exploit the dynamics information of exemplars and match the sequences according to their dynamics.

Our proposed Koopman pooling framework can be readily applied to one-shot action recognition when combined with Dynamic Mode Decomposition (DMD) [6]. During training, we follow the same process with the full-dataset setting. A feature extraction backbone is trained on seen classes, which maps the original action sequences into a linear space. Then during testing, the exemplar sequences from M novel classes are first fed to the backbone, resulting in feature sequences $\mathbf{X}^i = (x_1^i, x_2^i, \dots, x_T^i)$, where $i = 1, 2, \dots, M$. The class-wise dynamics matrices $\mathbf{K}_1, \mathbf{K}_2, \dots, \mathbf{K}_M$ are calculated as

$$\mathbf{K}_i = \mathbf{X}^i_{2:T}(\mathbf{X}^i_{1:T-1})^\dagger, i = 1, 2, \dots, M. \tag{20}$$

$\mathbf{K}_i$ summarizes the temporal dynamics of exemplar sequence and can be viewed as the class-specific linear dynamics prototype. For each given testing sample, suppose its extracted feature is $\mathbf{X} = (x_1, x_2, \dots, x_T)$, the classification result can be derived as

$$c = \underset{i \in \{1,\dots,M\}}{\arg\min} \|\mathbf{X}_{2:T} - \mathbf{K}_i \mathbf{X}_{1:T-1}\|_F. \tag{21}$$

Such matching protocol fully exploits the dynamics information of the exemplar and is better than the common practice of calculating distance in the embedding space for matching. Notice that under one-shot setting, the class-wise dynamics matrices $\mathbf{K}_i$ do not need further eigenvalue normalization, as $\mathbf{K}_i$ for each novel classes is calculated via DMD in this case rather than learned from data.

4. Experiments

4.1. Data Description and Evaluation Protocol

NTU RGB+D [51] contains 56,880 skeleton action sequences which are categorized into 60 classes, performed by 40 subjects. Each sequence is captured by three Microsoft Kinect v2 cameras. There're two standard evaluation protocols for this dataset: (1) cross-subject(CS), under which the 40 subjects are split into 20 training subjects and 20 testing subjects. (2) cross-view(CV), under which the training set contains sequences from cameras 2 and 3, and the testing set contains sequences from camera 1.

NTU RGB+D 120 [36] is an extension of NTU RGB+D, which contains 113945 skeleton action sequences of 120 classes, performed by 106 subjects. This dataset contains 32 setups (location and background). There're two standard evaluation protocols for this dataset: (1) cross-subject(CSub), under which the 106 subjects are split into 53 training subjects and 53 testing subjects. (2) cross-setup(CSet): under which the training set contains sequences with even setup IDs and the testing set contains sequences with odd setup IDs.

Northwestern-UCLA [64] has 1494 skeleton sequences of 10 classes performed by 10 subjects. Each sequence is captured by three Kinect cameras from different views. Following previous works, we use the sequences from the first two cameras as the training set and the sequences from the

Dataset	Modality	Model (%)		
		CTR-GCN	Koopman	E-Koopman
NTU120 Xsub	joint	84.9	85.6	**85.7**
	bone	85.7	87.1	**87.2**
	joint motion	81.4	82.5	**82.7**
	bone motion	81.2	83.0	**83.1**
NTU120 Xset	joint	86.4	87.2	**87.4**
	bone	87.9	88.1	**88.3**
	joint motion	83.0	84.2	**84.4**
	bone motion	83.0	**84.1**	**84.1**
NTU60 Xsub	joint	89.9	89.9	**90.2**
	bone	**90.6**	90.3	**90.6**
	joint motion	88.1	88.4	**88.5**
	bone motion	87.9	87.9	**88.0**
NTU60 Xview	joint	94.5	94.9	**95.2**
	bone	**94.7**	94.5	**94.7**
	joint motion	93.1	93.1	**93.3**
	bone motion	92.2	92.1	**92.4**

Table 2. **Classification accuracy comparison of our models and CTR-GCN on the four modalities of NTU RGB+D 60, 120.** In the table, "Koopman" denotes CTR-GCN + Koopman pooling (without eigen-normalization). "E-Koopman" denotes CTR-GCN + Koopman pooling (with eigen-normalization).

last camera as the testing set.

One-shot evaluation protocol. We also conduct one-shot action recognition experiments on NTU RGB+D 120 and NW-UCLA dataset. For NTU RGB+D 120, following prior works [8,36,39,40,65,84], the dataset is split into auxiliary set and evaluation set. The auxiliary set contains all sequences from 100 classes, and the evaluation set contains 20 novel classes. For each evaluation class, one sample is selected as the exemplar and the remaining are used for testing. For NW-UCLA, sequences from 10 classes are used as the auxiliary set, and the remaining are used for evaluation.

4.2. Implementation Details

The evaluations mainly adopt CTR-GCN [9] as the base model, a modern representative GCN-based recognition model that inspired many follow-up works [13, 28, 46, 73]. Nonetheless, we would emphasize that the proposed Koopman pooling method is a generic plug-and-play module. By substituting the last temporal average pooling layer, it can be inserted into most spatial-temporal recognition backbones. Tab. 4 also briefly presents the benefits brought by Koopman pooling + other GCN variants. To ensure fair comparison, we use the same hyperparameters as the original implementation of CTR-GCN [9] (learning rate, optimizer, weight decay, data preprocessing, etc.). The normalization factor p in Eq. (19) is set to 0.25. Random seed is set to 1, following [9].

Eq. (12) is not the unique solution of Eq. (11), as the dimension of solution space for this linear equation is $C - T$ (C is the feature dimension). CTR-GCN embedding is 256-dimensional. Larger temporal length T will lead to a smaller solution space and more stable recognition. Thus We increase its temporal length to 64 (originally 16) by removing the stride in temporal convolutions. Following prior practice, experiments are conducted on 4 different modalities (joint, bone, joint motion, and bone motion), and classification result is obtained by ensembling. For overall results, as the original CTR-GCN model realized by temporal average pooling contains first-order information and our Koopman model contains high-order dynamics information, we obtain the results by fusing them through ensembling the classification score of them.

Methods	NTU120 (%)	UCLA (%)
APSR [36]	45.3	-
SL-DML [40]	50.9	-
Skeleton-DML [39]	54.2	-
JEANIE [65]	57.0	-
ALCA-GCN [84]	57.6	-
Part-aware [8]	65.6	83.3
ST-GCN+ProtoNet [57]	61.1	79.8
MS-G3D+ProtoNet [57]	59.5	81.2
CTR-GCN+ProtoNet [57]	58.8	80.7
CTR-GCN+Koompan Pooling	**68.1**	**89.9**

Table 3. **Oneshot classification accuracy comparison against state-of-the-art methods on the NTU RGB+D 120 and NW-UCLA datasets.**

4.3. Quantitative Evaluations

Full-dataset setting. Tab. 1 presents the experimental results on all three datasets. As observed, the proposed Koopman pooling elevates the accuracy of CTR-GCN by a non-trivial margin. To further demonstrate the effectiveness of Koopman pooling, we compare the performance of the original CTR-GCN and our methods on each modality in Tab. 2. In the table, "Koopman" refers to CTR-GCN + Koopman pooling, and "E-Koopman" refers to Koopman pooling with eigenvalue normalization. As shown, after replacing the temporal average pooling layer of CTR-GCN with the proposed Koopman pooling, the performance on nearly every modality has improved. For example, on the large-scale dataset NTU RGB+D 120, for most modalities the performance has a gain larger than 1%. The effect of eigenvalue normalization is also demonstrated, and for most cases the improvement is around 0.2-0.3%, which is not significant but very consistent.

One-shot setting. Tab. 3 shows the performance of our proposed methods against existing one-shot methods on NTU RGB+D 120 and NW-UCLA datasets. In the table, the performances of CTR-GCN,MS-G3D, ST-GCN w/ ProtoNet [57] are from [8]. As shown, Koopman pool-

Models	joint (%)	bone (%)
CTR-GCN [9]	84.9	85.7
CTR-GCN w/o stride	84.6	86.2
CTR-GCN w/ Koopman Pooling	**85.7**	**87.2**
HD-GCN [28]	85.3	86.7
HD-GCN w/ Koopman Pooling	**85.7**	**87.3**
TCA-GCN [68]	85.1	86.8
TCA-GCN w/ Koopman Pooling	**85.3**	**87.4**

Table 4. **Ablation study on NTU RGB+D 120 cross-subject dataset.** CTR-GCN w/o stride denotes the baseline model that removes the stride in the temporal convolution of CTR-GCN. w/ Koopman Pooling denotes our method that substitutes the last temporal average pooling layer with proposed Koopman pooling.

ing achieves outstanding performance compared to existing methods for its advantage of exploiting temporal dynamics.

4.4. Ablation Study

To demonstrate that the performance improvement comes from Koopman pooling, we design two ablative experiments. As mentioned in 4.2, the stride in the temporal convolution of CTR-GCN is removed to prolong the temporal dimension, thus a baseline of CTR-GCN without stride is added to ensure the improvement doesn't come from this removal. Also, as Koopman pooling is a plug-and-play module that applies to any spatial-temporal backbones, we wish to further prove its effectiveness across various backbones. To this end, two recent works HD-GCN [28] and TCA-GCN [68] are adopted, which are GCN-based skeleton action recognition frameworks. The results on NTU RGB+D 120 cross-subject are shown in Tab. 4, which prove the generality of Koopman pooling.

4.5. Visualization

To demonstrate that the learned dynamics matrix $\hat{\mathbf{K}}_i$ indeed represents class-specific linear dynamics, we visualize the trajectories in the linear Koopman space using the same method as in Figure 3. Apart from the ground-truth class, we set i to be 10, 20, 30, 40, 50 and visualize the trajectory evolved by $\hat{\mathbf{K}}_i$ as a comparison. As shown, the trajectories of 1-step linear evolution by the dynamic matrix of ground-truth class manifest similar patterns with the original sequence, while the trajectories of 1-step linear evolution by the dynamic matrix of other classes seem random. This indicates that the learned dynamics matrix $\hat{\mathbf{K}}_i$ indeed contains class-specific temporal dynamics information.

Another key observation is that the trajectories of the same class exhibit highly similar patterns. For instance, the trajectories of two samples from class 18 share analogous shapes, norms and positions. This implies that with Koopman pooling, the backbone itself can map the action sequences of each class into corresponding class-specific linear space. Also, when comparing Figures 3 and 4, one

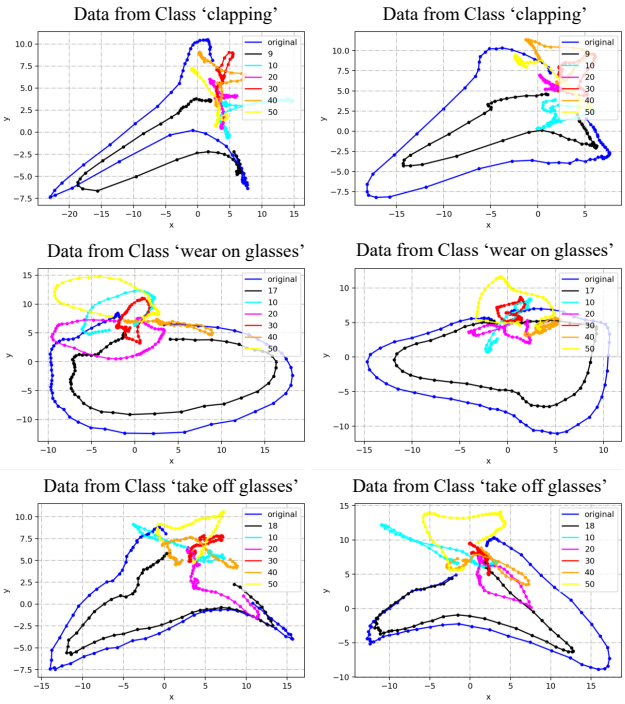

Figure 4. **Visualization of original trajectories and their 1-step linear evolution by the class-wise dynamics matrices $\hat{\mathbf{K}}_i$.** The ground-truth class name is annotated above each figure as "Data from Class x" and the corresponding trajectory is plotted in black. Best viewed in color and see the main text for more explanation.

can see that after eigenvalue normalization, the problem of norm decaying is greatly alleviated, leading to better linear fitting and more accurate dynamics matching.

5. Conclusion

This work presents a plug-and-play parameterized high-order pooling module called Koopman pooling. Unlike average/max pooling which abandons higher-order information, this work instead focuses on the true dynamics of the sequences. Koopman operator is deployed to ensure linear evolution of the temporal dynamics. The system is represented by the dynamics matrices which can then be used for classification. An eigenvalue normalization technique based on stability theory is proposed to ensure stable and non-decaying dynamics to further improve recognition performance. The proposed Koopman pooling method combined with Dynamic Mode Decomposition (DMD) can also be readily applied to construct a new one-shot action recognition framework. Extensive experiments demonstrate that Koopman pooling attains remarkably better performance under both full-dataset and one-shot settings.

Acknowledgement: This work is supported by National Key R&D Program of China (2020AAA0104401), Beijing Natural Science Foundation (Z190001) and Peng Cheng Laboratory Key Research Project No.PCL2021A07.

References

[1] Omri Azencot, N Benjamin Erichson, Vanessa Lin, and Michael Mahoney. Forecasting sequential data using consistent koopman autoencoders. In *International Conference on Machine Learning*, pages 475–485. PMLR, 2020. 2, 5

[2] Kaushik Balakrishnan and Devesh Upadhyay. Stochastic adversarial koopman model for dynamical systems. *arXiv preprint arXiv:2109.05095*, 2021. 2

[3] Aaron F Bobick. Movement, activity and action: the role of knowledge in the perception of motion. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 352(1358):1257–1265, 1997. 3

[4] Aaron F. Bobick and James W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on pattern analysis and machine intelligence*, 23(3):257–267, 2001. 3

[5] Matthieu Bray, Pushmeet Kohli, and Philip HS Torr. Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. *ECCV (2)*, 2, 2006. 3

[6] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019. 2, 3, 6

[7] Congqi Cao, Cuiling Lan, Yifan Zhang, Wenjun Zeng, Hanqing Lu, and Yanning Zhang. Skeleton-based action recognition with gated convolutional neural networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(11):3247–3257, 2018. 3

[8] Tailin Chen, Desen Zhou, Jian Wang, Shidong Wang, Qian He, Chuanyang Hu, Errui Ding, Yu Guan, and Xuming He. Part-aware prototypical graph network for one-shot skeleton-based action recognition. *arXiv preprint arXiv:2208.09150*, 2022. 6, 7

[9] Yuxin Chen, Ziqi Zhang, Chunfeng Yuan, Bing Li, Ying Deng, and Weiming Hu. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13359–13368, 2021. 1, 2, 3, 6, 7, 8

[10] Zhan Chen, Sicheng Li, Bing Yang, Qinghan Li, and Hong Liu. Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1113–1122, 2021. 3, 6

[11] Ke Cheng, Yifan Zhang, Congqi Cao, Lei Shi, Jian Cheng, and Hanqing Lu. Decoupling gcn with dropgraph module for skeleton-based action recognition. In *European Conference on Computer Vision*, pages 536–553. Springer, 2020. 6

[12] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 183–192, 2020. 6

[13] Hyung-gun Chi, Myoung Hoon Ha, Seunggeun Chi, Sang Wan Lee, Qixing Huang, and Karthik Ramani. Infogcn: Representation learning for human skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20186–20196, 2022. 1, 3, 7

[14] Ali Diba, Vivek Sharma, and Luc Van Gool. Deep temporal linear encoding networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2329–2338, 2017. 1, 2

[15] Wei Dong, Zhenwei Wang, Bingbing Zhang, Jianxin Zhang, and Qiang Zhang. High-order correlation network for video recognition. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2022. 2

[16] N Benjamin Erichson, Michael Muehlebach, and Michael W Mahoney. Physics-informed autoencoders for lyapunov-stable fluid flow prediction. *arXiv preprint arXiv:1905.10866*, 2019. 5

[17] Zilin Gao, Qilong Wang, Bingbing Zhang, Qinghua Hu, and Peihua Li. Temporal-attentive covariance pooling networks for video recognition. *Advances in Neural Information Processing Systems*, 34:13587–13598, 2021. 2

[18] Zilin Gao, Jiangtao Xie, Qilong Wang, and Peihua Li. Global second-order pooling convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2019. 2

[19] Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. *Advances in neural information processing systems*, 30, 2017. 2

[20] Minghao Han, Jacob Euler-Rolle, and Robert K Katzschmann. Desko: Stability-assured robust control with a deep stochastic koopman operator. In *International Conference on Learning Representations*, 2021. 2

[21] Alejandro Hernandez Ruiz, Lorenzo Porzi, Samuel Rota Bulò, and Francesc Moreno-Noguer. 3d cnns on distance matrices for human action recognition. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1087–1095, 2017. 3

[22] Catalin Ionescu, Orestis Vantzos, and Cristian Sminchisescu. Matrix backpropagation for deep networks with structured layers. In *Proceedings of the IEEE international conference on computer vision*, pages 2965–2973, 2015. 1, 2

[23] Tomoharu Iwata and Yoshinobu Kawahara. Neural dynamic mode decomposition for end-to-end modeling of nonlinear dynamics. *arXiv preprint arXiv:2012.06191*, 2020. 2

[24] Lipeng Ke, Kuan-Chuan Peng, and Siwei Lyu. Towards to-at spatio-temporal focus for skeleton-based action recognition. *arXiv preprint arXiv:2202.02314*, 2022. 3

[25] Shu Kong and Charless Fowlkes. Low-rank bilinear pooling for fine-grained classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 365–374, 2017. 2

[26] Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the national academy of sciences of the united states of america*, 17(5):315, 1931. 2

[27] Matthew Korban and Xin Li. Ddgcn: A dynamic directed graph convolutional network for action recognition. In *European Conference on Computer Vision*, pages 761–776. Springer, 2020. 6

[28] Jungho Lee, Minhyeok Lee, Dogyoon Lee, and Sangyoon Lee. Hierarchically decomposed graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:2208.10741*, 2022. 1, 2, 3, 7, 8

[29] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Skeleton-based action recognition with convolutional neural networks. In *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 597–600. IEEE, 2017. 3

[30] Peihua Li, Jiangtao Xie, Qilong Wang, and Zilin Gao. Towards faster training of global covariance pooling networks by iterative matrix square root normalization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 947–955, 2018. 2

[31] Peihua Li, Jiangtao Xie, Qilong Wang, and Wangmeng Zuo. Is second-order information helpful for large-scale visual recognition? In *Proceedings of the IEEE international conference on computer vision*, pages 2070–2078, 2017. 1, 2

[32] Shuai Li, Wanqing Li, Chris Cook, Ce Zhu, and Yanbo Gao. Independently recurrent neural network (indrnn): Building a longer and deeper rnn. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5457–5466, 2018. 3

[33] Wenbo Li, Longyin Wen, Ming-Ching Chang, Ser Nam Lim, and Siwei Lyu. Adaptive rnn tree for large-scale human action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 1444–1452, 2017. 3

[34] Yanghao Li, Sijie Song, Yuqi Li, and Jiaying Liu. Temporal bilinear networks for video action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8674–8681, 2019. 2

[35] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 1449–1457, 2015. 2

[36] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2684–2701, 2019. 6, 7

[37] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 143–152, 2020. 3, 6

[38] Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):1–10, 2018. 2

[39] Raphael Memmesheimer, Simon Häring, Nick Theisen, and Dietrich Paulus. Skeleton-dml: Deep metric learning for skeleton-based one-shot action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3702–3710, 2022. 7

[40] Raphael Memmesheimer, Nick Theisen, and Dietrich Paulus. Signal level deep metric learning for multimodal one-shot action recognition. *arXiv preprint arXiv:2004.11085*, 2020. 7

[41] Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1):309–325, 2005. 5

[42] Jeremy Morton, Freddie D Witherden, and Mykel J Kochenderfer. Deep variational koopman models: Inferring koopman observations for uncertainty-aware dynamics modeling and control. *arXiv preprint arXiv:1902.09742*, 2019. 2

[43] Eshed Ohn-Bar and Mohan Trivedi. Joint angles similarities and hog2 for action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 465–470, 2013. 3

[44] Omar Oreifej and Zicheng Liu. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 716–723, 2013. 3

[45] Shaowu Pan and Karthik Duraisamy. Physics-informed probabilistic learning of linear embeddings of nonlinear dynamics with guaranteed stability. *SIAM Journal on Applied Dynamical Systems*, 19(1):480–509, 2020. 5

[46] Ruihao Qian, Jiewen Wang, Jianxiu Wang, and Shuang Liang. Structural attention for channel-wise adaptive graph convolution in skeleton-based action recognition. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 01–06. IEEE, 2022. 1, 3, 7

[47] William T Redman, Maria Fonoberova, Ryan Mohr, Yannis Kevrekidis, and Igor Mezic. An operator theoretic view on pruning deep neural networks. In *International Conference on Learning Representations*, 2021. 2

[48] Yann Ricquebourg and Patrick Bouthemy. Real-time tracking of moving persons by exploiting spatio-temporal image slices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):797–808, 2000. 3

[49] Jens Rittscher, Andrew Blake, and Stephen J Roberts. Towards the automatic analysis of complex human body motions. *Image and Vision Computing*, 20(12):905–916, 2002. 3

[50] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010. 3

[51] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016. 6

[52] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7912–7921, 2019. 6

[53] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12026–12035, 2019. 3, 6

[54] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Adasgn: Adapting joint number and model size for efficient skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13413–13422, 2021. 3

[55] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1227–1236, 2019. 3, 6

[56] Chenyang Si, Ya Jing, Wei Wang, Liang Wang, and Tieniu Tan. Skeleton-based action recognition with spatial reasoning and temporal stack learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 103–118, 2018. 3

[57] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017. 7

[58] Yue Song, Nicu Sebe, and Wei Wang. Why approximate matrix square root outperforms accurate svd in global covariance pooling? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1115–1123, 2021. 2

[59] Yue Song, Nicu Sebe, and Wei Wang. On the eigenvalues of global covariance pooling for fine-grained visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2

[60] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. Constructing stronger and faster baselines for skeleton-based action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 6

[61] Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning koopman invariant subspaces for dynamic mode decomposition. *Advances in Neural Information Processing Systems*, 30, 2017. 2

[62] Hongsong Wang and Liang Wang. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 499–508, 2017. 3

[63] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Learning actionlet ensemble for 3d human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):914–927, 2013. 3

[64] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning and recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2649–2656, 2014. 6

[65] Lei Wang, Jun Liu, and Piotr Koniusz. 3d skeleton-based few-shot action recognition with jeanie is not so na\" ive. *arXiv preprint arXiv:2112.12668*, 2021. 6, 7

[66] Pichao Wang, Wanqing Li, Chuankun Li, and Yonghong Hou. Action recognition based on joint trajectory maps with convolutional neural networks. *Knowledge-Based Systems*, 158:43–53, 2018. 3

[67] Qilong Wang, Jiangtao Xie, Wangmeng Zuo, Lei Zhang, and Peihua Li. Deep cnns meet global covariance pooling: Better representation and generalization. *IEEE transactions on pattern analysis and machine intelligence*, 43(8):2582–2597, 2020. 2

[68] Shengqin Wang, Yongji Zhang, Fenglin Wei, Kai Wang, Minghao Zhao, and Yu Jiang. Skeleton-based action recognition via temporal-channel aggregation. *arXiv preprint arXiv:2205.15936*, 2022. 2, 8

[69] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015. 3

[70] Kailin Xu, Fanfan Ye, Qiaoyong Zhong, and Di Xie. Topology-aware convolutional neural network for efficient skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2866–2874, 2022. 1, 3

[71] Yangyang Xu, Jun Cheng, Lei Wang, Haiying Xia, Feng Liu, and Dapeng Tao. Ensemble one-dimensional convolution neural networks for skeleton-based action recognition. *IEEE Signal Processing Letters*, 25(7):1044–1048, 2018. 3

[72] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018. 3, 6

[73] Sen Yang, Xuanhan Wang, Lianli Gao, and Jingkuan Song. Mke-gcn: Multi-modal knowledge embedded graph convolutional network for skeleton-based action recognition in the wild. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 01–06. IEEE, 2022. 1, 3, 7

[74] Fanfan Ye, Shiliang Pu, Qiaoyong Zhong, Chao Li, Di Xie, and Huiming Tang. Dynamic gcn: Context-enriched topology learning for skeleton-based action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 55–63, 2020. 3, 6

[75] Alper Yilmaz and Mubarak Shah. Actions sketch: A novel action representation. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 984–989. IEEE, 2005. 3

[76] Tan Yu, Yunfeng Cai, and Ping Li. Toward faster and simpler matrix normalization via rank-1 update. In *European Conference on Computer Vision*, pages 203–219. Springer, 2020. 2

[77] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *Proceedings of the IEEE international conference on computer vision*, pages 2117–2126, 2017. 3

[78] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive neural networks for high performance skeleton-based human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1963–1978, 2019. 3

[79] Pengfei Zhang, Cuiling Lan, Wenjun Zeng, Junliang Xing, Jianru Xue, and Nanning Zheng. Semantics-guided neural networks for efficient skeleton-based human action recognition. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1112–1121, 2020. 6

[80] Songyang Zhang, Xiaoming Liu, and Jun Xiao. On geometric features for skeleton-based action recognition using multilayer lstm networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 148–157. IEEE, 2017. 3

[81] Shaoxiong Zhang, Yunhong Wang, and Annan Li. Cross-view gait recognition with deep universal linear embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9095–9104, 2021. 2

[82] Rui Zhao, Kang Wang, Hui Su, and Qiang Ji. Bayesian graph convolution lstm for skeleton based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6882–6892, 2019. 3

[83] Heliang Zheng, Jianlong Fu, Zheng-Jun Zha, and Jiebo Luo. Learning deep bilinear transformation for fine-grained image representation. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[84] Anqi Zhu, Qiuhong Ke, Mingming Gong, and James Bailey. Adaptive local-component-aware graph convolutional network for one-shot skeleton-based action recognition. *arXiv preprint arXiv:2209.10073*, 2022. 7

[85] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016. 3

[86] Xinqi Zhu, Chang Xu, Langwen Hui, Cewu Lu, and Dacheng Tao. Approximated bilinear modules for temporal modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3494–3503, 2019. 2