Supplementary Materials for "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

A. Additional Experiment Details

A.1. Model Details

CTR-GCN [2] is mainly used as the base model for the evaluation of our proposed plug-and-play Koopman pooling. The temporal length of the embedding is increased to 64 from 16 by removing the stride in temporal convolutions. Table 1 illustrates the architecture of three CTR-GCN models mentioned in the main text. The feature dimension of CTR-GCN is 256, thus our learned Koopman matrices have the size of 256×256.

A.2. One-shot Datasets Details

NTU RGB+D 120. The dataset is split into an auxiliary set and an evaluation set. Following prior work [3], the evaluation set contains 20 novel classes (namely $A1, A7, A13, \ldots, A115$), and the auxiliary set contains all sequences from the other 100 classes. One sample from each novel class of the evaluation set is selected as the exemplar. Details can be found in [3] and the corresponding Github repository[1].

NW-UCLA. Following the practice in [1], we decompose the dataset into two subsets, either as the evaluation or auxiliary set. The former has 10 classes for training ($A1, A3, A5, A7, A9$), and the latter is comprised of 10 novel classes (namely $A2, A4, A6, A8, A10$). In the evaluation set, a single sample from each novel class is chosen.

A.3. Training Details

To accelerate the training process, we utilize Pytorch Distributed Data Parallel (DDP) to train the model on 4 GPUs simultaneously. The experiments are conducted on four GTX 1080 Ti GPUs. To ensure fair comparison with CTR-GCN, we use the same optimization procedure as the original implementation. The model is trained with SGD with momentum 0.9, weight decay 0.0004, initial learning rate 0.1. The learning rate decays with a factor of 0.1 at epoch 35 and 55. During the second stage where the learned dynamics matrices are normalized and frozen, the feature extraction backbone is tuned with a smaller initial learning rate of 0.01.

For one-shot recognition, class-wise dynamics matrices $\mathbf{K}_i$ are obtained via DMD and the $rcond$ is set to 0.1 when calculating pseudo-inverse. This means that these singular values below $rcond$ are treated as zero and discarded in the computation to ensure the numerical stability.

*Corresponding author
[1]https://github.com/shahroudy/NTURGB-D

Layer	CTR-GCN	w/o stride	w/ Koopman pooling
1	Block(s=1)	Block(s=1)	Block(s=1)
2	Block(s=1)	Block(s=1)	Block(s=1)
3	Block(s=1)	Block(s=1)	Block(s=1)
4	Block(s=1)	Block(s=1)	Block(s=1)
5	Block(s=2)	Block(s=1)	Block(s=1)
6	Block(s=1)	Block(s=1)	Block(s=1)
7	Block(s=1)	Block(s=1)	Block(s=1)
8	Block(s=2)	Block(s=1)	Block(s=1)
9	Block(s=1)	Block(s=1)	Block(s=1)
10	Block(s=1)	Block(s=1)	Block(s=1)
11	GAP	GAP	Koopman pooling

Table 1. **The architecture of CTR-GCN, CTR-GCN without stride and CTR-GCN with Koopman pooling.** Block(s=x) denotes the TCN-GCN block proposed in [2], with a stride of x for temporal convolution. GAP denotes global average pooling. As shown, the stride operations of CTR-GCN at the 5-th and 8-th layers are removed for Koopman model.

A.4. Eigenvalue Normalization Details

We investigate the influence of the rank of learned class-wise dynamics matrix $\mathbf{K}_i$ on the classification accuracy. Suppose $\lambda_1, \lambda_2, \ldots, \lambda_C \in \mathbb{C}$ are the eigenvalues of $\mathbf{K}_i \in \mathbb{R}^{C\times C}$ and $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_C|$. We set the rank of $\mathbf{K}_i$ to be r by retaining $\lambda_1, \ldots, \lambda_r$ and setting $\lambda_{r+1}, \ldots, \lambda_C$ to be zero. Then the reconstructed $\mathbf{K}_i$ is used for classification. In other words, only the leading eigenvalues and their corresponding eigenvectors are considered. The result is in Figure 2. As shown, using only one leading eigenvalue and

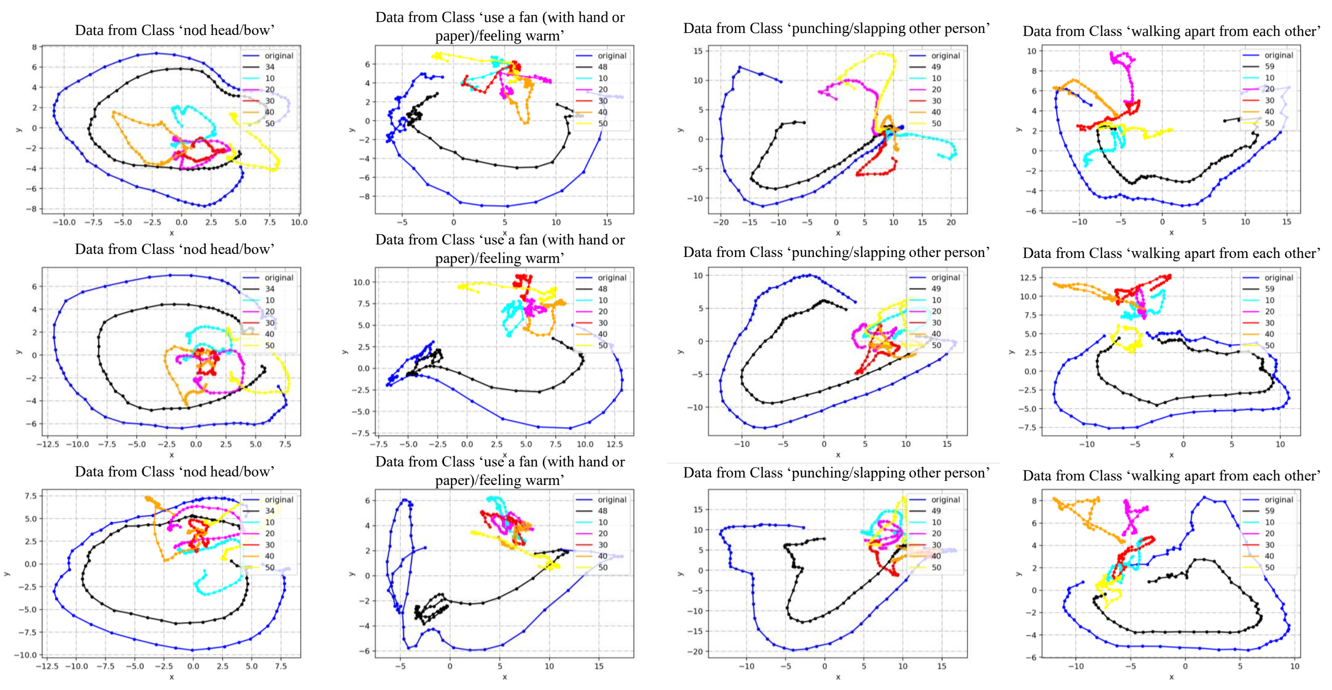

Figure 1. **Visualization of original trajectories and their 1-step linear evolution by the class-wise dynamics matrices $\hat{\mathbf{K}}_i$.** The ground-truth class name is annotated above each figure as 'Data from Class x' and the corresponding trajectory is plotted in black. Best viewed in color.

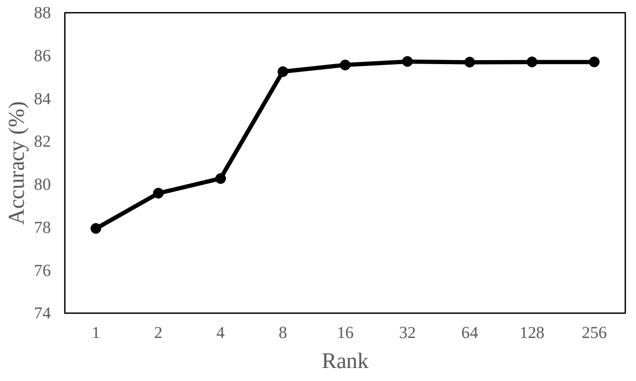

Figure 2. **The rank of dynamics matrix $\mathbf{K}_i$ and the corresponding accuracy.** The accuracy saturates when the rank empirically reaches 32.

eigenvector can achieve fairly good accuracy (78%), while the accuracy of using the full matrix is 85.7%. When the rank is 32, the accuracy reaches its maximum and ceases to rise. This indicates that the learned linear dynamics is low-rank in nature. Meaningful dynamics information lies in leading eigenvalues, while smaller eigenvalues are supposed to contain mostly noise. From the above observation, when conducting the proposed eigenvalue normalization, we only transform the top 32 leading eigenvalues and set others to zeros to reduce the noise.

B. Additional Visualization

More visualization of the trajectories in the linear Koopman space is presented in Figure 1. Trajectories that are evolved by the dynamic matrices of the ground-truth class (shown in black as in Figure 1) show similar patterns with the original ones (shown in the blue color), while the trajectories evolved by the dynamic matrices of other classes seem random. Also, the trajectories that belong to the same class exhibit similar patterns.

C. Additional Ablation Studies

We conduct additional experiments to show the influences of different p and T. The experiments are conducted on NTU120 cross-subject split and the results are presented in Tab. 2. Note that the maximum of T (*i.e.*, length of feature sequence) equals 64 as the length of the original action sequence is 64.

Table 2. Performance of different p and T on NTU120 X-sub.

p	0.1	0.25	0.5	0.75	**T**	64	32	16
joint(%)	85.4	85.7	85.8	85.7	**joint(%)**	85.6	85.3	83.4
bone (%)	86.9	87.2	87.1	87.0	**bone (%)**	87.1	86.9	86.5

D. Additional Literature Review

Apart from recent deep learning based method introduced in the main text, the idea of exploiting temporal dynamics information of human action has been adopted in many earlier classical works. The main idea is to extract the spatio-temporal dynamics of the sequence and conduct matching in databases. Many works focus on space-time volumes. For instance, some researchers extract spatio-temporal XT-slices from the image volume XYT and associate the reconstructed 2D trajectory with human motion. Others use motion history images, actions sketches to summarize the space-time volume. There are also some works which focus on silhouettes sequences and analyze them in eigenspace. Another line of works focus on geometric relationship between various body parts, and design hand-crafted features to represent the action sequence. These works mostly rely on physical and kinetic attributes, such as angle, distance and velocity.

Our proposed method is similar to the above classical methods in a sense, as we also focus on spatio-temporal dynamics of the action sequences and conduct dynamics matching. However, instead of directly extracting dynamics features, we first map the original sequences to a linear dynamical space, leveraging the power of deep neural nets. The linearity allows further operations such as dynamics matching, eigenvalues manipulation, etc.

References

[1] 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. 1

[2] 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

[3] 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. 1