BlockGCN: Redefine Topology Awareness for Skeleton-Based Action Recognition

Supplementary Material

A. Supplementary Material Structure

This supplementary material provides additional technical explanations and experimental validations to support and expand upon the main text of our work. The contents are organized as follows:

1. Detailed elaboration of the dynamic topological encoding scheme, Sec. B.
   (a) Definition and illustration of essential terms and concepts, Sec. B.1.
   (b) Theoretical foundation and methodology of persistent homology analysis for graph-structured data, Sec. B.2.
   (c) Comprehensive explanation of the adopted vectorization representation strategy, Sec. B.3.
2. In-depth discussion of the hyperparameter settings and optimization of BlockGCN, Sec. C.
3. Extended experimental validations and analysis, Sec. D.
   (a) Evaluation and comparison of single modality performance, Sec. D.1.
   (b) Investigation of the impact of different graph distance metrics on model performance, Sec. D.2.
   (c) Visual exploration and interpretation of the learned feature representations, Sec. D.4.

B. Technical Preliminaries

B.1. Fundamentals of Algebraic Topology

Topological data analysis (TDA) [54] leverages algebraic topology tools, such as persistent homology [22], to extract topological features, including connected components and cycles, from graph data that persist across multiple scales [2]. These topological descriptors have been shown to be effective representations for graph classification tasks [52, 75]. Furthermore, integrating these topological features with deep learning architectures has achieved significant success in enhancing the representational power of the models [18, 31, 47, 69, 72, 75]. In this section, we first introduce the core notations and concepts, followed by a general description of persistent homology analysis for graph data, and finally present a toy demonstration for intuitive understanding. For more detailed descriptions and formal illustrations of these techniques, we refer the reader to the corresponding literature in computational topology and topological data analysis [9, 23, 28].

Simplicial Complex: A simplicial complex is composed of simplices of different dimensions, such as vertices (0-simplices), edges (1-simplices), triangles (2-simplices), and tetrahedra (3-simplices). Given a k-simplex denoted as $\sigma = [v_0, ..., v_k]$, deleting one of its vertices v_i results in a $(k-1)$-simplex $[v_0, \dots, \hat{v}_i, \dots, v_k]$ ($\hat{v}_i$ denotes the deleted vertex), which is called the i-th *face* of σ. A simplicial complex $\mathcal{K}$ is defined as a set of simplices of varying dimensions that satisfies the following conditions:

1. Any face τ of a simplex $\sigma \in \mathcal{K}$ is also in $\mathcal{K}$ (i.e., $\tau \in \mathcal{K}$).
2. If $\sigma_1, \sigma_2 \in \mathcal{K}$ and $\sigma_1 \cap \sigma_2 \neq \emptyset$, then $\sigma_1 \cap \sigma_2$ is a face of both σ_1 and σ_2.

A graph $\mathcal{G}$ is a simplicial complex $\mathcal{K}$ consisting only of vertices (0-simplices) and edges (1-simplices).

Boundary Map: Given a simplicial complex $\mathcal{K}$, consider the vector space $C_\kappa(\mathcal{K})$ generated with $\mathbb{Z}2$ (the field with two elements). The boundary map is denoted as $\partial\kappa : C_\kappa(\mathcal{K}) \to C_{\kappa-1}(\mathcal{K})$. For a k-simplex $\sigma = [v_0, \dots, v_k) \in \mathcal{K}]$, the boundary map is defined as:

$$\partial_\kappa(\sigma) := \sum_{i=0}^{k} (v_0, \dots, v_{i-1}, v_{i+1}, \dots, v_k) \tag{7}$$

In other words, each vertex v_i of the simplex is omitted once. The boundary operator ∂ is a homomorphism between the simplicial chain groups, providing a precise way to define connectivity [31].

Homology: Homology theory employs commutative algebra tools to study topological features, such as connected components ($\kappa = 0$) and cycles ($\kappa = 1$) in a graph [23], using the boundary operator. The κ-th homology group $\mathbb{H}\kappa(\mathcal{K})$ of a simplicial complex $\mathcal{K}$ is defined as the quotient group:

$$\mathbb{H}\kappa(\mathcal{K}) := \mathbf{ker}\partial_\kappa / \mathbf{im}\partial_{\kappa+1} \tag{8}$$

The elements in $\mathbf{ker}(\partial_\kappa)$ and $\mathbf{im}(\partial_{\kappa+1})$ are called *κ-cycles* and *κ-boundaries*, respectively. The resulting homology groups $\mathbb{H}_\kappa(\mathcal{K})$ are topological invariants that remain unchanged under homeomorphisms and encode intrinsic information [28].

Betti Numbers: Betti numbers, defined as the ranks of the homology groups, serve as simpler invariants for classifying topological spaces. For $\mathbb{H}_\kappa(\mathcal{K})$, the 0-th Betti number $\beta_0 = \mathbf{rank}\mathbb{H}_0(\mathcal{K})$ represents the number of connected components, while the 1-st Betti number $\beta_1 = \mathbf{rank}\mathbb{H}_1(\mathcal{K})$ represents the number of cycles when $\kappa = 0$ and $\kappa = 1$, respectively. However, these counting-based topological summaries are too coarse to capture the complexity of graph structures. To address this limitation, a persistent version of homology-based topological invariant analysis is proposed, as described in the following section.

B.2. Persistent Homology Analysis for Graphs

In this subsection, we provide an overview of the persistent homology analysis for graphs, followed by an intuitive demonstration using a 5-node graph example. We then introduce the key notations and concepts for further reference.
Intuitive Demonstration: Consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a vertex set $\mathcal{V}$ and an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Given a threshold value ϵ, we can obtain a series of graphs by setting the edge weights $w_{ij}^{(\epsilon)}$ to 1 if $w_{ij}^{(\epsilon)} > \epsilon$, and 0 otherwise. Treating the graph $\mathcal{G}$ as a simplicial complex $\mathcal{K}$, we generate a sequence of simplicial complexes, termed as a *filtration*, $\{\mathcal{K}^i\}_{i=0}^m$, where $\emptyset = \mathcal{K}^0 \subseteq \mathcal{K}^1 \subseteq \ldots \subseteq \mathcal{K}^m = \mathcal{K}$, by increasing the threshold value ϵ. As the filtration parameter increases, more edges are removed from the graph. In extreme cases, when $\epsilon \to -\infty$, the graph becomes complete, and when $\epsilon \to \infty$, the graph reduces to a vertex set $\mathcal{V}$. For each sub-complex, we record the topological invariants, such as connected components and cycles, to describe the graph structure. During this filtration process, each topological object (i.e., homology) may appear at a specific ϵ_i and disappear at another value ϵ_j. The interval $\{\epsilon_i, \epsilon_j\}$ is called its *persistence*. *Persistent homology* analysis captures the global structure of graphs by recording these paired filtration values in the nested sequence. *Persistence barcodes* and *persistence diagrams* are used to represent the paired set $\{(b_i^{(0)}, d_i^{(0)})\}_{i=1}^n$, where $\mathcal{D}_i^{(0)} = (b_i^{(0)}, d_i^{(0)})$ and $b_i^{(0)}, d_i^{(0)} \in \{\epsilon_0, \epsilon_1, \ldots, \epsilon_k\}$ for connected components, and superscripts equal to 1 for cycles.

Figure 7 presents an intuitive demonstration of a 5-node graph filtration with threshold values $\epsilon = 0, 1, \ldots, 9$. As ϵ increases from 0 to 9, edges gradually appear, forming different combinations of connected components and cycles. For example, when ϵ increases from 0 to 1, the number of connected components decreases from 5 to 4 as one edge emerges. When ϵ increases from 2 to 3, a cycle appears and persists until $\epsilon = 9$. Through this counting and recording process, the geometrical structure of a weighted graph is explored globally.
Persistent Homology: Given a filtration of $\mathcal{K}$ denoted as $\{\mathcal{K}_i\}_{i=0}^m$, we have a corresponding sequence of chain complexes $C_\kappa(\mathcal{K}^i)$. The concept of homology groups is extended from $\mathbb{H}_\kappa^i(\mathcal{K}) := \mathbf{ker}\partial_\kappa^i / \mathbf{im}\partial_{\kappa+1}^i$ (dependent on a single simplicial complex $\mathcal{K}^i$) to its persistent version (from $\mathcal{K}^i$ to $\mathcal{K}^j$) as:

$$\mathbb{H}_\kappa^{i,j}(\mathcal{K}) := \mathbf{ker}\partial_\kappa^i / (\mathbf{im}\partial_{\kappa+1}^j \cap \mathbf{ker}\partial_\kappa^i) \tag{9}$$

The ranks of all the homology groups $\beta_\kappa^{i,j} = \mathbb{H}_\kappa^{i,j}(\mathcal{K})$ (namely the κ-th persistent Betti numbers) capture the number of homological features of dimensionality κ (e.g., connected components for $\kappa = 0$, cycles for $\kappa = 1$, etc.) that persist from i to (at least) j [29].
Persistence Barcodes of Filtration: For simplification,

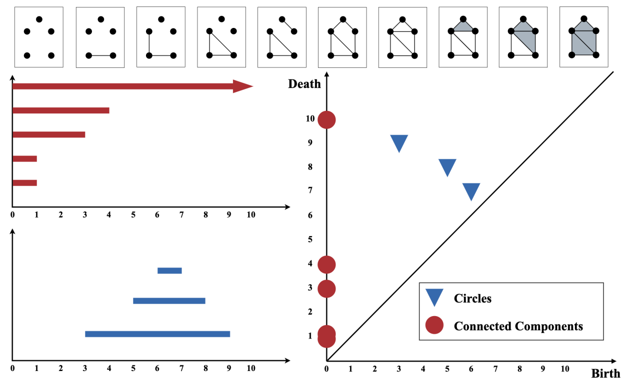


Figure 7. A graph filtration with $\epsilon = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$ (from left to right): (a) the persistence barcodes of connected components (up) and cycles (down); (b) corresponding persistent diagram of connected components (red disk) and cycles (blue triangle).[Best viewed in zoom and color]

we use $\mathbb{R}^2$ of $\{\mathcal{D}_1^{(0)}, \mathcal{D}_2^{(0)}, \ldots, \mathcal{D}_p^{(0)}\}$, where $\mathcal{D}_i^{(0)} = \{(b_i^{(0)}, d_i^{(0)})\}$, to denote the barcodes extracted from $\mathcal{K}$. Formally, the filtration sequence of $\mathcal{K}$ can be defined using a vertex filter function $f : \mathbb{V} \to \mathbb{R}$ with the filtration values $\epsilon_1 < \epsilon_2 \cdots \epsilon_m$, where $\epsilon_i \in \{f(v) : \{v\} \in \mathcal{K}\}$. With function f, the filtration of $\mathcal{K}$ is:

$$\mathcal{K}^{f,0} = \emptyset, \quad \mathcal{K}^{f,i} = \{\sigma \in \mathcal{K} : \max_{v \in \sigma} f(v) \leq \epsilon_i\} \tag{10}$$

for $1 \leq i \leq m$. Then, for the filtration of $\mathcal{K}$ and homology dimension κ ($\kappa = 0, 1$ in this work), we obtain the persistence barcode representation $\{\mathcal{D}_i^{(0)}\}_{i=1}^m = \{(b_i^{(0)}, d_i^{(0)})\}_{i=1}^m$, which we denote as $\mathcal{B}$.

B.3. Vectorization Representation

The inconsistency of using persistence barcodes $\{(b_i^{(0)}, d_i^{(0)})\}_{i=1}^m$ in machine learning tasks has led to the development of various vectorization approaches, including statistical analysis [4, 46], kernel methods [6, 11, 36, 38, 51], distance metrics [15, 45], and $\mathbb{R}^d$ elements [1, 3, 5, 8, 33].

Recently, learning-based techniques have been proposed to facilitate the integration of such graph descriptions into modern deep learning architectures by introducing learnable weights for each barcode [29, 31]. Typical embedding functions include the *rational hat* function [29], point transformation-based techniques [7], and the *DeepSets* approach [71] adopted in [31].

For computational efficiency and ease of implementation, we employ the *rational hat* function, as described in [29], for vectorization extraction due to its differentiability and expressive power in representing graphs. Mathematically, the *barcode coordinate function* maps a barcode in $\mathcal{B}$ to a real value by aggregating the points in the persistence

Table 9. Default Hyperparameters for BlockGCN on NTU RGB+D, NTU RGB+D 120, and Northwestern-UCLA.

Config.	NTU RGB+D 60 and 120	NW-UCLA
random choose	False	True
random rotation	True	False
window size	64	52
weight decay	4e-4	3e-4
base lr	0.05	0.05
lr decay rate	0.1	0.1
lr decay epoch	110, 120	90 100
warm up epoch	5	5
batch size	64	16
num. epochs	140	120
optimizer	Nesterov Accelerated Gradient	Nesterov Accelerated Gradient

Table 10. Classification Accuracy (%) of BlockGCN using Different Modalities on NTU RGB+D, NTU RGB+D 120, and Northwestern-UCLA Dataset.

Modality	NTU-RGB+D		NTU-RGB+D 120		NW-UCLA
	X-Sub	X-View	X-Sub	X-Set	
Joint	90.9	95.4	86.9	88.2	95.5
Bone	91.3	95.3	88.1	89.3	93.3
Motion	88.7	93.3	82.7	84.6	92.9
Bone Motion	88.3	92.6	83.0	84.8	88.8
Ensembled	93.1	97.0	90.3	91.5	96.9

diagram via a weighted sum:

$$\Psi : \mathbb{B} \to \mathbb{R} \quad \mathcal{B} \to \sum_{(b,d)\in\mathcal{B}} s(b, d) \tag{11}$$

where $s : \mathbb{R}^2 \to \mathbb{R}$ is a differentiable function that vanishes on the diagonal of $\mathbb{R}^2$. The rational hat structure element from [30] is defined as:

$$p \in \mathcal{B} \quad p \to \frac{1}{1 + \|p - c\|_1} - \frac{1}{1 + ||r| - \|p - c\|_1|} \tag{12}$$

where $c \in \mathbb{R}^2$ and $r \in \mathbb{R}$ are learnable parameters. This function evaluates the ”centrality” of each point $p \in \mathbb{B}$ with respect to a learned center c and a learned shift/radius r.

In our implementation, we adopt the modified version of the *rational hat* function provided in the *Pytorch-topological*[2] library, which is based on the original implementation by [29]. This vectorization approach allows us to transform the persistence barcodes into fixed-length feature vectors that can be readily integrated with deep learning models, such as the BlockGCN architecture used in our work. By learning the parameters of the *rational hat* function, we can adaptively capture the most informative topological features for the given graph classification task, enhancing the expressive power and discriminative capability of our model.

[2] https://pypi.org/project/torch-topological/

C. Hyperparameter Settings

In this section, we provide the default hyperparameter settings used for training our BlockGCN model on the NTU RGB+D, NTU RGB+D 120, and Northwestern-UCLA datasets. Throughout our experiments, we consistently train a 10-layer BlockGCN with a maximum channel dimension of 256. Table 9 presents the default hyperparameters for our BlockGCN model on these datasets. These hyperparameter settings have been carefully tuned to achieve optimal performance on each dataset while maintaining a balance between model complexity and computational efficiency. By using consistent hyperparameter settings across all experiments, we ensure a fair comparison and evaluation of our BlockGCN model's performance on different datasets and modalities.

D. Extended Experimental Results

In this section, we present additional experimental results to provide a more comprehensive evaluation of our BlockGCN model's performance on various datasets and modalities.

D.1. Single Modality Performance

To gain further insights into the contribution of each modality to the overall performance of our BlockGCN model, we conduct experiments training the model on each single modality separately. Table 10 provides detailed results of our BlockGCN's performance on each modality for the different benchmark datasets. These results demonstrate the effectiveness of our BlockGCN model in learning discriminative features from individual modalities, such as skeleton, RGB, depth, and infrared data. By examining the performance on each modality, we can identify the strengths and weaknesses of our model in capturing modality-specific information and guide future research efforts towards improving the fusion of multi-modal features. The single modality performance also serves as a baseline for evaluating the benefit of multi-modal fusion in our BlockGCN model. By comparing the results of single modality training with those of multi-modal fusion, we can quantify the synergistic effect of combining complementary information from different modalities to enhance the overall recognition accuracy.

D.2. Selection of Graph Distance for Static Topological Encoding

In the main text, we discuss the use of relative distances between joint pairs on the graph to symbolize graph topology. Theoretically, any proper graph distance can serve this purpose. In our work, we investigate two common graph distances for our Static Topological Encoding: the shortest path distance and the distance in the level structure [19]. Table 11 compares these two distances. Interestingly, both distances lead to an equivalent improvement, suggesting that they fundamentally convey the same information, i.e., bone connectivity. To streamline our approach, we default to employing the shortest path distance.

The choice of graph distance for Static Topological Encoding is an important consideration, as it directly influences the model's ability to capture the intrinsic topology of the skeleton graph. By comparing the performance of different graph distances, we can identify the most informative and computationally efficient representation for encoding the graph topology. The equivalent improvement observed when using either the shortest path distance or the distance in the level structure indicates that both distances effectively capture the essential connectivity information of the skeleton graph. This finding simplifies the implementation of our Static Topological Encoding, as we can focus on using the shortest path distance without compromising the model's performance.

Table 11. Comparing different graph distances for our Static Topological Encoding.

Graph Distance		Acc(%)
shortest path distance	level difference	
-	-	86.7
✓	-	86.9
-	✓	86.9

D.3. Choice of Simplicial Complex

In addition to the graph distance, we also explore the choice of simplicial complex for persistent homology analysis used in our dynamic topological encoding. Table 12 shows the comparison between two commonly used simplicial complexes: the Vietoris-Rips Complex and the Cubical Complex. The results indicate that using the Cubical Complex leads to a slight decrease of 0.2% in accuracy and significantly longer run time compared to the Vietoris-Rips Complex. Based on these findings, we adopt the Vietoris-Rips Complex for our dynamic topological encoding.

The choice of simplicial complex is crucial for efficient and effective persistent homology analysis. The Vietoris-Rips Complex, which is based on pairwise distances between points, provides a good balance between topological expressiveness and computational efficiency. On the other hand, the Cubical Complex, which is based on a cubical grid, may introduce additional computational overhead without providing significant benefits in terms of accuracy. By selecting the Vietoris-Rips Complex for our dynamic topological encoding, we ensure that our model can efficiently capture the evolving topological features of the skeleton graph over time, while maintaining high recognition accuracy.

Table 12. Comparing different simplicial complices.

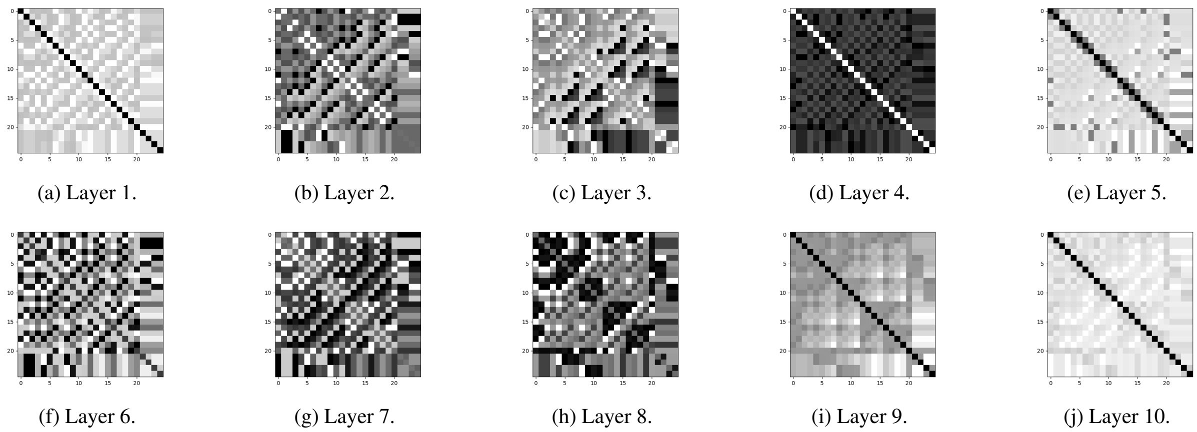
Vietoris–Rips Complex	Cubical Complex	Acc(%)
✓	-	86.9
-	✓	86.7

D.4. Visualization of Learned Representations

To gain further insights into the learned representations of our BlockGCN model, we provide additional visualizations of the Static Topological Encodings and the learned adjacency matrices.

Figure 8 presents more examples of the learned Static Topological Encodings, showcasing the model's ability to capture the intrinsic topology of the skeleton graph. These visualizations illustrate how our model learns to encode the relative distances between joint pairs, effectively representing the connectivity information of the skeleton.

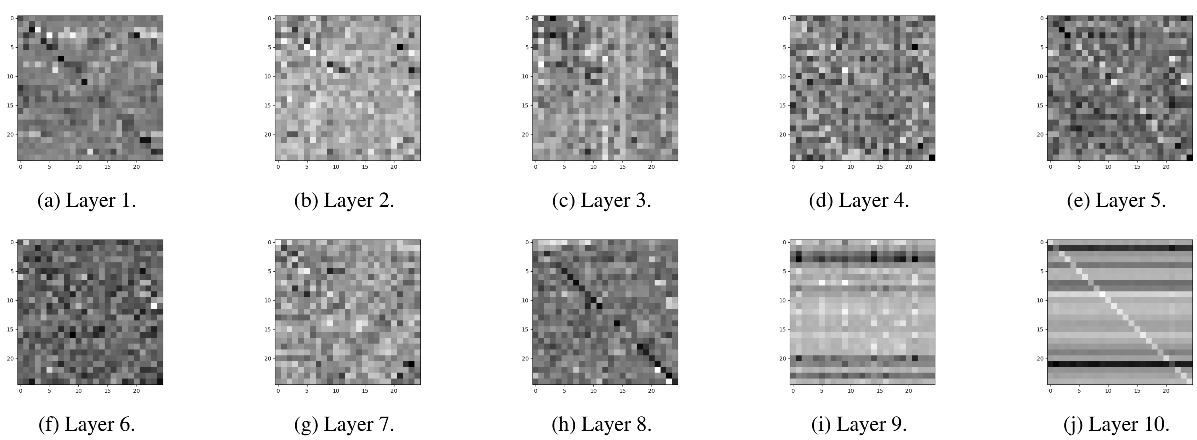
Figure 9 visualizes the learned adjacency matrices of our BlockGCN model. These matrices represent the learned graph structure and the strength of connections between different joints. By examining these visualizations, we can gain insights into how our model adapts the graph structure to better capture the dependencies and relationships between joints for action recognition. The visualizations of the learned Static Topological Encodings and adjacency matrices provide a qualitative assessment of our BlockGCN model's learning process.



(a) Layer 1. (b) Layer 2. (c) Layer 3. (d) Layer 4. (e) Layer 5.

(f) Layer 6. (g) Layer 7. (h) Layer 8. (i) Layer 9. (j) Layer 10.

Figure 8. The learned Static Topological Encodings of our BlockGCN at each layer. It can be seen that the learned weights are diverse and adapted to different levels of semantics.



(a) Layer 1. (b) Layer 2. (c) Layer 3. (d) Layer 4. (e) Layer 5.

(f) Layer 6. (g) Layer 7. (h) Layer 8. (i) Layer 9. (j) Layer 10.

Figure 9. The learned adjacency matrices of the GCN baseline model at each layer (Darker colors stand for larger weights). It can be seen that the learned weights vary dramatically among different layers and deviate far from the bone connections, which are used for initialization.

References

[1] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18, 2017. 2

[2] Mehmet E Aktas, Esra Akbas, and Ahmed El Fatmaoui. Persistence homology of networks: methods and applications. *Applied Network Science*, 4(1):1–28, 2019. 5, 1

[3] Nieves Atienza, Rocío González-Díaz, and Manuel Soriano-Trigueros. On the stability of persistent entropy and new summary functions for topological data analysis. *Pattern Recognition*, 107:107509, 2020. 2

[4] Eric Berry, Yen-Chi Chen, Jessi Cisewski-Kehe, and Brittany Terese Fasy. Functional summaries of persistence diagrams. *Journal of Applied and Computational Topology*, 4 (2):211–262, 2020. 2

[5] Peter Bubenik et al. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16(1):77–102, 2015. 2

[6] Mathieu Carriere, Marco Cuturi, and Steve Oudot. Sliced wasserstein kernel for persistence diagrams. In *International Conference on Machine Learning*, pages 664–673. PMLR, 2017. 2

[7] Mathieu Carrière, Frédéric Chazal, Yuichi Ike, Théo Lacombe, Martin Royer, and Yuhei Umeda. Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In *International Conference on Artificial Intelligence and Statistics*, pages 2786–2796. PMLR, 2020. 2

[8] Kit C Chan, Umar Islambekov, Alexey Luchinsky, and Rebecca Sanders. A computationally efficient framework for vector representation of persistence diagrams. *The Journal of Machine Learning Research*, 23(1):12281–12313, 2022. 2

[9] Frédéric Chazal and Bertrand Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Frontiers in Artificial Intelligence*, 4, 2021. 1

[10] 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. 2, 3, 4, 6, 7

[11] Yen-Chi Chen, Daren Wang, Alessandro Rinaldo, and Larry Wasserman. Statistical analysis of persistence intensity functions. *arXiv preprint arXiv:1510.02502*, 2015. 2

[12] 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*, pages 1113–1122, 2021. 7

[13] 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, 2020. 2, 3, 4, 6, 7, 8

[14] 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. 2, 3, 4, 6, 7, 8

[15] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. In *Proceedings of the Twenty-first Annual Symposium on Computational Geometry*, pages 263–271, 2005. 2

[16] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019. 2

[17] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in Neural Information Processing systems*, 29, 2016. 1

[18] Andac Demir, Baris Coskunuzer, Yulia Gel, Ignacio Segovia-Dominguez, Yuzhou Chen, and Bulent Kiziltan. Todd: Topological compound fingerprinting in computer-aided drug discovery. *Advances in Neural Information Processing Systems*, 35:27978–27993, 2022. 1

[19] Josep Díaz, Jordi Petit, and Maria Serna. A survey of graph layout problems. *ACM Computing Surveys (CSUR)*, 34(3): 313–356, 2002. 4

[20] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1110–1118, 2015. 1, 2

[21] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2969–2978, 2022. 7

[22] Herbert Edelsbrunner. Persistent homology: theory and practice. 2013. 5, 1

[23] Herbert Edelsbrunner and John L Harer. *Computational topology: an introduction*. American Mathematical Society, 2022. 1

[24] Lingling Gao, Yanli Ji, Yang Yang, and HengTao Shen. Global-local cross-view fisher discrimination for view-invariant action recognition. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5255–5264, 2022. 2, 3

[25] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, pages 1263–1272, 2017. 2

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015. 7

[27] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020. 2

[28] Felix Hensel, Michael Moor, and Bastian Rieck. A survey of topological machine learning methods. *Frontiers in artificial intelligence*, 4:681108, 2021. 1

[29] Christoph Hofer, Florian Graf, Bastian Rieck, Marc Niethammer, and Roland Kwitt. Graph filtration learning. In *International Conference on Machine Learning*, pages 4314–4323. PMLR, 2020. 5, 2, 3

[30] Christoph D Hofer, Roland Kwitt, and Marc Niethammer. Learning representations of persistence barcodes. *Journal of Machine Learning Research*, 20(126):1–45, 2019. 3

[31] Max Horn, Edward De Brouwer, Michael Moor, Yves Moreau, Bastian Rieck, and Karsten Borgwardt. Topological graph neural networks. In *International Conference on Learning Representations*, 2021. 1, 2

[32] Xiaohu Huang, Hao Zhou, Bin Feng, Xinggang Wang, Wenyu Liu, Jian Wang, Haocheng Feng, Junyu Han, Errui Ding, and Jingdong Wang. Graph contrastive learning for skeleton-based action recognition. *arXiv preprint arXiv:2301.10900*, 2023. 7

[33] Umar Islambekov and Hasani Pathirana. Vector summaries of persistence diagrams for permutation-based hypothesis testing. *arXiv preprint arXiv:2306.06257*, 2023. 2

[34] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. A new representation of skeleton sequences for 3d action recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4570–4579, 2017. 1, 2

[35] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2016. 1, 3

[36] Genki Kusano, Yasuaki Hiraoka, and Kenji Fukumizu. Persistence weighted gaussian kernel for topological data analysis. In *International Conference on Machine Learning*, pages 2004–2013. PMLR, 2016. 2

[37] Jungho Lee, Minhyeok Lee, Dogyoon Lee, and Sangyoun Lee. Hierarchically decomposed graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10444–10453, 2023. 7

[38] Chunyuan Li, Maks Ovsjanikov, and Frederic Chazal. Persistence-based structural recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1995–2002, 2014. 2

[39] Hao Li, Jinfa Huang, Peng Jin, Guoli Song, Qi Wu, and Jie Chen. Toward 3d spatial reasoning for human-like text-based visual question answering. *arXiv preprint arXiv:2209.10326*, 2022. 2

[40] Hao Li, Jinfa Huang, Peng Jin, Guoli Song, Qi Wu, and Jie Chen. Weakly-supervised 3d spatial reasoning for text-based visual question answering. *IEEE Transactions on Image Processing*, 2023. 2

[41] 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

[42] Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68(68):346–362, 2017. 1, 2

[43] 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. 2, 3, 6, 7

[44] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2

[45] Yuriy Mileyko, Sayan Mukherjee, and John Harer. Probability measures on the space of persistence diagrams. *Inverse Problems*, 27(12):124007, 2011. 2

[46] Chul Moon and Nicole A Lazar. Hypothesis testing for shapes using vectorized persistence diagrams. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 72(3): 628–648, 2023. 2

[47] Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. Topological autoencoders. In *International Conference on Machine Learning*, pages 7045–7054. PMLR, 2020. 1

[48] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. 4

[49] Wei Peng, Xiaopeng Hong, Haoyu Chen, and Guoying Zhao. Learning graph convolutional network for skeleton-based human action recognition by neural searching. In *Proceedings of the AAAI conference on artificial intelligence*, pages 2669–2676, 2020. 2

[50] Katharina Prasse, Steffen Jung, Yuxuan Zhou, and Margret Keuper. Local spherical harmonics improve skeleton-based hand action recognition. *arXiv preprint arXiv:2308.10557*, 2023. 2

[51] Jan Reininghaus, Stefan Huber, Ulrich Bauer, and Roland Kwitt. A stable multi-scale kernel for topological machine learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4741–4748, 2015. 2

[52] Bastian Rieck, Christian Bock, and Karsten Borgwardt. A persistent weisfeiler-lehman procedure for graph classification. In *International Conference on Machine Learning*, pages 5448–5458. PMLR, 2019. 5, 1

[53] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*, pages 593–607. Springer, 2018. 1

[54] Lee M Seversky, Shelby Davis, and Matthew Berger. On time-series topological data analysis: New data and opportunities. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition Workshops*, pages 59–67, 2016. 1

[55] 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

[56] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018. 2

[57] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12026–12035, 2019. 2, 3

[58] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017. 1, 2

[59] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. Constructing stronger and faster baselines for skeleton-based action recognition. *arXiv preprint arXiv:2106.15125*, 2021. 2, 3, 7

[60] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021. 4

[61] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. Resmlp: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*, 2021. 4

[62] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 3

[63] 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

[64] Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and improving relative position encoding for vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10033–10041, 2021. 2

[65] Hailun Xia and Xinkai Gao. Multi-scale mixed dense graph convolution network for skeleton-based action recognition. *IEEE Access*, 9:36475–36484, 2021. 2, 3

[66] Wangmeng Xiang, Chao Li, Yuxuan Zhou, Biao Wang, and Lei Zhang. Language supervised training for skeleton-based action recognition. *arXiv preprint arXiv:2208.05318*, 2022. 7

[67] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks. In *International Conference on Learning Representations*, 2018. 2

[68] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, pages 7444–7452, 2018. 2, 3

[69] Zuoyu Yan, Tengfei Ma, Liangcai Gao, Zhi Tang, Yusu Wang, and Chao Chen. Neural approximation of graph topological features. *Advances in Neural Information Processing Systems*, 35:33357–33370, 2022. 1

[70] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34: 28877–28888, 2021. 2, 3

[71] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in Neural Information Processing Systems*, 30, 2017. 2

[72] Sebastian Zeng, Florian Graf, Christoph Hofer, and Roland Kwitt. Topological attention for time series forecasting. *Advances in Neural Information Processing Systems*, 34: 24871–24882, 2021. 1

[73] 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. 1, 2

[74] 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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1112–1121, 2020. 2

[75] Qi Zhao and Yusu Wang. Learning metrics for persistence-based summaries and applications for graph classification. *Advances in Neural Information Processing Systems*, 32, 2019. 5, 1

[76] Huanyu Zhou, Qingjie Liu, and Yunhong Wang. Learning discriminative representations for skeleton based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10608–10617, 2023. 7

[77] Yuxuan Zhou, Zhi-Qi Cheng, Chao Li, Yanwen Fang, Yifeng Geng, Xuansong Xie, and Margret Keuper. Hypergraph transformer for skeleton-based action recognition. *arXiv preprint arXiv:2211.09590*, 2022. 3

[78] Yuxuan Zhou, Wangmeng Xiang, Chao Li, Biao Wang, Xihan Wei, Lei Zhang, Margret Keuper, and Xiansheng Hua. Sp-vit: Learning 2d spatial priors for vision transformers. *arXiv preprint arXiv:2206.07662*, 2022. 2