2025 IEEE/SICE International Symposium on System Integration (SII)
January 21-24, 2025, Munich, Germany

Multifingered Object Recognition with Tactile Sensors and Graph Convolutional Networks using Topological Graph Segmentation

Shardul Kulkarni, Satoshi Funabashi, Alexander Schmitz, Tetsuya Ogata, and Shigeki Sugano

Abstract— **This study investigates the application of topological segmentation to Graph Convolutional Networks (GCNs) for object recognition using tactile data from a multi-fingered robotic hand. While GCNs have shown promise in processing tactile information, the large volume of data from distributed tactile sensors poses challenges. Inspired by neurological research indicating intra-digit segmentation in human hand topology, we propose two methods of topological segmentation for GCNs: segmenting by digits and palm, and segmenting by individual skin patches. We evaluate these methods against a non-segmented GCN baseline using various input modalities including tactile features, taxel positions, and joint angles. Data was collected from an Allegro Hand equipped with uSkin tactile sensors, manipulating eight everyday objects. Our results demonstrate that topological segmentation enhances object recognition performance, with the best model achieving a 92.92% recognition rate using patch-level segmentation and tactile features with joint angles as input. UMAP analysis of GCN features reveals that segmentation methods produce distinct representations for each hand segment. Additionally, topological segmentation significantly reduces computational resource requirements compared to non-segmented GCNs. This study contributes the first application of topological segmentation to GCNs for tactile processing in robotic hands, achieving high object recognition rates and providing insights into feature extraction capabilities. The proposed method shows potential for improving efficiency and performance in tactile-based robotic manipulation tasks.**

I. INTRODUCTION

Achieving complex object manipulation poses a significant challenge when employing robots in both industrial and non-industrial settings. In an uncontrolled setting, the complexities arising from the diversity of objects with varying properties further exacerbate the difficulty of the problem. A multifingered robotic hand can enhance a robot's ability to manipulate complex objects due to its high dexterity. However, performing dexterous tasks presents a significant challenge in itself. Tactile sensors can provide rich data of touch states to enhance the performance of such tasks [1]. With tactile data, it is possible to detect intrinsic and extrinsic properties of objects and the objects themselves, improving the performance of dexterous tasks. However, processing abundant tactile information is demanding.

Some research has been conducted on processing tactile information in a spatial context [2][3][4]. In our previous works, we used Convolutional Neural Networks (CNNs) and Graph Convolutional Neural Networks (GCNs) for the spatial processing of tactile data from a multifingered hand to achieve tasks such as dexterous manipulation, object recognition, and object property recognition [5][6][7]. While CNNs are effective in the spatial processing of grid-like structured data, the geometry of a multifingered hand does not adhere to grid-like arrangement. The non-Euclidean nature of the graph feature space and its inherent permutation invariance [8][9], along with the construction of the sensor map using an adjacency matrix, the sensor map in GCNs more closely resembles the actual layout of the sensors on the hand. Hence, in the case of a multifingered hand with distributed tactile sensors, GCNs are more effective due to their superior ability to process topological information. We achieved promising results of manipulation and object property detection with GCNs. We could also confirm that GCNs can produce a topological representation of the geometry of the multifingered hand. However, with a graph of 384 distributed tactile sensors and 1401 edges, the volume of information represented by the graph is substantial.

Although a topological representation of the body is

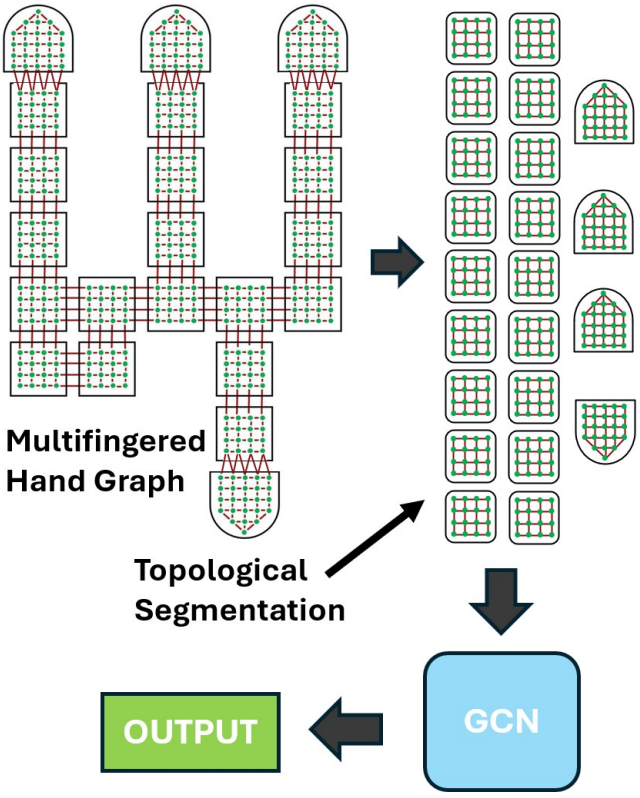

Fig. 1. Proposed method for topological graph segmentation in GCNs.

This research was supported by the Japan Science and Technology Agency, ACT-I Information and Future Acceleration Phase with a grant number of JPMJPR18UP and Moonshot R&D with a grant number of JPMJMS2031.

The authors are with Waseda University, Okubo 3-4-1, Shinjuku, Tokyo 169-8555, Japan. (e-mail: shardul@fuji.waseda.jp)

produced in the human brain [10], wherein the human hand with its digits is also represented, research also indicates that there is intra-digit segmentation of the hand's topology [11]. Hence, this study explores whether the complexities added to the graph by the large volume of data could be mitigated by topological segmentation of the multifingered hand, while enhancing the feature extraction capability and performance. We further present an analysis of the features produced by GCNs following topological segmentation.

Hence, this study explores whether topological segmentation of the multifingered hand can mitigate the complexities added to the graph by the large volume of data while enhancing feature extarction capabilities and performance. We further present an analysis of the features produced by GCNs following topological segmentation.

We trained GCNs without topological segmentation and GCNs with two topological segmentation methods for the task of object recognition. This task was chosen because it would provide valuable insight into the models' pattern extraction abilities in spatial feature processing. However, we believe the proposed method can be used in other applications, such as motion generation, and will be evaluated for such tasks in the future. The inputs to the GCN models were touch states, taxel coordinates, and joint angles of the robot hand. With the combination of these inputs and topological segmentation methods, we trained a total of 18 models to analyze the effect of topological segmentation along with the modality of input data. We make the following contributions to the field with this paper:

- The first attempt to apply topological segmentation to a multifingered hand and distributed tactile sensors to GCNs.
- Achieved high object recognition rates with the proposed methods.
- Analysis and comparison of GCN features with and without the segmentation methods.

II. Related Works

A. Graph Neural Networks

Neural networks utilizing graph structures excel in geometry-related tasks, such as identification of molecular fingerprints [12], and due to their strong topological feature extraction capabilities, they are finding applications in robotics [13]. Research on using graph neural networks for robot path planning [14], active information acquisition [15], and robot manipulation [16], among other applications, is ongoing. As far as topological segmentation of graphs is concerned, to the author's best knowledge, there is very limited research in this area. [17] introduces a topological segmentation method using mutual information, inspired by concepts from graph neural networks, as an efficient and scalable solution for map creation.

B. Tactile Spatial Processing with Tactile Sensors and Object Recognition

While tactile sensors provide robotic systems with the ability to acquire information about robot-object interaction from touch states, processing the abundant volume of information acquired from tactile sensors is challenging. Deep learning methods are effective in feature extraction from the large volume of tactile information. CNNs are effective in the spatial processing of tactile data [18][19], and hence can also be used for object recognition [20][21][5]. Methods other than CNNs are also used for processing tactile information. [22] recognizes objects through their texture properties. [23] implements spiking neural networks for object recognition using touch state data from a tactile sensor array. [24] achieved object recognition through a spiking graph neural network.

Although graph neural networks are effective in tactile feature extraction [25][26], there are limited works using graph neural networks in tactile feature extraction. [27][28] achieve grasp stability detection through graph neural networks and tactile sensing. [29] achieved live object grasping through graph-based reinforcement learning. Our previous work [6][7] involve in-hand manipulation and object property recognition with a multifingered hand and tactile sensors, respectively. [7] extracts spatial features by embedding geometric and topological data in graph edges. While [7] and several other works that used graph-based neural networks applied to tactile data focus on feature extraction and spatial processing, this work focuses on topological segmentation to simplify graph complexity for GCNs. Hence, the contribution of this work is the proposal and implementation of a method to make graph-structured data processing simpler and more efficient.

III. Proposed Method

A. Graph Convolutional Networks

We created a graph from distributed tactile sensors, where each taxel (tactile element) acts as a node. The sensor values in the X, Y, and Z directions, along with the sensor's X, Y, and Z coordinates in the Cartesian frame of the robot hand, serve as the node features. The output of the n^{th} GCN layer, $f(H^n, A)$, is given by:

$$f(H^n, A) = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^nW^n) \tag{1}$$

In this equation, H^n represents the input node features for the n^{th} Graph Convolutional layer, while A is the adjacency matrix that captures the graph structure of the uSkin sensors on the Allegro Hand. The output of this layer, H^{n+1}, serves as the input for the next $(n+1)^{th}$ Graph Convolutional layer. The matrix $\hat{A}$ is derived by adding the adjacency matrix A and an identity matrix I, allowing for the inclusion of both neighboring nodes and the target node in the adjacency. A symmetric normalization $\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^n$ is applied to counteract feature scale variations. W^n denotes the weight matrix of the n^{th} GCN layer, and σ is a nonlinear activation function.

B. Topological Segmentation

The graph constructed from the distributed tactile sensors on the robotic hand consists of 384 nodes, representing 384 distributed taxels, and 1401 edges, as shown in Fig. 2(a).

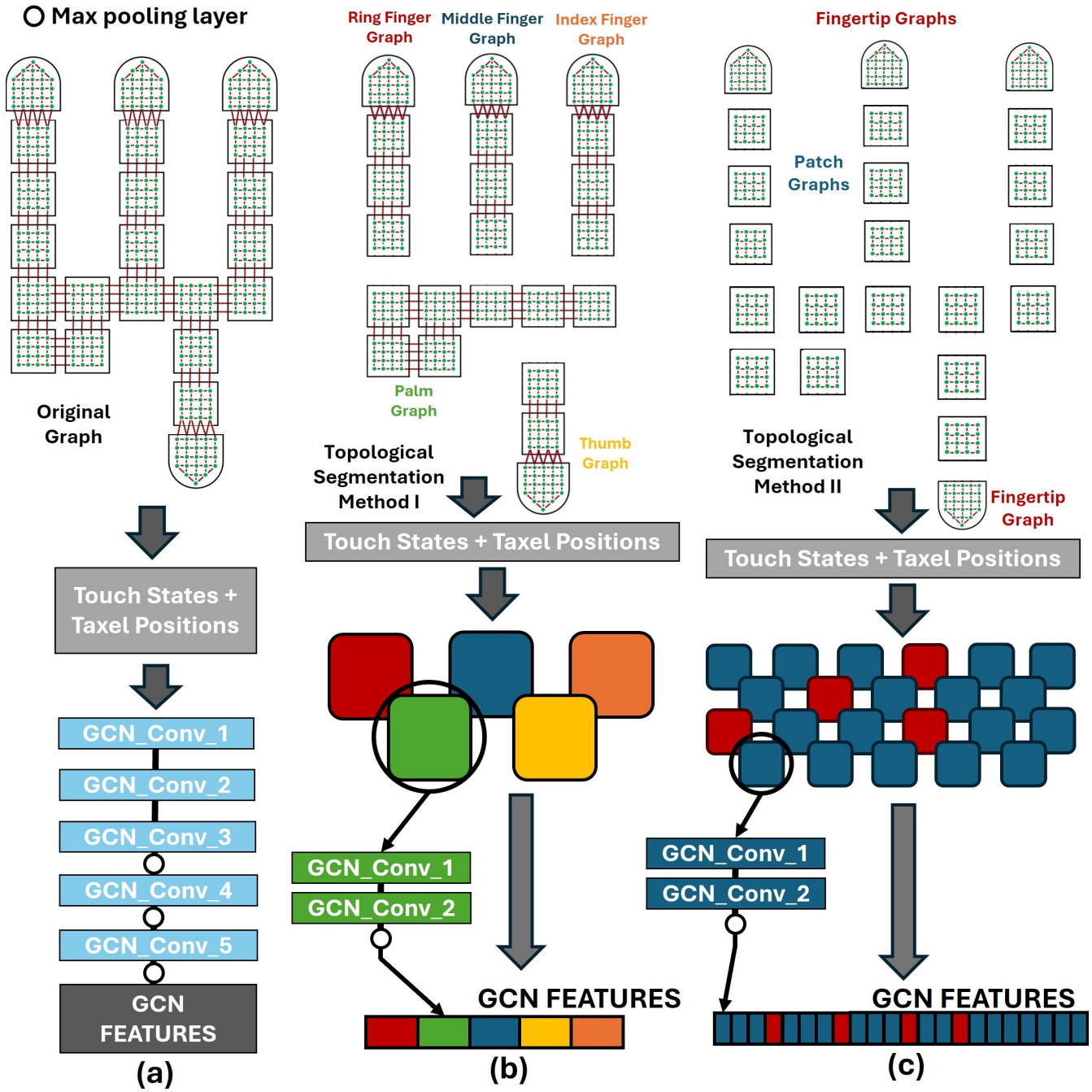

Fig. 2. Three different approaches to processing tactile data from a robotic hand using Graph Convolutional Networks (GCNs): (a) Shows the original full hand graph processed through a single, deep GCN with 5 convolutional layers. (b) Illustrates Topological Segmentation Method I, segmenting the hand into larger parts (fingers, thumb, palm). Each segment is processed by its own 2-layer GCN, with features concatenated afterwards. (c) Depicts Topological Segmentation Method II, where the hand is divided into many small patch and fingertip graphs. Each segment is processed by a shallow 2-layer GCN, with features from all segments combined at the end.

We implemented two methods to produce the topological segmentation. The segmentation of the graph was inspired by the finger somatotopy in the somatosensory cortex of the human brain, wherein different regions corresponding to the fingers and further corresponding to the phalanges and fingertips can be observed [30][11]. The graphs produced from each segment created using the methods described below, have their respective GCNs.

In Segmentation Method I, we produce the topological segmentation according to the digits and palm of the hand, as shown in Fig. 2(b). In this setting, we have five different graphs: four associated with the digits and one representing the palm.

In Segmentation Method II, we create individual graphs for each uSkin patch representing parts of the palm, phalanges, and fingertips, as shown in Fig. 2(c). Since there are a total of 22 skin patches on the hand, this method results in 22 small graphs, each performing spatial processing on the limited portion of the hand, of area if a skin patch.

Fig. 2 also shows the GCN network structure without

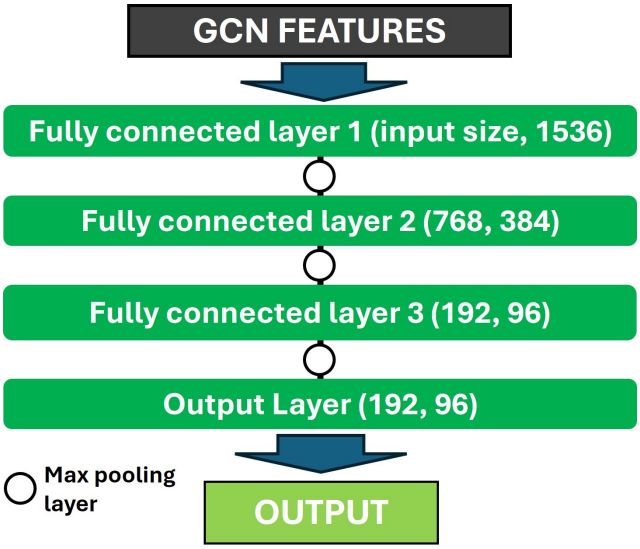

Fig. 3. FNN Layers of the network architecture. The input of the FNN layers is the output of GCN features and the output of the FNN layers is the predictor of object recognition.

segmentation and with the two topological segmentation methods described above. Without segmentation, we have a single thread of GCN with five layers. This architecture is the same as the one used in our previous work on object property recognition [7]. We used the same architecture because it is optimized for the recognition task.

For topological segmentation method I, we have five GCNs for Index Finger, Middle Finger, Ring Finger, Thumb and Palm respectively. The number of nodes and edges for the GCNs of the finger graphs are 72 and 258, respectively; for the thumb graph, they are 56 and 202, respectively; and for the palm graph, they are 112 and 392, respectively. From equation (1), the output of all the concatenated GCN features is expressed as :

$$H = (f(H^N_{index}, A_f), f(H^N_{middle}, A_f), f(H^N_{ring}, A_f), f(H^N_{thumb}, A_{th}), f(H^N_{palm}, A_{pa})) \quad (2)$$

Where Af, Ath, and Apa are the adjacency matrices of the finger graph, thumb graph, and palm graph, respectively.

For topological segmentation Method II, we have a GCN for each skin patch. The graph structure changes for the palm and phalanges patches, which are planar patches with 16 taxels in a 4x4 arrangement, and for the fingertip patch, which has 24 taxels distributed along the geometry of the fingertip. Hence, the patch-graphs consist of 16 nodes and 48 edges, and the fingertip-graphs consist of 24 nodes and 72 edges, respectively. From equation (1), the final output of all concatenated GCN layers can be expressed as :

$$H = (f(H^N_{p1}, A_p), f(H^N_{p2}, A_p), f(H^N_{p3}, A_p), f(H^N_{t1}, A_t), ..., f(H^N_{p22}, A_p)) \quad (3)$$

Where A_p and A_t are the adjacency matrices of the uSkin patch graph and the uSkin fingertip graph, respectively.

These output features are finally fed to the Feedforward Neural Network (FNN) layers, as shown in Fig. 3.

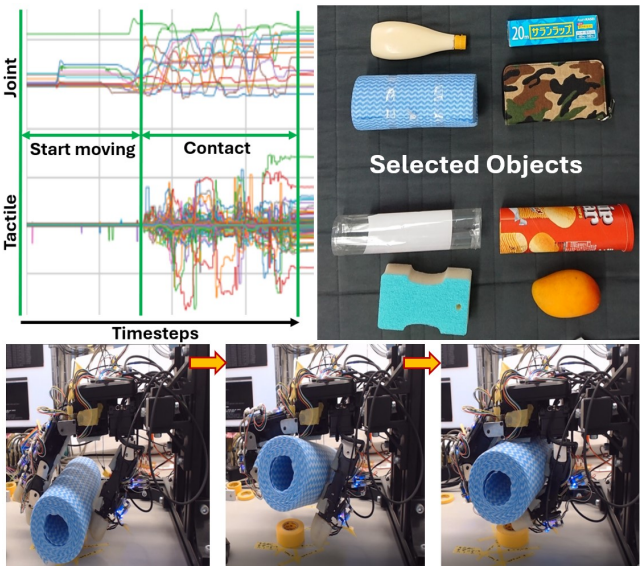

Fig. 4. Data collection setting. The top right image displays the objects used for training data. The top left image presents a plot of joint trajectories and tactile values over time. The bottom images illustrate an example of data collection involving kitchen paper.

IV. EXPERIMENTAL SETUP

A. Data Collection

This study employed the Allegro Hand, which features 16 degrees of freedom (DOF) across its four fingers. The device, created by Wonik Robotics, offers advanced dexterity suitable for various manipulative tasks. Following the approach of previous research [6], the fingertips, phalanges, and palms of the Allegro Hand were fitted with uSkin tactile sensors sold by Xela Robotics [31]. The uSkin is a magnetic tactile sensor consisting of magnets embedded in a soft silicone skin and Hall effect sensors positioned beneath the magnets. These Hall effect sensors track the movement of the magnets caused by the deformation of the silicone skin upon touch, measuring displacement along the X, Y, and Z axes by monitoring the magnetic fields. The uSkin operates at a data sampling frequency of 100 Hz. Overall, the Allegro Hand records 16 measurements from the four fingers' joint angles and 1,152 measurements from the tactile sensors distributed

TABLE I
NEURAL NETWORK SETTING

Inputs	Input types / sizes		
	X	I	II
Tac	(3 x 384)	F : (3 x 72) Th : (3 x 56) Pa : (3 x 112)	Patch : (3 x 16) Tip : (3 x 24)
Pos	(3 x 384)	F : (3 x 72) Th : (3 x 56) Pa : (3 x 112)	Patch : (3 x 16) Tip : (3 x 24)
Tac, Pos	(6 x 384)	F : (6 x 72) Th : (6 x 56) Pa : (6 x 112)	Patch : (6 x 16) Tip : (6 x 24)
GCN Layer Sizes	(14, 28, 56(P)28, 112(P)28, 112(P)7)	(14, 56(P)7)	(14, 56(P)7)
FNN Input Size	2688 + 16 Joints		
FNN Layer Sizes	(1536(P)768, 384(P)192, 96(P)48)		
Output	8 Object Labels		

TABLE II

COMPARISON OF RECOGNITION RATES WITH ALL INPUT COMBINATIONS AND SEGMENTATION METHODS

	No Joint			Joint		
	Tac	**Pos**	**Tac, Pos**	**Tac**	**Pos**	**Tac, Pos**
X	85.64 ± 0.28	75.03 ± 1.23	84.69 ± 0.34	88.66± 0.18	85.81 ± 0.02	**91.01 ± 0.38**
I	89.05 ± 0.04	78.63 ± 0.34	87.65 ± 0.07	**91.95 ± 0.08**	85.51 ± 0.1	**91.56 ± 0.22**
II	89.41 ± 0.1	80.16 ± 0.28	87.03 ± 0.49	**92.92 ± 0.09**	85.64 ± 0.28	**92.04 ± 0.06**

TABLE III

COMPARISON OF ROC VALUES WITH ALL INPUT COMBINATIONS AND SEGMENTATION METHODS

	No Joint			Joint		
	Tac	**Pos**	**Tac, Pos**	**Tac**	**Pos**	**Tac, Pos**
X	0.9189	0.8586	0.9132	0.9358	0.9195	**0.9491**
I	0.9382	0.879	0.9298	**0.9545**	0.9177	**0.9523**
II	0.9402	0.8874	0.9264	**0.9599**	0.9232	**0.9549**

across the fingertips, phalanges, and palms [(4 fingertips × 24 uSkin sensor chips) + (11 phalanges + 7 sensors on the palm) × 16 uSkin sensor chips] × 3 axes, which amounts to 1,168 measurements.

Objects were randomly positioned beneath the Allegro Hand, and data were collected while the hand picked up and held eight everyday objects. The hand's movements were remotely controlled using a CyberGlove (22-sensor model) from CyberGlove Systems, depicted in Fig. 4. This setup was designed to maximize contact between the hand and the objects, providing a rich and reliable tactile data. Each object underwent 10 successful trials, with each trial lasting approximately 17 seconds, totaling 80 trials across all objects. Prior to training the Graph Convolutional Network (GCN), the collected data underwent preprocessing. This dataset was then split into three parts: 60% for training the model, 10% for validation, and 30% for testing, ensuring a robust evaluation of the network's performance.

Following objects were chosen for data collection and evaluation : a mayonnaise tube, saran wrap, kitchen paper, a wallet, a plastic tube, a potato chip cylindrical cardboard case, a sponge, and a replica of a mango, as shown in Fig. 4. The labels are one-hot vectors of dimension 8 for the 8 objects, where one of the 8 elements of this one-hot vector is set to '1' depending on the object it represents, and the rest of the elements are '0'.

B. Neural Network Settings

Table I shows the settings of the neural networks used for the evaluation. The Roman numerals I and II refer to Topological Segmentation Method I and Method II, respectively, while X refers to the case where no segmentation is used. 'Tac' refers to the input tactile features, and 'Pos' refers to input taxel positions in the Cartesian space of the robotic hand. We trained models using the two mentioned modalities and their combinations, resulting in the following three types of input combinations:

- GCN trained with tactile features
- GCN trained with taxel positions
- GCN trained with tactile features plus taxel positions

All these models were trained with and without joint angle input from the Allegro Hand. We trained GCN with both segmentation methods and without segmentation for each input combination with tactile features, taxel positions, and joint angles. Thus, we trained a total of 18 models to evaluate the effect of topological segmentation along with input modalities. Each of these models was trained five times for 2001 epochs per iteration. The mean detection rate of these five iterations was calculated to account for variance.

Without segmentation, the GCN has five layers of the sizes shown in Table I. This model size has been hyperparameter-tuned from our previous work [7]. For the segments of the hand, we created smaller GCNs with only two layers for each segment in either of the methods. For Topological Segmentation Method I, there are five GCN models, each corresponding to a segment of the hand. For Topological Segmentation Method II, there are twenty-two GCN models, corresponding to the skin patches and fingertip patches on the hand. The individual GCN size was reduced for segmentation because the segment graphs are considerably smaller than the graph of the entire hand. Hence, using deeper GCNs with a large number of features can lead to saturation in GCNs, affecting the model's performance.

We used Max-Pooling layers for optimization, denoted by (P) in Table I. The value before (P) indicates the number of output features of the GCN before Max-Pooling, whereas the value after (P) is the number of features after Max-Pooling, which is also the input to the next GCN layer or fully connected (FNN) layers. Pooling layers are also used between the FNN layers for optimization, as indicated in Table I.

V. EVALUATION

A. Analysis of GCN Performance

Table II presents the average percentage recognition rates and percentage variance of recognition rates observed in case of each of the five iteration for each of the models. One immediate observation from this table is that the taxel position modality alone is quite insufficient for achieving decent results for the task. Although the performance of the GCNs with taxel positions improves with joint angle input, it still falls short compared to the models with tactile input, whether segmentation is used or not. Also, regardless of segmentation method, networks perform significantly better when joint angles are inputted than their counter-parts without joint angle.

When segmentation is not used, we can observe that the tactile and tactile + taxel position models with joint angles show decent performance. Without segmentation, the GCN requires all three inputs — tactile states, taxel positions, and joint angles — to produce the best result of 91.01%. The model accuracy decreases without joint angles when tactile and taxel position modalities are used, indicating that the taxel position modality requires joint angles to add context to feature extraction for the model. This tendency is observed consistently whether segmentation exists or not.

In every type of input combination, with the exception of taxel positions and joint angles, it is observed, that the models with either segmentation method applied outperform the models without segmentation. This indicates the effectiveness of topological segmentation for feature extraction with GCNs. An interesting observation is that, when joint angles are included, GCNs with tactile and joint angle input perform better than those with combined tactile and taxel positions with joint angle input, regardless of the segmentation method used. This suggests that while the segmentation method provides sufficient topological information about the hand, proprioceptive information from the joints is necessary to further enhance the model's performance.

Finally, the best model is the GCN with Topological Segmentation Method II and tactile features plus joint angle input, achieving a 92.92% recognition rate. This demonstrates not only the effectiveness of the proposed topological segmentation approach but also emphasizes that increased granularity in feature extraction enhances its efficiency.

Table III shows the average ROC values of the five iterations for each network. It can be observed that the ROC values follow the same trend as the recognition rates, with the GCN using Topological Segmentation Method II and tactile features plus joint angle input achieving the highest ROC value.

B. Analysis of GCN Features

Fig. 5 shows the UMAP analysis of the GCN features. Fig. 5(a) displays the features from the GCN without segmentation, Fig. 5(b) shows the features from the GCN with Segmentation Method I, and Fig. 5(c) presents the features from the GCN with Segmentation Method II. All models in these figures use tactile features and joint angles as input.

To create these plots, we decomposed the node features produced by the GCNs. The features indicate that when segmentation methods are applied, the GCNs generate independent representations for each segment. In Fig. 5(a), the GCN without segmentation reveals the hand's topology, with node features of each finger connected to the palm.

In Fig. 5(b), the GCN with Segmentation Method I shows features that correspond to each segment independently. Five distinct clusters are visible, representing the three fingers, thumb, and palm.

For Segmentation Method II (Fig. 5(c)), separate clusters for each uSkin patch are observed. This indicates that tactile information from each patch is processed independently, enhancing granularity in spatial feature extraction.

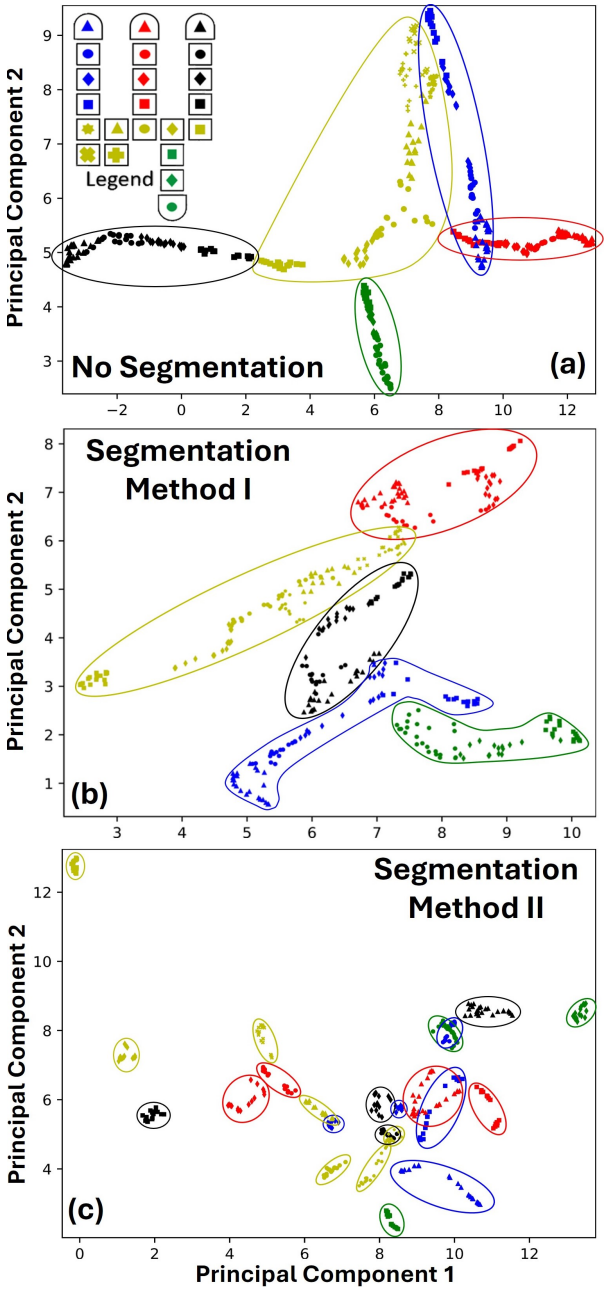

Fig. 5. Feature visualization of the GCN layers. (a) Without Segmentation: Basic hand topology. (b) Segmentation Method I: Clusters for hand segments. (c) Segmentation Method II: Detailed clusters for uSkin patches.

C. Analysis of GCN Computational Requirements

While training the model, we found that topological segmentation of graphs reduces the computational resource requirements. Table IV shows the GPU memory required by each type of segmentation for the same inputs. We recorded the GPU memory usage during training for 250 epochs, averaged these values, and rounded the average.

It is observed that the GPU memory required for models without topological segmentation is approximately double that of models using topological segmentation methods with the same input combination. Additionally, Segmentation Method II, which segments the hand into skin patches, further reduces GPU memory usage compared to Segmenta-

TABLE IV

COMPARISON OF GPU MEMORY USAGE FOR MODELS IN MEGA BYTES

	No Joint			Joint		
	Tac	**Pos**	**Tac, Pos**	**Tac**	**Pos**	**Tac, Pos**
X	1523	1534	1533	1531	1537	1488
I	757	752	747	752	743	748
II	689	692	697	697	696	701

tion Method I. Thus, the proposed topological segmentation method not only enhances performance, but also improves efficiency by lowering resource requirements.

VI. CONCLUSION

This study demonstrates the effectiveness of topological segmentation in enhancing Graph Convolutional Networks for object recognition using tactile data from a multi-fingered robotic hand. By segmenting the hand into its constituent parts, we achieved higher object recognition rates than non-segmented GCNs, with the best model reaching a 92.92% recognition rate. The UMAP analysis of GCN features revealed that segmentation enables more focused feature extraction from different parts of the hand. Additionally, the segmented models significantly reduced computational resource requirements, using about half the GPU memory compared to their non-segmented counterparts. These findings have important implications for robotic manipulation, which could lead to more effective and resource-efficient systems capable of complex tasks.

The low computational resource consumption of the proposed method makes it suitable for a real-time deployment. Furthermore, the ability of granular processing touch-states can be further exploited is potentially beneficial for achieving fine manipulation with a multi-fingered hand. Overall, future work involves exploring the application of this method for motion generation and in-hand manipulation using a multi-fingered hand with tactile sensors, extending its application to other tactile-based multi-fingered tasks, and investigating its scalability to more complex hand designs and diverse tactile sensors.

REFERENCES

[1] A. Yamaguchi and C. G. Atkeson, "Recent progress in tactile sensing and sensors for robotic manipulation: can we turn tactile sensing into vision?" *Advanced Robotics*, vol. 33, no. 14, pp. 661–673, 2019. [Online]. Available: https://doi.org/10.1080/01691864.2019.1632222

[2] V. Müller, T.-L. Lam, and N. Elkmann, "Sensor design and model-based tactile feature recognition," in *2017 IEEE SENSORS*, 2017, pp. 1–3.

[3] T. Sugaiwa, G. Fujii, H. Iwata, and S. Sugano, "A methodology for setting grasping force for picking up an object with unknown weight, friction, and stiffness," in *2010 10th IEEE-RAS International Conference on Humanoid Robots*, 2010, pp. 288–293.

[4] A. J. Spiers, M. V. Liarokapis, B. Calli, and A. M. Dollar, "Single-grasp object classification and feature extraction with simple robot hands and tactile sensors," *IEEE Transactions on Haptics*, vol. 9, no. 2, pp. 207–220, 2016.

[5] S. Funabashi, G. Yan, F. Hongyi, A. Schmitz, L. Jamone, T. Ogata, and S. Sugano, "Tactile transfer learning and object recognition with a multifingered hand using morphology specific convolutional neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2022.

[6] S. Funabashi, T. Isobe, F. Hongyi, A. Hiramoto, A. Schmitz, S. Sugano, and T. Ogata, "Multi-fingered in-hand manipulation with various object properties using graph convolutional networks and distributed tactile sensors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2102–2109, 2022.

[7] S. Kulkarni, S. Funabashi, A. Schmitz, T. Ogata, and S. Sugano, "Tactile object property recognition using geometrical graph edge features and multi-thread graph convolutional network," *IEEE Robotics and Automation Letters*, vol. 9, no. 4, pp. 3894–3901, 2024.

[8] N. A. Asif, Y. Sarker, R. K. Chakrabortty, M. J. Ryan, M. H. Ahamed, D. K. Saha, F. R. Badal, S. K. Das, M. F. Ali, S. I. Moyeen, M. R. Islam, and Z. Tasneem, "Graph neural network: A comprehensive review on non-euclidean space," *IEEE Access*, vol. 9, pp. 60 588–60 606, 2021.

[9] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[10] T. Orlov, T. R. Makin, and E. Zohary, "Topographic representation of the human body in the occipitotemporal cortex," *Neuron*, vol. 68, no. 3, pp. 586–600, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0896627310007749

[11] M. A. Schweisfurth, J. Frahm, and R. Schweizer, "Individual fmri maps of all phalanges and digit bases of all fingers in human primary somatosensory cortex," *Frontiers in Human Neuroscience*, vol. 8, 2014. [Online]. Available: https://www.frontiersin.org/journals/human-neuroscience/articles/10.3389/fnhum.2014.00658

[12] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," 2018. [Online]. Available: https://arxiv.org/abs/1812.08434

[13] F. Pistilli and G. Averta, "Graph learning in robotics: A survey," *IEEE Access*, vol. 11, pp. 112 664–112 681, 2023.

[14] Q. Li, F. Gama, A. Ribeiro, and A. Prorok, "Graph neural networks for decentralized multi-robot path planning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 11 785–11 792.

[15] M. Tzes, N. Bousias, E. Chatzipantazis, and G. J. Pappas, "Graph neural networks for multi-robot active information acquisition," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3497–3503.

[16] Y. Lin, A. S. Wang, E. Undersander, and A. Rai, "Efficient and interpretable robot manipulation with graph neural networks," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2740–2747, 2022.

[17] M. Liu, F. Colas, and R. Siegwart, "Regional topological segmentation based on mutual information graphs," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3269–3274.

[18] J. M. Gandarias, A. J. García-Cerezo, and J. M. Gómez-de Gabriel, "Cnn-based methods for object recognition with high-resolution tactile sensors," *IEEE Sensors Journal*, vol. 19, no. 16, pp. 6872–6882, 2019.

[19] M. Polic, I. Krajacic, N. F. Lepora, and M. Orsag, "Convolutional autoencoder for feature extraction in tactile sensing," *IEEE Robotics and Automation Letters*, vol. 4, pp. 3671–3678, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:199058386

[20] J. M. Gandarias, A. J. García-Cerezo, and J. M. G. de Gabriel, "Cnn-based methods for object recognition with high-resolution tactile sensors," *IEEE Sensors Journal*, vol. 19, pp. 6872–6882, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:149633731

[21] A. Karamipour and S. H. Sadati, "Tactile object recognition using fluid-type sensor and deep learning," *IEEE Sensors Letters*, vol. 7, no. 9, pp. 1–4, 2023.

[22] M. Kaboli, A. De La Rosa T, R. Walker, and G. Cheng, "In-hand object recognition via texture properties with robotic hands, artificial skin, and novel tactile descriptors," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 1155–1160.

[23] J. Kim, S.-P. Kim, J. Kim, H. Hwang, J. Kim, D. Park, and U. Jeong, "Object shape recognition using tactile sensor arrays by a spiking neu-

ral network with unsupervised learning," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 178–183.

[24] F. Gu, W. Sng, T. Taunyazov, and H. Soh, "Tactilesgnet: A spiking graph neural network for event-based tactile object recognition," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9876–9882.

[25] W. Fan, H. Bo, Y. Lin, Y. Xing, W. Liu, N. Lepora, and D. Zhang, "Graph neural networks for interpretable tactile sensing," in *2022 27th International Conference on Automation and Computing (ICAC)*, 2022, pp. 1–6.

[26] ——, "Graph neural networks for interpretable tactile sensing," in *2022 27th International Conference on Automation and Computing (ICAC)*, 2022, pp. 1–6.

[27] T. Mi, D. Que, S. Fang, Z. Zhou, C. Ye, C. Liu, Z. Yi, and X. Wu, "Tactile grasp stability classification based on graph convolutional networks," in *2021 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 2021, pp. 875–880.

[28] A. Garcia-Garcia, B. S. Zapata-Impata, S. Orts-Escolano, P. Gil, and J. Garcia-Rodriguez, "Tactilegcn: A graph convolutional network for predicting grasp stability with tactile sensors," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8.

[29] Z. Hu, Y. Zheng, and J. Pan, "Living object grasping using two-stage graph reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1950–1957, 2021.

[30] D. Janko, K. Thoenes, D. Park, W. R. Willoughby, M. Horton, and M. Bolding, "Somatotopic mapping of the fingers in the somatosensory cortex using functional magnetic resonance imaging: A review of literature," *Frontiers in Neuroanatomy*, vol. 16, 2022.

[31] T. P. Tomo, S. Somlor, A. Schmitz, L. Jamone, W. Huang, H. Kristanto, and S. Sugano, "Design and characterization of a three-axis hall effect-based soft skin sensor," *Sensors*, vol. 16, no. 4, 2016. [Online]. Available: https://www.mdpi.com/1424-8220/16/4/491