2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

Hypergraph Convolutional Networks Based Spatial Tactile Modeling for Object Geometric Property Recognition

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

Abstract— **This paper presents the application of Hypergraph Convolutional Networks (HGCNs) for tactile spatial processing in multifingered robotic hands. Building on prior work employing Graph Convolutional Networks (GCNs) for modeling irregular sensor layouts, we address the architectural complexity introduced by topological segmentation approaches through the use of hypergraphs, which naturally capture higher-order relationships among tactile sensors. We evaluate HGCNs, standard GCNs, and feedforward neural networks (FNNs) on object geometric property recognition using eight objects and multimodal input (touch states, taxel coordinates, and joint angles). Our results demonstrate that HGCNs achieve high recognition rates of 96.61% while reducing model redundancy, and that hyperedge structure and types of hypergraph adjacencies significantly influence model performance. These findings suggest HGCNs offer scalable and effective tactile data processing.**

I. INTRODUCTION

Robot object interaction is still an open problem, with a number of approaches being explored worldwide. With large amount of diversity in the objects and a large number of complex motions required for the interaction, the problem becomes very complex. The multifingered hand covered with tactile sensors offers added dexterity and sensory data for active perception and can enhance the robot-object interaction. The rich data from the touch states from the tactile sensors is useful for several tasks such as object recognition, object property recognition, tactile manipulation [1]. However, tactile data is often abundant in volume, and needs sophisticated methods for processing.

Recent studies have investigated various approaches for interpreting tactile signals, with a particular emphasis on how spatial features contribute to tactile perception [2][3][4]. In earlier research, we applied convolutional neural networks (CNNs) to analyze tactile data. However, the irregular configuration of sensors on a multifingered robotic hand complicates the direct use of convolutional filters, which are designed for grid-based data. Therefore we adopted graph convolutional neural networks (GCNs), which are better suited for processing data from non-uniform sensor layouts. By representing the tactile sensors and their connections as a graph, GCNs enable more accurate modeling of the hand's physical structure. This method has allowed us to successfully perform tasks such as dexterous manipulation,

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)

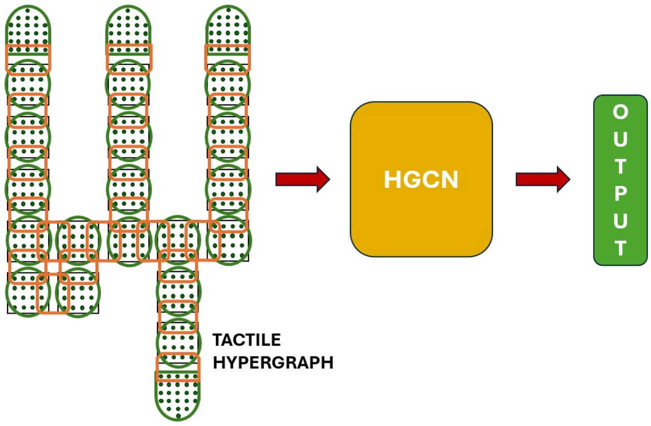

Fig. 1. HGCN - Tactile data from a multifingered robotic hand as a hypergraph of sensors connected by hyperedges. A Hypergraph Convolutional Network (HGCN) processes this to efficiently analyze tactile information.

object identification, and recognition of object properties using tactile feedback [5][6][7]. Our findings indicate that constructing adjacency matrices to reflect sensor relationships enhances the GCN's ability to capture the underlying topology of the hand. While this approach yields robust results in both manipulation and property recognition, it also introduces computational challenges due to the complexity of the sensor network, which in our case involved 384 sensors and 1401 connections.

Addressing this issue to make the tactile spatial processing more efficient, we implemented topological segmentation on the graph [8], and trained separate and independent GCN models for each of the graph segment. We used the task of object recognition to evaluate this method and observed high recognition rate. Yet, the model that performed the best, had its graph segmentation done with respect to every patch, and there are 22 skin patches on our multifingered hand setup. Hence we had to employ 22 small GCNs, one for every patch segment, making the architecture complex to maintain.

To overcome this architectural complexity, we explored an alternative representation using hypergraphs, a generalization of traditional graphs where edges, called hyperedges, can connect more than two nodes simultaneously. This property makes hypergraphs particularly suitable for modeling higher-order relationships and interactions among groups of tactile sensors. Unlike standard graphs that capture only pairwise connections, hypergraphs can represent complex, non-local dependencies in a more compact and semantically meaningful way. This enables a more holistic view of tactile data while potentially reducing the number of individual models or layers required. As a result, hypergraph-based methods offer the advantage of scalability and reduced architectural redundancy [9], making them a promising direction for representing tactile sensing data more efficiently in multifingered robotic hands.

We trained Hypergraph Convolutional Networks (HGCNs) and evaluated their performance against regular GCNs. The task chosen to evaluate the models was 'object geometric property recognition'. We also trained feedforward neural networks (FNN) as the baselines for data processing to measure against the spatial processing of the graph based networks. The inputs given to the models for training were touch states, taxel coordinates, and joint angles of the robot hand. With the combination of these inputs and types of networks, we trained a total of 9 models to evaluate the performance of the HGCNs. We further trained different HGCNs with different types of hypergraph adjacencies, which describe the relationship between a nodes and a hyperedge, and analyzed how different types of hypergraph adjacencies affect the performance of HGCNs. Data from 8 objects was collected using Allegro hand covered in tactile sensors. This data was used and models were trained for geometrical property recognition of these objects. We make the following contributions to the field with this paper:

- To the best of author's knowledge, the first work that applies HGCNs for tactile spatial processing.
- Achieved high object geometrical property recognition rates HGCNs.
- Analysis of the HGCN features and effect on HGCNs with different hyperedge structures.

II. Related Works

A. Hypergraph Convolutional Networks

Originally successful in domains like molecular structure analysis [10], graph-based neural networks are now gaining traction in robotics due to their capability to capture complex spatial and topological relationships [11]. Hypergraph Convolutional Networks [12][13][14] are an extension of traditional Graph Convolutional Networks designed to model higher-order relationships that naturally arise in complex data. Unlike standard graphs, where edges connect only pairs of nodes, hypergraphs allow each hyperedge to connect multiple nodes simultaneously, enabling the representation of group-wise interactions.

Recent advances have demonstrated the effectiveness of HGCNs in a variety of domains, including hyperspectral image classification, recommendation systems, and semantic segmentation [15]. For example, [12] proposed a multi-ordering HGCN for event co-reference resolution, showcasing the model's ability to handle complex cross-document relationships. Additionally, new frameworks such as DHM-Conv have introduced directed hypergraph convolution, enabling the modeling of directional high-order relationships and further improving representational power in tasks involving directional data [16]. Owing to the advantages presented by the Hypergraph structure, there are a few works exploring the use of Hypergraph based neural networks in various applications [17][18][19][20]. Yet there has been no attempt to explore the potential of HGCNs for tactile spatial processing.

B. Tactile Spatial Processing with Tactile Sensors and Object Recognition

Recent advances in tactile sensing have led to an increasing reliance on machine learning models for interpreting the high-dimensional data generated by sensor arrays. Among these, convolutional neural networks (CNNs) have proven particularly useful due to their ability to process local spatial patterns within tactile maps [21], [22]. Nonetheless, alternative methods have been explored to capture different aspects of tactile perception. For example, object properties such as texture can be leveraged for classification tasks, as demonstrated in [23]. Meanwhile, spiking neural networks [24] and spiking graph neural networks [25] offer biologically-inspired models that process tactile input through event-driven architectures. The range of techniques explored underscores the complexity of tactile data and the need for specialized models depending on the application context.

Recent advances have demonstrated that graph neural networks (GNNs) can effectively capture features from tactile sensor data [26][27], yet their broader use in tactile perception tasks is still emerging. While some research has integrated GNNs with tactile feedback to assess the stability of robotic grasps [28][29], and others have leveraged graph-based reinforcement learning for dynamic object manipulation by robotic hands [30], the application of Hypergraph Convolutional Networks (HGCNs) for tactile representation remains unexplored. Our prior studies [6][7][8] have addressed challenges such as in-hand manipulation, recognition of object properties, and object identification using multifingered robotic hands equipped with tactile sensors. In these works, we primarily focused on encoding spatial relationships via graph edge features and employing topological segmentation to enhance GCN efficiency. In contrast, the current method explores Hypergraph structures for tactile representation and utilizes HGCNs to simplify the complexities arising from topological segmentation while simultaneously achieving high recognition performance at high computational efficiency.

III. Proposed Method

A. Hypergraphs and Hypergraph Convolutional Networks

A graph consists of nodes, N and and edges connecting those nodes E. In a graph one edge connects one node as shown in Fig. 2. A hypergraph also consists of nodes, N, and instead of edges, there are hyperedges, E. One hyperedge connects several nodes at a time, and if a hyperedge was to connect only two nodes at a time, it would represent an edge of a regular graph. One can imagine a hypergraph to be a generalization of the regular graph. The output of the n^{th} HGCN layer, $f(X^n, H)$, is given by:

$$f\left(X^{(n)}, H\right) = \sigma\left(D_v^{-\frac{1}{2}} H W B_e^{-1} H^\top D_v^{-\frac{1}{2}} X^{(n)} \Theta^{(n)}\right) \quad (1)$$

In this equation, $X^{(n)}$ denotes the input node features for the n^{th} Hypergraph Convolutional layer, and H is the

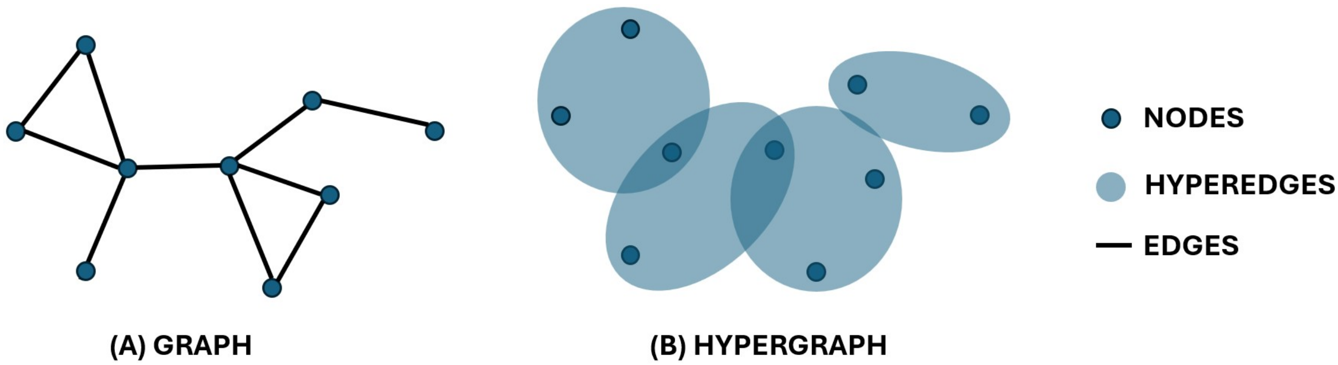

Fig. 2. Graphs vs Hypergraphs. Graphs represent relationships between pairs of objects, where edges connect individual nodes and each edge links exactly two vertices. Hypergraphs generalize this concept by allowing "hyperedges" to connect any number of nodes, capturing more complex, multi-way relationships within a single structure.

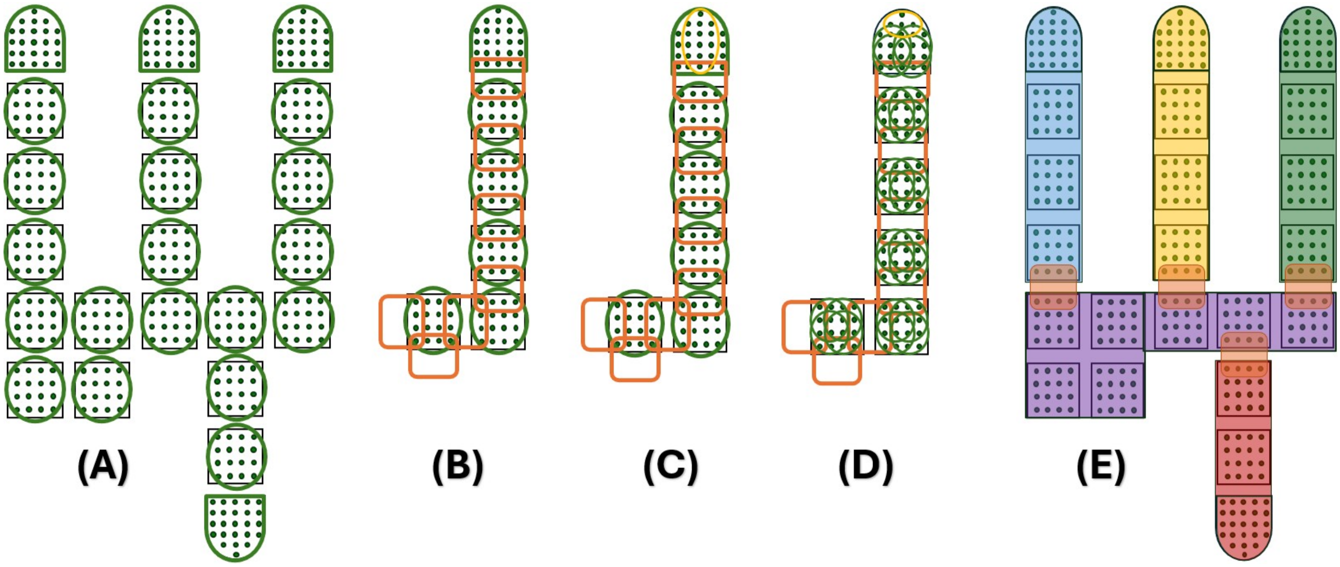

Fig. 3. Different hypergraph adjacencies. The hypergraph (A) makes hyper edges out of every patch, with adjacency becoming gradually more dense till (D). For (A) to (D), the green hyperedges represent intra-patch hyperedges, orange hyperedges represent inter-patch hyperedges and yellow hyper-edges represent intra-patch hyperedge on the fingertip. Hypergraph (E) is a sparse hypergraph with only 9 hyperedges, and it follows the topology of the hand, with every digit and palm being associated with a respective hyperedge represented by every colour, and orange hyperedges representing nodes around the joints of these parts.

hypergraph incidence matrix of the hypergraph. The incidence matrix of a hypergraph encodes the membership of each node in each hyperedge, thereby capturing higher-order relationships among the taxels. The matrix D_v is a diagonal matrix containing the degrees of the nodes, while B_e is a diagonal matrix containing the degrees of the hyperedges. W represents the hyperedge weight matrix, which can be used to assign different importances to each hyperedge. The term $H^\top$ is the transpose of the incidence matrix, enabling the aggregation of information across hyperedges. A symmetric normalization, $D_v^{-\frac{1}{2}} H W B_e^{-1} H^\top D_v^{-\frac{1}{2}} X^{(n)}$, is applied to ensure stable feature propagation and to counteract variations in node and hyperedge connectivity. $\Theta^{(n)}$ denotes the learnable weight matrix of the n^{th} HGCN layer, and σ is a nonlinear activation function. The output of this layer serves as the input for the subsequent $(n+1)^{th}$ Hypergraph Convolutional layer.

B. Hypergraph Representation of Tactile Sensors

The taxels of the Allegro hand serve as the nodes in the hypergraph, since there are 384 taxels on the Allegro hand, there are 384 nodes in the hypergraph. To capture higher-order relationships among these taxels, we introduce hyperedges, which connect groups of nodes that share a particular spatial and functional relationship, following the topology of the hand. The process of defining these hyperedges is a key design choice and can significantly influence the expressive power of the resulting hypergraph.

Fig. 3 shows various types of hypergraphs (A), (B), (C), (D) and (E) used for the assessment in this study. The consideration for constructing this matrix is how dense the hypergraph adjacency is, i.e. how many hyperedges are there in the hypergraph and how much overlap is present in the hyperedges. As can be observed in Fig. 3, the hypergraph (A) makes hyper edges out of every patch, and with every hypergraph, there are gradual and different types of overlaps in the hyper-edges till hypergraph (D) with hypergraph (D) being the densest hypergraph. Hypergraph (E) is a sparse hypergraph with only 9 hyperedges, and it follows the topology of the hand, with every digit and palm being associated with a respective hyperedge. In the evaluation section we have

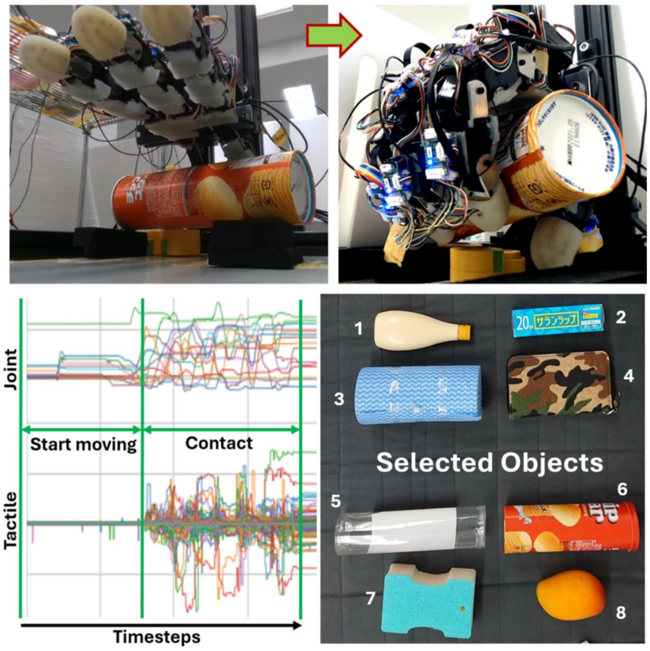

Fig. 4. Data collection setting. The top images show the moement of robot object interaction for data collection . The bottom left image presents a plot of joint trajectories and tactile values over time. The bottom right image shows the objects used for training data.

analyzed how these different hyperedge configurations affect the performance of the HGCN.

The output features from HGCNs and GCNs are further passed through feedforward layers and finally through the output layer. The activation function of the output layer is sigmoid function and mean squared error loss was used for loss calculation.

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 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: 80% for training the model and 20% for testing, ensuring a robust evaluation of the network's performance.

TABLE I
OBJECTS AND LABELS

Objects	Properties				
	cuboidal	curved	flat	non-flat	irregular
Mayonnaise Tube	0	1	0	1	1
Saran Wrap	1	0	0	1	0
Kitchen Paper	0	1	0	1	0
Wallet	1	0	1	0	0
Plastic Tube	0	1	0	1	0
Potato Chip Cylinder	0	1	0	1	0
Sponge	1	0	0	1	0
Mango Replica	0	1	0	1	1

The experimental trials incorporated eight selected objects — mayonnaise tube (1), saran wrap (2), kitchen paper (3), wallet (4), plastic tube (5), potato chip cylinder (6), sponge (7), and a mango replica (8), each shown in Fig. 4. Each object was labeled using a 5-dimensional binary vector representing the presence 1 or absence 0 of the following properties: cuboidal, curved, flat, non-flat, and irregular. For example, For example, the mango replica is labeled [0, 1, 0, 1, 1] for these properties in order. All the objects and their respective labels are shown in Table I.

B. Neural Network Settings

Table II shows the settings of the neural networks used for the evaluation. 'Tac' refers to the input tactile features, and 'Pos' refers to input taxel positions, which are the absolute coordinates of taxels in the Cartesian space of the robotic hand. We trained HGCN, GCN and FNN models using the two mentioned modalities and their combinations, resulting in the following three types of input combinations:

- HGCN, GCN and FNN trained with tactile features and joint angles
- HGCN, GCN and FNN trained with taxel positions and joint angles
- HGCN, GCN and FNN trained with tactile features plus taxel positions and joint angles

Thus, we trained a total of 9 models to evaluate the effect

TABLE II
NEURAL NETWORK SETTING

	Tac	Pos	Tac, Pos
Input Size	(3 x 384)	(3 x 384)	(6 x 384)
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	5 Geometric Object Property Labels		

of topological segmentation along with input modalities. Each of these models was trained three times for 1001 epochs per iteration. The mean detection rate of these three iterations was calculated to account for variance.

Both the HGCN and GCN architectures are composed of five layers, with their respective dimensions detailed in Table II. The chosen network sizes result from hyperparameter optimization conducted in our earlier research [7].

To enhance model efficiency, Max-Pooling layers were incorporated, as indicated by (P) in Table II. The number preceding (P) specifies the output feature count from the GCN layer before Max-Pooling, while the number following (P) represents the feature count after pooling, which then serves as input for the subsequent GCN or fully connected (FNN) layer. Max-Pooling is also applied between FNN layers for further optimization, as shown in the table. Fig. 5 shows the neural network setting for GCNs and HGCNs in this study.

For the feedforward neural networks (FNNs), we trained them using flattened tactile data combined with absolute morphological node attributes. Depending on the input configuration—tactile features alone, taxel positions alone, or a combination of tactile features and taxel positions—the FNN input dimensions are 1152, 1536, and 2688, respectively, in addition to 16 joint angle values. The sizes of the layers of FNNs are as follows: (32768(P)16384, 4096(P)2048, 1024(P)512). These substantial sizes were selected to ensure a fair comparison by matching the high neuron counts typically present in the HGCN and GCN layers.

V. Evaluation

A. *Effect of Different Hypergraph Adjacency Matrix*

TABLE III

Comparison of Recognition Rates with Different Hypergraph Adjacencies

	(A)	(B)	(C)	(D)	(E)
Detection Rate	96.61	96.28	95.99	94.88	92.11
Number of Hyperedges	22	44	48	109	9

Table III shows the detection rates of HGCN with the different configurations of hyperedges. Configuration (A) simply follows the skin patches and gradually upto configuration (D) there are overlaps in hyperedges, with configuration (D) having most number of overlaps and largest number of hyperedges. Configuration (E) on the other hand, is very sparse with hyperegdes following digits and pal of the and, and small overlap at the joints.

The results show, that larger overlap in hyperedges lead to reduced performance with patch-wise hyperedges delivering the best performance, and it gradually decreasing as the hyeredges become denser. Yet, huge hyperdegdes with large number of nodes are not helpful either, as it reduces the ability of granular spatial processing of HGCN.

B. *Recognition Performance*

Table IV presents the average percentage recognition rates and percentage variance of recognition rates observed in case of each of the three iteration for each of the models. All the other models are compared against the HGCN with adjacency martrix A, which produces the best performance among HGCNs with different adjacencies.

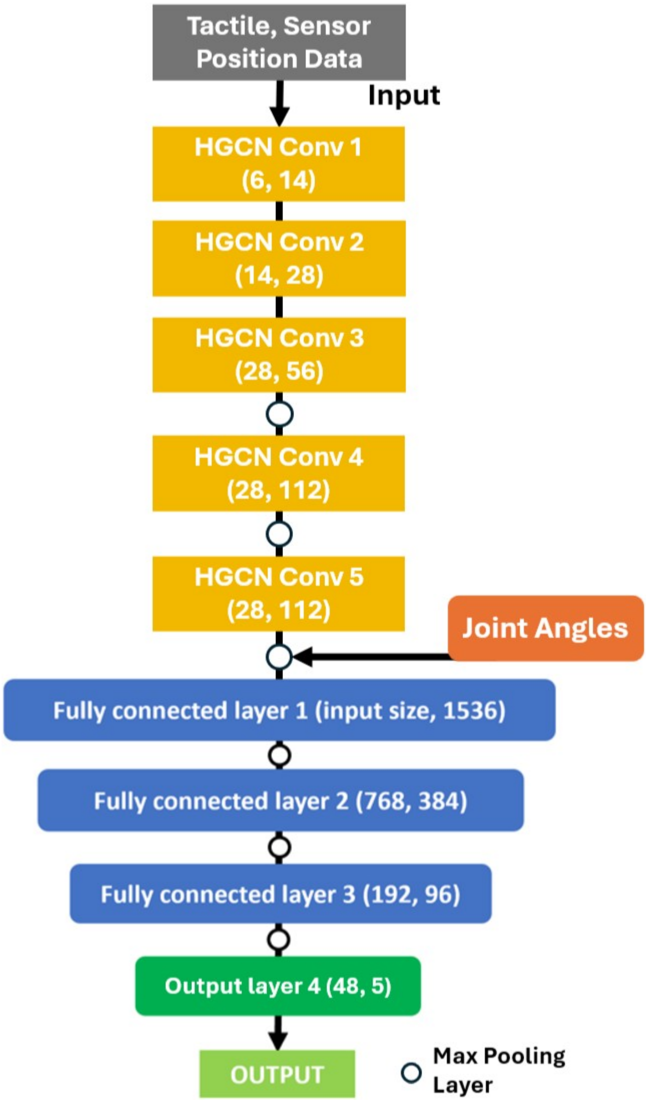

Fig. 5. Model The HGCN model consists of five HGCN layers and four fully connected layers. The white circles indicate pooling layer.

TABLE IV

Comparison of Recognition Rates

	Tac	Pos	Tac, Pos
HGCN	94.60	92.5	**96.61**
GCN	94.42	91.86	95.37
FNN	89.53	92.84	86.98

From Table IV, it can be seen that overall, FNNs produce the worst detection rate with the highest detection rate being 92.94% with taxel coordinate information.

With GCNs and HGCNs we see an opposite pattern of the detection rate against input combination, compared to FNNs. The worst detection rate is encountered with taxel position input, and the best detection rate with taxel position + tactile input combination. It can be seen that for every input combination, HGCN has superior performance to the GCN, and the best detection rate is 96.61% with the combination of tactile input and taxel position input, showing that HGCNs are better than GCNs at spatial processing.

C. Analysis of Computational Requirements of HGCN

TABLE V

COMPARISON OF GPU MEMORY USAGE FOR MODELS IN MEGA BYTES

	Tac	Pos	Tac, Pos
HGCN	732 (1.93)	727 (1.96)	719 (2.01)
GCN	1162 (2.59)	1148 (2.59)	1204 (2.69)
FNN	2670 (3.14)	2673 (3.13)	3388 (3.62)

To compare the computational efficiency of HGCN with other models, we measured GPU memory usage and training time per epoch across 100 training epochs. Table V presents the results: the values outside parentheses indicate average GPU usage in megabytes (MB), while the values in parentheses show the average time per epoch in seconds.

HGCN models consistently required 400–500 MB less GPU memory than their GCN counterparts, demonstrating their lower memory footprint. In contrast, FNNs consumed the most memory due to their large, fully connected layers with many neurons.

In terms of training speed, HGCNs also outperformed the others, requiring the least time per epoch. FNNs were the slowest, and GCNs had intermediate training times. These results clearly demonstrate that HGCNs are not only more memory-efficient but also faster to train than equivalent GCN and FNN models.

VI. CONCLUSION

This study demonstrates the effectiveness of Hypergraph Convolutional Networks (HGCNs) for tactile spatial processing in multifingered robotic hands. By leveraging hypergraphs to capture higher-order relationships among tactile sensors, our approach achieved superior object geometric property recognition rates compared to standard GCNs, while reducing architectural complexity. These results highlight the potential of hypergraph-based representations to enable more effective and scalable robotic manipulation systems. The proposed HGCN model achieved the best recognition rate of 96.61% and we also demonstrated how hypergraph-adjacency affects the HGCN performance. Future work will investigate applying HGCNs to a wider range of tactile tasks, real-time operation, and adaptation to different robotic hand configurations and sensor technologies.

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] K. Shardul, S. Funabashi, A. Schmitz, T. Ogata, and S. Sugano, "Multifingered object recognition with tactile sensors and graph convolutional networks using topological graph segmentation," in *2025 IEEE/SICE International Symposium on System Integration (SII)*, 2025, pp. 671–678.

[9] N. Yadati, M. Nimishakavi, P. Yadav, V. Nitin, A. Louis, and P. P. Talukdar, "Hypergcn: A new method for training graph convolutional networks on hypergraphs." in *NeurIPS*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 1509–1520. [Online]. Available: http://dblp.uni-trier.de/db/conf/nips/nips2019.htmlYadatiNYNLT19

[10] 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

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

[12] S. Bai, F. Zhang, and P. H. Torr, "Hypergraph convolution and hypergraph attention," *Pattern Recognition*, vol. 110, p. 107637, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320320304404

[13] S. Kim, S. Y. Lee, Y. Gao, A. Antelmi, M. Polato, and K. Shin, "A survey on hypergraph neural networks: An in-depth and step-by-step guide," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 6534–6544. [Online]. Available: https://doi.org/10.1145/3637528.3671457

[14] N. Yadati, M. Nimishakavi, P. Yadav, V. Nitin, A. Louis, and P. Talukdar, "Hypergcn: A new method for training graph convolutional networks on hypergraphs," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[15] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. [Online]. Available: https://doi.org/10.1609/aaai.v33i01.33013558

[16] W. Zhao, Z. Ma, and Z. Yang, "DHMConv: Directed hypergraph momentum convolution framework," in *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, S. Dasgupta, S. Mandt, and Y. Li, Eds., vol. 238. PMLR, 02–04 May 2024, pp. 3385–3393. [Online]. Available: https://proceedings.mlr.press/v238/zhao24c.html

[17] Y. Deng, R. Zhang, P. Xu, J. Ma, and Q. Gu, "Pre-trained hypergraph convolutional neural networks with self-supervised learning," *Transactions on Machine Learning Research*, 2024. [Online]. Available: https://openreview.net/forum?id=0VWXWPmctm

[18] H. Shi, Y. Zhang, Z. Zhang, N. Ma, X. Zhao, Y. Gao, and J. Sun, "Hypergraph-induced convolutional networks for visual classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 10, pp. 2963–2972, 2019.

[19] X. Tong, S. Li, J. Li, X. Liu, Y. Hou, S. Chen, and J. Yang, "Spatio-temporal hypergraph convolutional network based network traffic prediction," in *Proceedings of the International Conference on Computing, Machine Learning and Data Science*, ser. CMLDS '24.

New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3661725.3661727
[20] L. Wang and S. Deng, "Hypergraph convolution network classification for hyperspectral and lidar data," *Sensors*, vol. 25, no. 10, 2025. [Online]. Available: https://www.mdpi.com/1424-8220/25/10/3092
[21] 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.
[22] 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
[23] 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.
[24] 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 neural network with unsupervised learning," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 178–183.
[25] 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.
[26] F. Wen, 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.
[27] 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.
[28] 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.
[29] 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.
[30] 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.
[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