IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Tactile Object Property Recognition using Geometrical Graph Edge Features and Multi-Thread Graph Convolutional Network

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

***Abstract*— Performing dexterous tasks with a multi-fingered robotic hand remains challenging. Tactile sensors provide touch states and object features for multifingered tasks, yet the variety in shapes, sizes, textures, deformabilities and masses of everyday objects makes the task conditions diverse. Despite these challenges, humans accomplish these difficult tasks by producing a sensory-motor representation of their body. This combined tactile and proprioceptive representation enables humans to accommodate the diversity in daily objects. Referring to this concept, this paper presents a method for object property recognition using Graph Convolutional Networks (GCNs), leveraging robot hand proprioception and morphology with spatial embeddings derived from geometrical graph edge features acquired from real tactile sensor alignments on an Allegro Hand. Additionally, a Multi-Thread GCN (MT-GCN) architecture is introduced to process these edge features and basically multi-modal data in a graph. Training data was acquired using a data-glove, from tri-axial tactile sensors distributed across the fingertips, finger phalanges, and palm of an Allegro Hand, producing a total of 1152 tactile measurements. MT-GCN with proposed edge features, tactile features and joint angles achieved a high recognition rate, 86.08% for six classes of object property combinations from eight objects. The effect of variation in graph adjacency on MT-GCN was examined. The proposed network showed clusters following the robot hand configuration with t-SNE analysis. Furthermore, analysis of learned parameters in the edge-feature encoder demonstrated its ability to discern joint positions on the hand, acquiring proprioceptive features effectively. Consequently, we could confirm that the proposed method was effective for multi-fingered dexterous tasks.**

I. INTRODUCTION

The prospect of utilizing multi-fingered robotic hands poses numerous challenges. The sheer diversity in various dexterous task settings and motions associated with them makes the problem of generalization difficult. Tactile sensors are an effective solution to obtain touch states and the object features involved in various tasks [1]. However, even with such rich data of touch states, achieving dexterous multi-fingered tasks is further complicated because of the underlying complexities owing to irregularities in shapes, sizes, textures, deformabilities and mass of miscellaneous objects involved, which also makes the task-conditions diverse. Subsequently, the sensory data is diverse. Even deep learning methods struggle to handle and learn such non-linear data.

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)

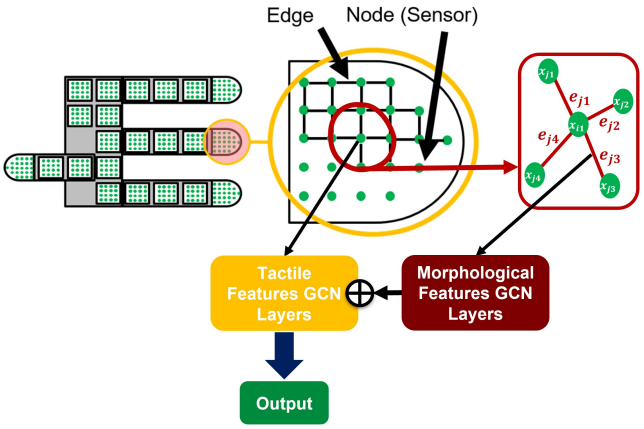

Fig. 1. Proposed method of GCN with tactile features and geometrical edge features. The tactile features are basically graph node features, and the geometrical edge features embed spatial information of the graph.

Even though vision can recognize certain external properties of an object, it fails to detect the internal properties, like weight of the object, texture details, softness etc. Furthermore, often the vision might be blocked either by the robot or by external factors like in the case of a household environment. Therefore learning to recognize properties with touch states is pertinent for multifingered dexterous tasks [2]. However, as mentioned, the diversity of features and subsequent diversity of sensory data coming from day-to-day objects make it challenging.

In our previous work, we implemented Morphology-Specific Convolutional Neural Networks (MS-CNN) for the task of object recognition with touch states using multi-fingered hand with distributed tactile sensors [3]. However, applying a rectangular CNN filter on tactile sensors implemented morphologically onto surface of irregular geometry is difficult. Furthermore, the sensor sequence of sensor-map can vary from task to task, if the morphology and configuration of the hand is to be taken into consideration, and hence, a sequence invariant sensor map for CNN can affect the output of the network.

To address these issues, we implemented Graph Convolutional Neural Network (GCN) and achieved in-hand manipulation with 8 objects using distributed tactile sensors [4]. Since the graph feature space is non-Euclidean and inherently permutation invariant [5][6], and the sensor-map is constructed with an adjacency matrix, the sensor-map for GCNs resembles more closely to the real configuration of the sensors on the hand. This model could produce decent success rate for object manipulation. Yet, the model requires manually prepared object property labels. We observed, that

these input property labels made the hand motion more adaptive towards the respective object, which emphasizes the importance of features extraction of day-to-day objects and recognizing their properties.

One of the many factors that enables humans to achieve complex dexterous tasks including object property recognition is the ability to recognize the shape and proprioceptive state of the body. Research suggests that if the communication between the region of the brain that has the topological map of the body (S1 Cortex) and the region of the brain responsible for motor control (Motor Cortex) is hampered, the ability to perform dexterous tasks will be affected [7]. Furthermore, tactile perception is accomplished with integration of tactile states and proprioceptive states [8]. With the awareness of the proprioceptive state and the form of the hand, humans know exactly how to use their hands to accommodate the diversity and randomness of the everyday object features. However, there has not been enough investigation of multi-fingered dexterous tasks with tactile sensors in this aspect.

Hence, to enhance the object property recognition performance, this paper proposes a novel approach of incorporating proprioception and hand morphology with GCNs. We propose the concept of utilizing the edges of graphs to produce spatial feature embeddings between target nodes and their neighbours. We also introduce a novel GCN architecture for employing these edge features. The proposed architecture, in abstract terms, is presented for learning multiple feature modalities of nodes within the same graph. We trained GCN models and evaluated the effectiveness of proposed edge features and GCN architecture for object property recognition. We further trained Feed Forward Neural Networks (FNNs) by flattening the input features and Convolutional Neural Networks (CNNs) by creating feature maps from the input features to establish baselines against the proposed method. We used 8 objects in the process and trained the models to recognize 6 properties for each object. We make following contribution to the field with this paper :

- A novel approach of utilizing graph edges in GCNs to embed intra-graph spatial features.
- A novel Multi-Thread Graph Convolutional Network (MT-GCN) architecture to treat multi-modal data in a graph.
- Achieving high object property recognition rates with the proposed methods.
- Analysis of proposed MT-GCN performance with different edge connections, that is with different adjacencies.

II. Related Work

A. Spatial Processing and Tactile Sensing

Tactile sensors provide robotic systems with an emulation of the tactile sense. To obtain large volume of tactile information, the robot can be covered with an array of them to maximum possible extent. In order to process the huge volume of tactile information, deep learning methods specifically, CNNs are effective at spatial processing [9]. Tactile sensors can provide rich haptic information of global and local features of an object, which is extremely useful for object recognition and object feature recognition. Works [3][10][11][12] achieve object recognition with tactile sensors and multi-fingered robot hands. However, these works recognize the objects but not the properties, limiting their ability to generalize. [11] presents an object feature extraction with tactile sensor data and random forest algorithm to predict stability grasp with an under-actuated gripper. There are a few works, which detect object features with tactile sensors and a multi-fingered hand. [13] demonstrated object property recognition with the tactile sensor 'ReSkin' and a Long Short Term Memory (LSTM) network.

Along with local and global object features, tactile sensors also provide rich geometrical and morphological information, and hence Graph Neural Networks are extremely effective for processing tactile data [14]. Geometrical information, such as the contact states of vertices and edges provided by tactile sensors interacting with an object, can be useful for classifying object features [15]. Several studies used Graph Neural Networks to apply them to non-rectangular shaped tactile sensing areas [14]. Multiple studies investigated the effects of varying number of graphs edges on the performance of Graph Neural Networks. It was observed, that increasing the number of edges does not necessarily augment the useful information; yet, it inflates the features for training the networks. This generally led to a worsening of performance [16], [17], [18]. Overall, while exploring the morphological perspective for improvements in robotic tactile tasks, ensuring stable enhancements remains ambiguous.

B. Graph Neural Networks

Graph-based neural networks are very effective for learning geometry based applications such as molecule fingerprints [19]. In robotics, graph based networks are used for motion planning [20], among other tasks [21]. GCNs are also very powerful in extracting topological and morphological features. Edge information can also provide useful information for graph structured applications [22]. However, the mathematical treatment of the graph edge features poses a problem, as the matrix multiplication is impossible when the nonzero values in the adjacency matrix are replaced with a (1 x n) row vectors of edge features. There are some studies that address this problem. [23] proposes a node-edge feature convolutional model with significant emphasis on attention. [24] employs a dual level attention for node features and edge features, respectively. However, our work is related to morphology, not node or edge importance. Hence, we argue, that all edge-features are equally important to acquire entire morphological information of the hand. In [25], aggregated edge features of edges originating from a target node according to the adjacency are concatenated during message passing.

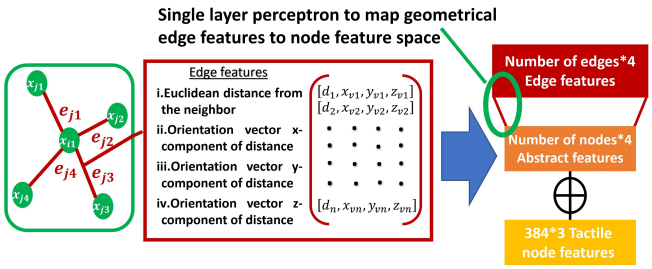

Fig. 2. Geometrical edge features, and proposed method for their inclusion in the GCN. We use a single layer perceptron to map edge features to the features space of nodes.

III. Proposed Method

A. Graph Convolutional Networks

We used a Graph Convolutional Neural Network to learn the tactile states. The graph is made of distributed tactile sensors, each taxel being a node with features being the sensor values in X, Y and Z directions, respectively. The output of n^{th} GCN layer, $f(H^n,A)$ can be expressed as :

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

Here, H^n are the input node features of n^{th} Graph Convolutional layer. A is the adjacency for the graph structure of the uSkin sensors on the Allegro Hand. The output of this layer is $(n+1)^{th}$ Graph Convolutional layer's input, H^{n+1}. $\hat{A}$ is the sum of adjacency matrix, A and an identity matrix I, to consider both neighbouring 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 introduced against the scale change of the features. W^n is the weight matrix of the n^{th} Graph Convolutional layer and σ is a nonlinear activation function.

B. Geometrical Edge Features

The node features are the tactile sensor values in X, Y and Z directions, whereas the edge features carry spatial information of the hand, tracing positions of nodes (taxels) relative to their neighbourhood. As shown in Fig. 2, the edge feature vector is formulated as follows :

$$e_{ij} = [d_{ij},\ v_{xij},\ v_{yij},\ v_{zij}] \quad (2)$$

Where, i and j are target node and neighbour node respectively, d_{ij} is the Euclidean distance between target node and neighbour node. $(v_{xij},v_{yij},v_{zij})$ is the orientation vector of d_{ij} along X, Y and Z directions, respectively, calculated by simply subtracting the coordinates of the target nodes from the coordinates of neighbour node.

The adjacency matrix A in equation (1) consists of 1's when two nodes are connected (edge is present) and 0's when two nodes are not connected (edge is absent). Replacing the 1's with edge feature vectors renders matrix multiplication impossible in equation (1). Hence, to address the problem posed by the edge features, we propose a method to map the edge features into node feature space. Edge features are mapped into node feature space using a single-layer perceptron, with an input dimension of (number of edges * 4) and an output dimension of (number of nodes * 4). In this paper, this particular perceptron layer is denoted as the 'edge feature encoder'. The output of the edge feature encoder can be used for Graph Convolutions, as shown in Fig. 2.

C. Multi-Thread Graph Convolutional Network (MT-GCN)

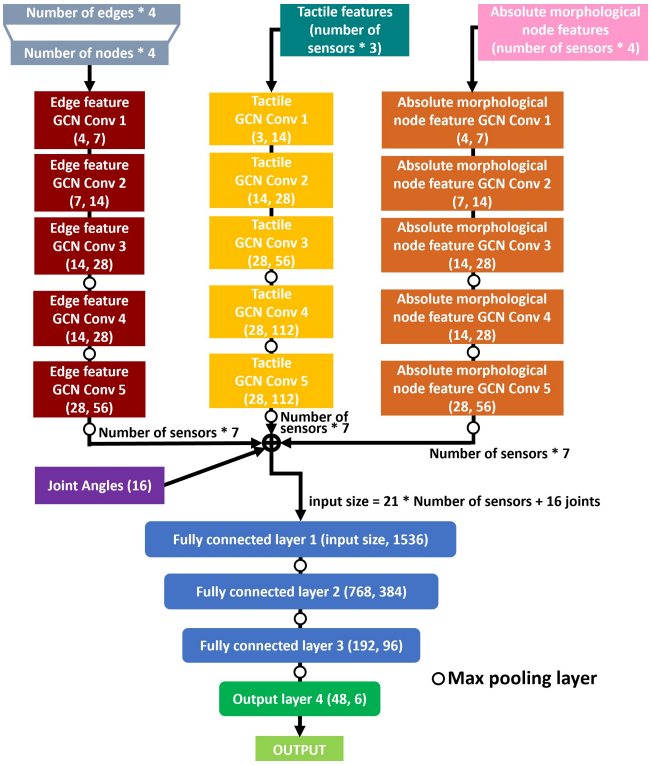

Fig. 3. Proposed MT-GCN model. Different modalities of graph node features are convoluted in parallel and independent Graph Convolutional layers threads, and the extracted features are fused after the final Graph Convolutional layer.

Instead of concatenation of node features of different modalities, to pass it through as single Graph Convolutional layer thread, we propose a method of simultaneous Graph Convolutions on features of different modalities and different feature spaces associated with the same graph.

In the proposed MT-GCN architecture, we process each node feature modality of the graph nodes in parallel independent threads of Graph Convolutional layers as shown in Fig. 3. After the final Graph Convolutional layers of the respective threads of each modality, the features are fused for further utilization. Hence, for a MT-GCN model for a graph with M modalities of node features and with N number of GCN layers in each modality thread, the output of the MT-GCN at the end all GCN layers is expressed as :

$$H^{output} = (f(H_1^N,A), f(H_2^N,A), f(H_3^N,A), ..., f(H_M^N,A)) \quad (3)$$

This equation assumes that adjacency, A of all the node features is same, since they belong to the same graph. However, even for the same graph, if we have different adjacency matrices for different node feature modalities, this equation can be generalized. For a feature modality m, we can simply replace the adjacency matrix A by adjacency matrix A_m. Similarly, for dissimilar number of GCN layers for each modality, the layer number term N can be replaced

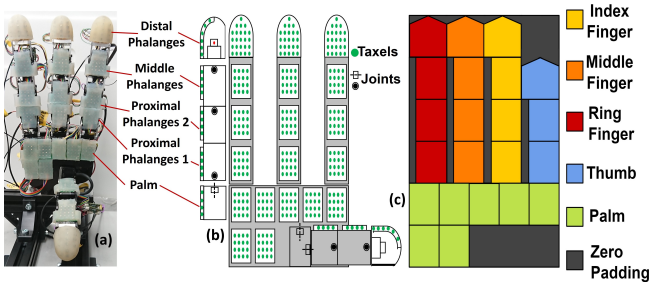

Fig. 4. (a) Allegro hand with uSkin, (b) Sensor map of Allegro hand with uSkin, (c) Feature map of Allegro hand with uSkin.

by N_m for modality m. Since the goal for this work is to investigate task improvement across different modalities and various combinations thereof, the adjacency matrix and the number of GCN layers in the GCN threads are identical across all modalities.

D. Absolute Morphological Node Features

One of the more common methods to embed morphological features is by using the global position of the taxels. To conduct a thorough investigation, we trained GCNs with feature embeddings representing the absolute positions of each taxel at each timestep in the global frame of reference. For a fair comparison, we ensured that these features are similar to the predefined edge features. An absolute morphological node feature vector is formulated as follows :

$$a_{io} = [d_{io},\ v_{xio},\ v_{yio},\ v_{zio}] \tag{4}$$

Where, i is the target node and o is the origin. d_{io} is the absolute euclidean distance of the target node. $(v_{xio}, v_{yio}, v_{zio})$ is orientation vector of d_{io} along X, Y and Z directions, respectively. Since these features describe the global position of the taxels, they are associated with the nodes of the graph and not the edges.

IV. Experiment Design

A. Training Data

For the experimental setup of this study, we used the Allegro Hand, a 16 DOF multi-fingered (four fingered) robotic hand made by Wonik Robotics, shown in Fig. 4. We covered its fingertips, phalanges and palms with uSkin distributed tactile sensors, as in our previous study [4]. The tactile sensor 'uSkin' from Xela Robotics is a magnetic type tactile sensor, which senses the movement of small magnets infused in a soft silicone skin, by measuring raw values of their magnetic fields along X, Y, and Z directions, respectively [26]. uSkin's sampling rate is 100 Hz. A uSkin fingertip is shown in Fig. 5. Overall, the number of measurements on the Allegro Hand adds up to : 16 (4 fingers * 4 joint angles) + 1152 ((4 fingertips * 24 uSkin sensor chips) + (11 finger phalanges + 7 uSkin sensors on a palm) * 16 uSkin sensor chips) * 3 axes = 1168 (Fig. 4).

Each object was put in a random pose below the Allegro Hand and the data was collected from a motion of picking up and holding eight such objects from a desk by remotely controlling the Allegro Hand using a CyberGlove (22-sensor model) from CyberGlove Systems as shown in Fig. 5. This

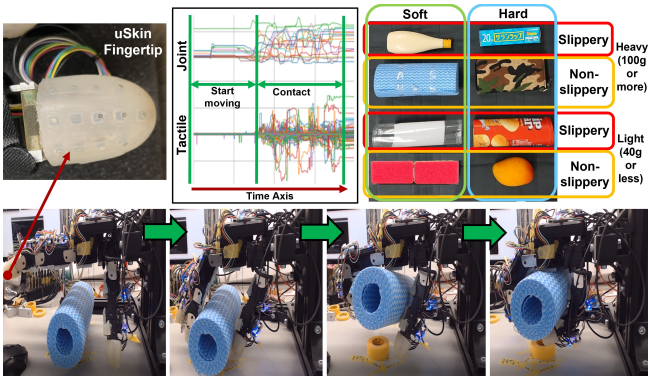

Fig. 5. Data collection setting. Top right image shows objects used for training data. Top middle image shows a plot of joint trajectories and tactile values against time. Top left image shows a uSkin fingertip. Bottom images show an example of motion for data-collection with kitchen paper.

motion was chosen because it facilitates contact between the object and the entire hand, resulting in a robust and reliable dataset for the task of object property recognition. For each object, 10 successful trials of the target motion were conducted, each for about 17 seconds. Thus, with eight objects, a total of 80 trials of training data was collected. Each recorded trial was pre-processed before using as training data for the GCN. 70% of the data collected from each object was used for training, and 30% for testing. One third of the test dataset (10% overall) was used as the validation set. Data collected in each time-step can provide the entire spatial information of touch states for that respective object. Therefore, data in each timestep serves as an input to the network. We did not use a time-scaled network in this evaluation because, for a given task of object property recognition, although the input comprises time-series data, the output, namely, the object property label, is time independent. Therefore, the focus of inquiry in this work is spatial processing.

We selected six object properties commonly encountered in everyday objects, based on heaviness, hardness, and slipperiness. Eight common daily objects were chosen as shown in Fig. 5, a plastic tube (light, soft and slippery), a sponge (light, soft and non-slippery), mayonnaise (heavy, soft, and slippery), kitchen paper (heavy, soft, and non-slippery), a potato chip cylinder (light, hard, and slippery), a replica of a mango (light, hard, and non-slippery), saran wrap (heavy, hard, and slippery) and a purse (heavy, hard, and non-slippery) were used. We prepared six labels in the following sequence: light, heavy, hard, soft, non-slippery, and slippery, and assigned a value of 1 to each label when the object possessed that respective property; otherwise, it was assigned a value of 0. For example, for a heavy, soft, and non-slippery object (kitchen paper), the labels would be [0,1,0,1,1,0]. The proposed method was expected to learn to predict these labels.

B. Neural Network Settings

Table I shows the setting of the neural networks used for the evaluation. The Roman numerals next to the models listed below, as well as assigned to Table I columns are associated with their respective input combinations. In Table I, T stands

TABLE I

NEURAL NETWORK SETTING

Input Combinations →	I	II	III	IV	V	VI	VII
GCN, CNN Input Types and Size	T (3 x 384)	E (4 x edges)	A (4 x 384)	T (3 x 384), E (4 x edges)	T (3 x 384), A (4 x 384)	E (4 x edges), A (4 x 384)	T (3 x 384), E (4 x edges), A (4 x 384)
GCN, CNN Layers Size / MT-GCN, MT-CNN Layers Size	(14, 28, 56(P)28, 112(P)28, 112(P)7) / T-Thread : (14, 28, 56(P)28, 112(P)28, 112(P)7) E, A - Thread : (7, 14, 28(P)14, 28(P)14, 56(P)28)						
FC Layers Input size GCN, CNN	2688 + 16 Joints						
MT-GCN, MT-CNN	-			5376 + 16 Joints	5376 + 16 Joints	5376 + 16 Joints	8064 + 16 Joints
FC Layers Size	(1536(P)768, 384(P)192, 96(P)48)						
Output	6 Object Property Labels						

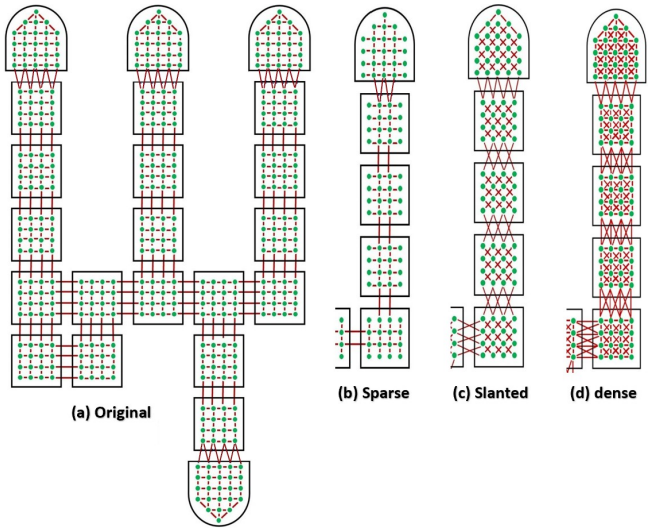

Fig. 6. Different adjacencies: We test the network with 4 adjacencies. Sparse (b) adjacency, with vertical and horizontal edges with reduced density, Slanted (c) with only slanted edges, Original (a) used in our previous GCN work, with only vertical and horizontal edges, except of edges at the end of fingertips and dense (d) where there are edges between all neighbours of a node and the target node. This adjacency has maximum edge density.

for Tactile input, E stands for Edge Feature Input and A stands for Absolute Morphological Node Feature input. Thus, with these inputs, to evaluate the performance of proposed method, we trained the following GCN, CNN and FNN models :

- GCN, CNN and FNN trained with tactile features (I)
- GCN trained with geometrical edge features (II)
- GCN, CNN and FNN trained with absolute morphological node features (III)
- GCN trained with tactile & geometrical edge features combined (IV)
- GCN, CNN and FNN trained with tactile & absolute morphological node features combined (V)
- GCN trained with geometrical edge features & absolute morphological node features combined (VI)
- GCN trained with tactile, geometrical edge features & absolute morphological node features combined (VII)
- MT-GCN trained with tactile & geometrical edge features (IV)
- MT-GCN, MT-CNN trained with tactile & absolute morphological node features (V)
- MT-GCN trained with geometrical edge features & absolute morphological node features (VI)
- MT-GCN trained with tactile, geometrical edge features & absolute morphological node features (VII)

All these models were trained with and without joint angle input from the Allegro Hand. Hence we have total twenty two GCN models, eight CNN models and six FNN models. FNNs and CNNs were trained to review the performance of the GCNs against the established methods. However, FNNs and CNNs were not trained with edge features, since edges only exist in graphs. We did not train a time-scaled network similar to [13] for comparison, since the scope of this work is limited to spatial processing. Each model was trained five times and the mean accuracy of these five iterations was calculated to accommodate for variance, and in turn produce fair results.

CNNs were trained by converting the tactile and absolute morphological node features input into feature maps, as shown in Fig. 4. We also trained MT-CNNs, with parallel Convolutional layers threads for different modalities of input features. The layer size and optimizations applied to both CNNs and GCNs, as shown in Table I, are identical to obtain fair comparison. Max-Pooling layers are denoted by (P) in Table I. The value before (P) indicates number of output features of GCN and CNN layer before Max-Pooling, whereas the value after (P) is the number of features after the Max-Pooling, which is also the input to the next GCN and CNN layer. Pooling layers are also used between the fully connected layers of GCNs and CNNs for optimization as shown in the Table I. The input size to fully connected layer depends on the outputs of the GCNs and CNNs.

The FNNs were trained with flattened tactile and absolute morphological node features. The input size of FNNs for input combinations Tactile, Absolute Morphological Node Features and Tactile + Absolute Morphological Node Features is 1152, 1536 and 2688 respectively without joint angle input, and the mentioned dimensions plus 16 joint angle features in case of joint angle input. The FNN layer sizes are (32768(P)16384, 4096(P)2048, 1024(P)512). The FNN layer sizes are large to compensate for the large number of neurons otherwise present in the GCN and CNN layers in order to draw a fair comparison. The aforementioned hyper-parameters were empirically and heuristically chosen.

Since the proposed method involves features embedded through the graph edges, we also evaluated the performance of the proposed model for different settings of edge connections - 'sparse', 'slanted', 'original', 'dense'. As shown in Fig. 6, 'sparse' adjacency matrix has least number of edges, followed by 'slanted', 'original' and finally 'dense'.

TABLE II

COMPARISON OF RECOGNITION RATES WITH ALL INPUT COMBINATIONS AND CNN, FNN AND MT-CNN BASELINES

	Joint Angles not Inputted							Joint Angles Inputted						
	T	E	A	T + E	T + A	E + A	T + E + A	T	E	A	T + E	T + A	E + A	T + E + A
FNN	73.52	-	75.44	-	77.23	-	-	78.31	-	80.43	-	79.15	-	-
CNN	82.77	-	64.98	-	83.84	-	-	81.72	-	74.59	-	83.19	-	-
MT-CNN	-	-	-	-	82.37	-	-	-	-	-	-	81.05	-	-
GCN	79.47	81.75	72.00	84.79	79.20	81.75	85.09	84.90	81.56	79.78	84.92	85.05	81.56	84.92
MT-GCN	-	-	-	84.80	80.86	82.67	83.54	-	-	-	**86.08**	85.85	81.50	85.57

We also aim to compare the results of the proposed model with adjacencies 'sparse', 'slanted' and 'dense' against result of proposed model with 'original' adjacency, which was used in our previous work with GCN [4]. We aim to examine the effect of variation in graph adjacency on the network performance in this evaluation because we are using the graph edge features in our method.

V. EVALUATION

A. Comparison of Network Recognition Rates

Table II shows the recognition rates of each model we trained as the part of experimental setting. Among the GCNs, the lowest recognition rate is produced by the GCN with only absolute morphological node features without joint angles. The recognition rate of GCN with only tactile input without joint angles is also quite low. However, both of the recognition rates increase significantly with joint angle input. We can see this pattern in case of both GCN and MT-GCN models as well, which have the input combination of tactile and absolute morphological node features. This indicates that absolute morphological node features are not sufficient on their own for extracting proprioceptive features.

The recognition rate of the GCN model with only edge features changes minimally with respect to the joint angle input. This can also be observed with both GCN and MT-GCN models, which have a combined input of edge features and absolute morphological node features. Furthermore, GCN models with tactile data and edge features, even without joint angle input, have almost the same recognition rates as GCN models with tactile features and joint angles. This is because edge features and joint angles fall under the same umbrella of modality. Yet, this result also indicates the ability of the edge features to convey proprioceptive information.

In case of CNNs, the best model is with input of tactile and absolute morphological node features without joint angles. Without the joint angles, the CNN's ability of pure spatial processing and pattern extraction of tactile features can be observed to be better than GCN. However, when tactile modality is present, both CNNs and MT-CNN perform worse when joint angles are given, as opposed to most of the GCN and FNN models. This indicates the limitation of CNNs, in correlating proprioceptive modality to tactile modality. Furthermore, CNNs exhibit the lowest recognition rate among all the models when only absolute morphological node features are inputted without joint angles, indicating CNNs' limited ability to interpret the data as a morphological entity in spatial processing. With joint angles input, the recognition rate improves, as joint angles and absolute morphological node features lie in the similar realm of modalities.

For FNNs, we observe a pattern similar to that of GCNs, where models with joint angles outperform the models without joint angles for all the input combinations. Except for FNNs with absolute morphological node features inputs with and without joint angles, the rest of the FNNs results are worse than GCNs. Overall, the results of FNNs and CNNs emphasize the significance of morphological representation in features extracted from various modalities of data by the GCNs for dexterous tasks with multifingered robotic hands.

The best model is the MT-GCN (86.08%) with tactile and edge features with joint angles. Given that it is superior to MT-GCN with a combination of tactile and absolute morphological node features with joint angles indicates that edge features are more effective than absolute morphological node features in case of the multi-thread architecture. From the result, it can be seen that the proposed MT-GCN has a better tendency of feature extraction from the inputs compared to other models.

B. Analysis of Edge Feature Encoder

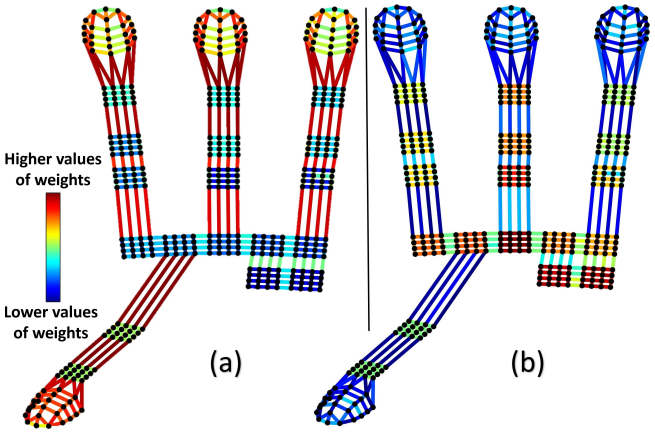

Fig. 7. PCA analysis of edge feature encoder parameter matrix of proposed MT-GCN model shows that the encoder can distinguish between inter-patch edges and intra-patch edges. (a) shows the map of MT-GCN without joint angles input, while (b) shows the map of MT-GCN with joint angles.

We visualized the weights of the edge feature encoder of MT-GCN with tactile and edge features input combination. The size of edge feature encoder's parameter matrix is (number of nodes x 4) rows X (number of edges x 4) columns. Hence, every 4 columns are associated with four features of one edge, which are mapped as 4 features in the node space. We calculated mean of absolute values of these 16 features for each edge and obtained a matrix of dimension (number of edges) x (number of nodes). We further used principle component analysis to reduce the dimensions associated with each edge to 1, thus obtained a vector of dimension equal to

number of edge features. We then produced a colour-map out of this vector. Fig. 7 shows the edges presented with colours associated with their respective weight values according to the colour-map.

Fig. 7 (a) shows the result of this analysis with MT-GCN and without joint-angle input. We can see a clear pattern with the values of the learned parameters, where the edges between the joints have similar values marked by redder colours while edges within patches, and edges between patches on the palm are mostly bluer. The edges within fingertips are warmer in colour than the rest of the intra-patch edges because fingertips, being the last links of the fingers with only one joint at their base, experience maximal movement with respect to the hand's origin, being farthest from it. The figure shows that edge feature encoder can recognize the positions of the joints without the input of joint angles, which implies that edge features can convey the information of proprioception to the model.

Identical analysis of the learned parameters of the edge feature encoder in MT-GCN with joint angle input reveals the same pattern, except that the joint edges are bluer and intra-patch edges are redder, as shown in Fig. 7 (b). Since the joint angles are provided, identifying joint positions is sufficient for the edge feature encoder. As a result, the model effectively learns proprioception, but the weight values associated with joint edges are smaller, resulting in bluer edges.

C. Analysis on Edge Connections

TABLE III

RECOGNITION RATE OF MT-GCN WITH DIFFERENT ADJACENCY MATRIX WITH DIFFERENT EDGE DENSITY.

Adjacency	Sparse	Slanted	Original	Dense
Number of Edges	1000	1184	1401	2488
Accuracy	84.79	85.23	86.08	85.9

We evaluated recognition rate of the MT-GCN with four adjacencies of different number of edges and hence different edge densities shown in Fig. 6. With increase in edge density, the model receives more edge features. Table III shows the recognition rates of the MT-GCN model with tactile and edge features and joint angle input for each adjacency.

Until the 'original' adjacency, as the number of edges in the graph, i.e. the edge density increases, the recognition rate of the network increases due to increase in the spatial features provided to the model through the edges. In previous works of Graph Neural Networks with tactile features, [16][17][18], an opposite result has been obtained. In the mentioned works, the performance of the model almost always decreased with increase in the edge-density. Hence, this result indicates that with presence of edge features in a GCN, increase in edge density in adjacency matrix may lead to increase performance from the network.

However, a very small dip in the recognition rate in case of 'Dense' adjacency is observed, although its recognition rate is still higher than 'Sparse' and 'Slanted' adjacency. 'Dense' adjacency has more than double number of edges than the original adjacency, and this might have led to a degree of saturation in the graph.

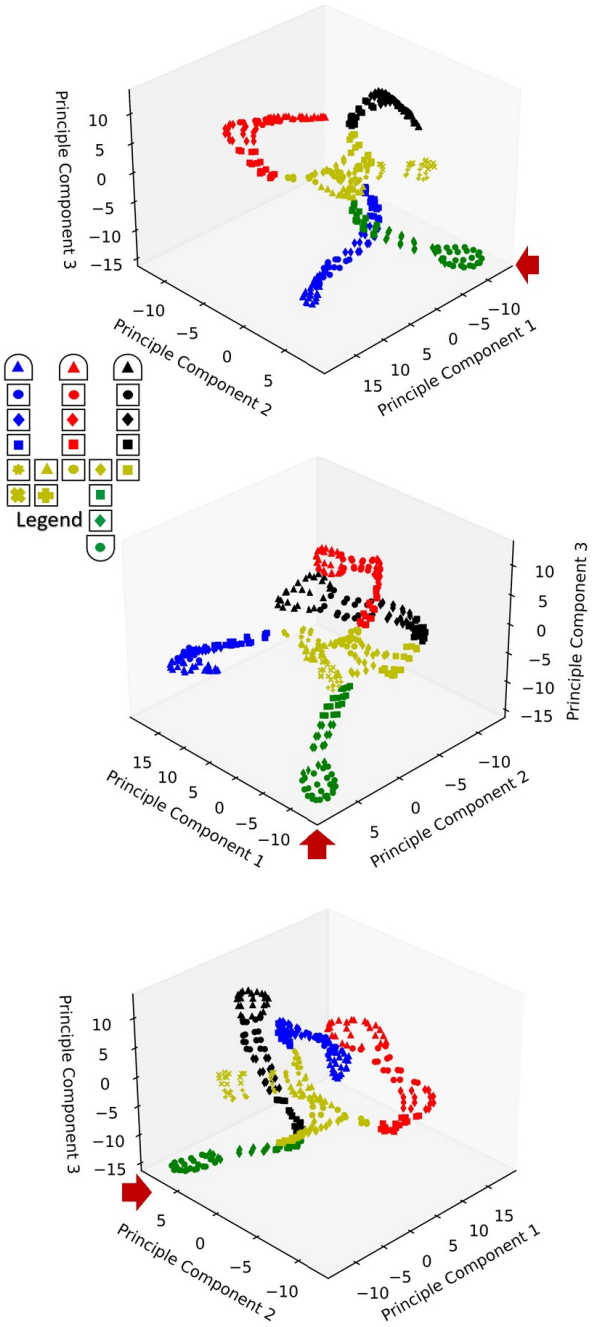

Fig. 8. Feature visualization of last layer of MT-GCN model with t-SNE. Clear clusters corresponding to different parts of the hand can be observed. Moreover, the clusters follow the morphology of the hand and exact sequence of the sensor patches on the Allegro Hand.

D. Visualization of GCN Features

Fig. 8 shows the t-SNE plot of features generated from the output of the final layer of the tactile thread of MT-GCN with tactile features, edge features and joint angles. Three different views of the same figure are presented. The red arrow in the figure serves as the reference to determine the viewing angle of the plot. The adjacent small figure of hand plays the role of 'legend' for each colour and shape corresponding to each sensor patch on the Allegro Hand.

The plot shows clear clusters formed from features extracted by the proposed model. Each cluster represents a

part of the hand, namely the digits and the palm. The clusters follow not only the exact sequence of the digits of the Allegro Hand, but also the sequence of the sensor patches within the digits. We can see from this result, that the proposed Graph Convolutional Network can effectively extract the morphological features of the hand with tactile information from distributed tactile sensors.

VI. CONCLUSIONS

This paper proposes a formulation of geometrical graph edge features, which are vectors comprising the Euclidean distances between graph nodes and their neighbours, along with their orientation, to create a spatial embedding in a graph for combined tactile and proprioceptive representation. The edge features are then mapped to node feature space using an edge feature encoder. A deep learning model, MT-GCN, is presented to incorporate the geometrical edge features for object property recognition. The proposed MT-GCN, with edge features, tactile features, and joint angle input achieved the best object property recognition rate.

Furthermore, we demonstrated the utility of the edge features for learning proprioception. We showed that increasing the number of edges in the graph is effective when using edge features. Additionally, we exhibited the importance of combining proprioceptive and tactile representations for dexterous tasks, demonstrating the limitations of CNNs in achieving them and effectiveness of the GCNs in producing them.

For future research, we plan to implement and evaluate the proposed network for real-time object property recognition and motion generation.

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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] A. Thompson, D. Murphy, F. Dell'Acqua, C. Ecker, G. McAlonan, H. Howells, S. Baron-Cohen, M.-C. Lai, and M. V. Lombardo, "Impaired communication between the motor and somatosensory homunculus is associated with poor manual dexterity in autism spectrum disorder," *Biological Psychiatry*, vol. 81, no. 3, pp. 211–219, 2017, genetic and Epigenetic Risks for Autism Spectrum Disorder. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0006322316325331

[8] H. T. S. Warren JP, Santello M, "Effects of fusion between tactile and proprioceptive inputs on tactile perception." in *PLoS One. 2011 Mar 25;6(3):e18073. doi: 10.1371/journal.pone.0018073. PMID: 21464943; PMCID: PMC3064587.*

[9] 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.

[10] 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.

[11] 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.

[12] J. Hoelscher, J. Peters, and T. Hermans, "Evaluation of tactile feature extraction for interactive object recognition," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 310–317.

[13] R. Bhirangi, A. DeFranco, J. Adkins, C. Majidi, A. Gupta, T. Hellebrekers, and V. Kumar, "All the feels: A dexterous hand with large area sensing," 2023.

[14] 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.

[15] W. Yuan, C. Zhu, A. Owens, M. A. Srinivasan, and E. H. Adelson, "Shape-independent hardness estimation using deep learning and a gelsight tactile sensor," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 951–958.

[16] 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.

[17] 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.

[18] 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.

[19] 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

[20] 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.

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

[22] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 11–33, 2016.

[23] L. Gong and Q. Cheng, "Exploiting edge features in graph neural networks," 2018. [Online]. Available: https://arxiv.org/abs/1809.02709

[24] Y. Yang and D. Li, "Nenn: Incorporate node and edge features in graph neural networks," in *Proceedings of The 12th Asian Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, S. J. Pan and M. Sugiyama, Eds., vol. 129. PMLR, 18–20 Nov 2020, pp. 593–608. [Online]. Available: https://proceedings.mlr.press/v129/yang20a.html

[25] T. Xie and J. C. Grossman, "Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties," *Physical Review Letters*, vol. 120, no. 14, apr 2018. [Online]. Available: https://doi.org/10.1103%2Fphysrevlett.120.145301

[26] 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