Abstract

Point set classification aims to build a representation learning model that distinguishes between spatial and categorical configurations of point set data. This problem is societally important since in many applications domains such as immunology, and microbial ecology. This problem is challenging since the interactions between different categories of points are not always equal; as a result, the representation learning model must selectively learn the most relevant multi-categorical relationships. The related works are limited (1) in learning the importance of different multi-categorical relationships, especially for high-order interactions, and (2) do not fully exploit the spatial distribution of points beyond simply measuring relative distance or applying a feed-forward neural network to coordinates. To overcome these limitations, we leverage the dynamic graph convolutional neural network (DGCNN) architecture to design a novel multi-category DGCNN (MC-DGCNN), contributing location representation and point pair attention layers for multi-categorical point set classification. MC-DGCNN has the ability to identify the categorical importance of each point pair and extends this to N-way spatial relationships, while still preserving all the properties and benefits of DGCNN (e.g., differentiability). Experimental results show that the proposed architecture is computationally efficient and significantly outperforms current deep learning architectures on real-world datasets.

1 Introduction

Point set classification aims to build a representation learning model which can learn spatial patterns in point set data to distinguish between spatial and categorical configurations of point set data for the underlying task. When the point set additionally consists of categorical point set features, it is intuitive that the importance of distinct categorical interactions between points is additionally significant for the classification task. For example, the impact of cytotoxic T lymphocytes (CTLs) on nearby cancer cells may be affected by other immune cells (e.g., T regulatory cells) [18]. A multi-categorical point set is a collection of spatially-defined objects with corresponding categorical features (e.g., immune and tumor cells). For example, Fig. 1(b) shows the corresponding point set representation of a pathologist-driven field of view (FOV) of a tissue sample stained with hematoxylin/eosin (H&E). Each point shows the centroid coordinates of a cell in a pixel and the color of its corresponding cell type.

Figure 1: (a) A field of view of a tissue sample stained with hematoxylin/eosin (H&E), (b) the corresponding point set representation.
relationships are not always equal, which requires that the representation learning model learn these distinctions. Second, the discontinuous spatial distribution of multi-category point sets (e.g., immune and tumor cells) form complex structural and higher-order local spatial relationships. Third, the representation learning model needs to maintain certain invariance properties, such as permutation invariance, in order to achieve a robust surrogate representation.

PointNet [12] was the pioneer deep learning technique for point cloud data, which applies a multi-layer perceptron to each data point separately and then accumulates the features using an asymmetric function. PointNet++ [13], a variant of PointNet, proposes to represent local information by choosing reference points using farthest point sampling, subsequently building their corresponding k-nearest neighborhoods, and then applying a shared PointNet network to learn the local structure. However, these techniques are both limited in learning local information mainly due to operating on points independently at a localized scale to maintain permutation invariance. In addition, the sampling procedure and graph coarsening operation induce loss of information since data points get discarded in each iteration. Dynamic graph convolutional neural network (DGCNN) [17] addresses this issue by proposing an edge convolution (EdgeConv) operation to capture local geometric information by aggregating features along the edges using a neighborhood graph. However, this technique is not designed to learn spatial relationships in multi-category point sets and does not fully exploit the spatial distribution of points beyond simply measuring relative distance or applying a feed-forward neural network to spatial coordinates [7]. More recent work, a spatial-relationship aware neural network (SRNet) [10] proposes an architecture to model relationships between points of different categories. However, SRNet is limited to only binary spatial relationships, and the importance between different binary category pairs is assumed to be equal.

To overcome these limitations, we leverage DGCNN architecture to design a Multi-Category DGCNN (MC-DGCNN), a neural network architecture for multi-category point set classification, with novel location representation and point pair attention layers. MC-DGCNN provides a promising way to identify categorical N-way spatial relationships and the relative importance of point pair instances while preserving all the properties of the DGCNN. A point pair instance describes the spatial relationship between a point and its neighbor belonging to two categories.

Fig. 2 shows the overall framework of the proposed MC-DGCNN. As shown in Fig. 2(a), we first aim to provide a better way to represent local information using a multi-scale position embedding and spatial feature decomposition (described in Section 5.1) before applying EdgeConv [17]. As indicated in Fig. 2(b), a point pair attention network is designed to specify attention coefficients of point pair instances in multi-category point interactions in an N-way spatial relationship (described in Section 5.3). Two and three spatial relationships indicate by hyperedges connecting vertices belonging to different categories. Lastly, we use an asymmetric function (e.g., average pooling) to aggregate information across all nodes neighboring node \( \hat{v}_i \). The thickness of different edges shows the attention co-efficient between different category pairs in the overall representation of \( \hat{v}_i \). However, the choice of aggregation function is not limited to average pooling. For example, a MAX pooling operation, followed by average pooling, may be used to exploit the most important \( k' \) point pair instances out of \( k \) point pairs.

**Key Contributions.** The key contributions of our work are as follows:

- We propose a novel neural network architecture, namely MC-DGCNN, to identify N-way (e.g., tertiary, ternary, etc.) spatial relationships.
- We conduct extensive experiments to show that the proposed model significantly outperforms existing deep neural network architectures.
- We show that our proposed model achieves the best trade-off in computational time complexity and classification accuracy with competing DNN architecture (e.g., SRNet).

**Scope:** In this work, we do not evaluate the proposed method with larger datasets due to a lack of public benchmarks. Analyzing the effect of standard data augmentation techniques (e.g., rotation) is not in the scope of this paper. Field trials to assess the clinical value of the proposed method also fall outside the scope of this study.

**Organization:** The rest of the paper is organized as follows: Section 2 briefly describes an important application domain of this problem. Section 3 introduces the formal definition of the problem and very briefly describes the background of this work. Related work is reviewed in Section 4. Section 5 describes the details of our proposed work. In Section 6 we present the results. Finally, Section 7 concludes this paper and outlines some future research.

2 An Illustrative Application Domain

The recent development of multiplex immunofluorescence (MxIF, Fig. 1) technology has enabled exploration into the complexity of tumor-immune microen-
environments within spatially-preserved metastatic tissue and in the therapeutic context of immune checkpoint inhibitors (ICI). ICI therapy works by augmenting the antitumor properties of pre-existing tumor-specific CTL, which become more efficient in infiltrating tumor masses and destroying cancer cells. Through cyclic rounds of antibody staining, imaging and dye inactivation, MxIF technology provides a state-of-the-art method to visualize and identify many cell subtypes (e.g., immune and malignant) and their corresponding spatial coordinates using single-cell analysis of formalin-fixed paraffin-embedded (FFPE) tissue sections including metastatic melanoma lymph nodes. With continuous refinement of these techniques, it is currently possible to identify over 50 cellular phenotypic (e.g., CD3, FoxP3, CD14) and their corresponding cellular characteristics within a single tissue section.

Emerging research in this area has begun to highlight the importance of an automated process to analyze the complex spatial relationships among different cellular subsets and functional states in the context of ICI therapy, which allows identifying critical intercellular interactions relevant to clinical outcomes [20]. Furthermore, it is clinically crucial to examine the importance of cell species in complex spatial relationships along with their activation states in a spatially informed manner due to the clinical implications of interactions in close spatial proximity. For example, a CTL will likely be unable to kill a cancer cell if it is nearby a tumor-associated macrophage that expresses PDL1 on its surface. In contrast, a CTL is more likely to kill a cancer cell if it is expressing Granzyme B and is not in close proximity to FoxP3-expressing regulatory T cells (Treg). To this end, it would be beneficial to provide an algorithmic description determining the importance of different relationships in a tumor-microenvironment, potentially revealing insights that a manually visual inspection by a pathologist cannot capture.

3 Problem Formulation.

3.1 Problem Statement: The main goal of this study is to build a representation learning model that distinguishes between point sets of different classes given multi-categorical point set data and its associated class labels for the point set classification. The primary objective is to achieve a high solution quality (e.g., overall accuracy). Fig. 3 shows two filed of view point set representations of MxIF images of two distinct classes of tumor-margin classification.

There are three key building blocks in constructing our model. The first is to build a neighborhood graph that is fed into a multi-scale position embedding to model the spatial distribution of one point and its neighbors (See Section 5.1). The second is to use the DGCNN to learn local information and a semantic representation by dynamically updating the neighborhood graph at each layer (see Section 5.2). The third is an attention network to distinguish between point pair instances that belong to different categories by learning an attention co-efficient and using an asymmetric function (e.g., average pool) to aggregate the information of one point and all its neighbors (see Section 5.3).

Background Knowledge: A spatial co-location is an intuitive representation to help understand the spatial interactions of a multi-categorical point set by identifying a subset of points frequently located in close spatial proximity to one another [14]. Spatial co-location interest measures, namely cross-k and participation ratio, commonly used in spatial data mining to quantify multi-categorical point sets [1].

4 Related Work

The success of convolutional neural networks (CNNs) in many pattern recognition tasks (e.g., [2, 3]) has inspired researchers to generalize convolution-like operations to directly apply to 2D/3D point cloud data without any computationally expensive intermediate conversion layers. PointNet [12], the first neural network architecture that directly applies to point cloud data, learns...
point features independently through several fully connected neural network layers and aggregates them using an asymmetric function operation (e.g., Max pooling). PointNet++ [14], a variation of PointNet accounts for the local structure by applying graph coarsening operation and a shared PointNet recursively to a set of local points chosen by farthest point sampling and subsequently their k-nearest neighbors. However, these techniques are limited in learning fine-grained local structures since learning representation of each point independently at a localized scale to preserve permutation invariance.

DGCNN proposes a graph dynamic graph CNN that dynamically updates the graph network at each layer by learning both local and global information [17]. This work is inspired by PointNet using a simple operation known as EdgeConv, where rather than independently applying to individual points, construct a locally connected neighborhood graph to exploit from both center nodes and edge features. Many other efforts have been made to learn local structure. For example, SpiderCNN [21] proposes a multi-scale hierarchical that extend convolutional operations from regular grids to irregular point sets that can be embedded to IR^n. PointCNN [9] proposes X-transformation that are learned from the input points in a local neighborhood, thus could potentially be more adaptive to local structures. However, these approaches are not designed to learn spatial relationships in multi-categorical point sets. In addition, they do not fully exploit the spatial distribution of points beyond simply measuring relative distance or applying a discretization or feed-forward neural network to coordinates [15, 7]. SRNet [10] is a neural network architecture designed to handle multi-categorical point set data, but it is limited to only learning binary spatial relationships. In addition, SRNet has assumed the importance of different category pairs to be equal and does not distinguish between the contribution of distinct point pair instances.

5 Multi-Category DGCNN.

We propose a multi-category dynamic graph CNN (MC-DGCNN, Fig. 4) inspired by DGCNN. The primary objective of the proposed neural network architecture is to learn N-way spatial relationships in a multi-category point set while preserving all properties of DGCNN (e.g., differentiability). The MC-DGCNN differs from DGCNN in two aspects: (1) it has a location representation layer, using a positional embedding module to learn the spatial distribution of a point and its neighbors in a locally connected neighborhood graph, and (2) it incorporates a point pair attention network, by learning the importance of point pairs in N-way spatial relationships based on their categorical attributes.

5.1 Location Representation Layer: Given a point set $P = \{p_i = (c_i, f_i)|p_1, ..., p_n\}$, where $c_i = (x_i, y_i)$ are the spatial coordinates and $f_i$ is the associated categorical attribute, we compute a locally connected graph $G = (V, E)$, where $V$ and $E$ are the vertices and edges, respectively. A graph can be represented as a symmetric adjacency matrix between all vertices, where $E = V \times V$. However, a full adjacency matrix has a high memory cost for our point set data, and we instead employ a $k$-nearest neighbor approach to build $G$ using the local neighborhood of each $c_i \in \mathbb{R}^2$, where $E = V \times K$.

The next step is to use position embedding to model the distribution of one point and its neighbor by only using spatial coordinates. The location representation layer is used to model the relative distance between a given point $c_i$ with the respect to its nearest points $c_j$, where $1 \leq j \leq k$, into a corresponding edge $e_{ij}$. The intuition behind the location representation layer is that spatial coordinates are illustrative location indicators; however, using them directly for downstream classification tasks is problematic. To this end, many efforts [15, 7] have been made to incorporate spatial information (e.g., GPS coordinates) by applying a discretization or feed-forward neural network to help improve related classification task performance. However, it was reported in [11] that such techniques are insufficient to capture the spatial distribution due to the lack of feature decomposition between spatial and categorical attributes. Inspired by the multi-scale periodic representation of grid cells in mammals [1] and vector representation of self-position [4], [11] proposed a multi-scale position encoder $PE$, which uses sine and cosine functions of different frequencies to encode positions in space.

Given a point $c_i$ in the studied 2D space, $e[c_i] = En_{\text{theory}}(c_i) = NN(PE(c_i))$, where $NN$ is fully connected ReLU layers and $PE(c_i)$ is a multi-scale representation $s_j, 1 \leq j \leq s$, to address the mixture distribution.
Figure 4: MC-DGCNN Architecture. The network architecture takes as input a multi-category point set containing n points, where a location representation layer calculates an embedding for each point using its coordinates and neighborhood spatial distribution. Position embeddings are then passed into the EdgeConv layer to specify an edge feature set of size k for each point. The categorical features are passed using a skip connection, where a point pair attention network calculates the importance between a point and its k nearest neighbors belonging to different category pairs. Lastly, an average pooling aggregates information from all k points to the center point. Notice that since a K-nn graph is built in the location representation layer, the reconstructing of the K-nn graph in the first EdgeConv layer is skipped.

Positional embeddings (PE) is as follows:

\[
PE_s(c_i) = \left[ PE_{s,1}(c_i); ...; PE_{s,s}(c_i) \right],
\]

\[
PE_{s,j}(c_i) = \left[ \cos\left( \frac{\langle c_i, a_j \rangle}{\lambda_{\min} \cdot g^{s/(S-1)}} \right); \sin\left( \frac{\langle c_i, a_j \rangle}{\lambda_{\min} \cdot g^{s/(S-1)}} \right) \right],
\]

where \( a_1 = [1, 0]^T, a_2 = [-1/2, \sqrt{3}/2]^T, a_3 = [-1/2, -\sqrt{3}/2]^T \) are unit vectors, the angles between every pair of vectors is \( 2\pi/3 \), \( \lambda_{\min}, \lambda_{\max} \) are the minimum and maximum grid scales, and \( g = \frac{\lambda_{\max}}{\lambda_{\min}} \). We define the input to PE as the distance between the center point \( c_i \) and its k-nearest neighbors \( c_j \), as \( PE(|c_i - c_j|) \), where \( 1 \leq j \leq k \).

5.2 DGCNN: DGCNN is a neural network with a simple EdgeConv operation to learn local information by building a locally connected graph and applying convolution along the edges connecting neighboring pairs of points \([17]\). Accordingly, the dynamic aspect of the DGCNN is that it dynamically update the network by constructing a new neighborhood graph after each pass through the hidden layer. Reconstructing the network graph in the embedding space produced by the hidden layer using nearest neighbors has been found to be empirically beneficial in related classification tasks, as shown in \([17]\).

The EdgeConv operation is defined as edge feature \( e_{ij} = h_{\Theta}(c_i, c_j) \), where \( h_{\Theta} : R^F \times R^F \rightarrow R^{F'} \) is a nonlinear function with a set of learnable parameters \( \Theta \). Lastly, an asymmetric operation (e.g., \( \sum \) or Max) is applied to aggregate information along all the edge features neighboring center node \( c_i \). The choice of \( h_{\Theta} \) is critical in defining EdgeConv, such as using the dot product between a set of filters \( \Theta = \{\theta_1, ..., \theta_M\} \) and image pixels \( x_j \) in a regular grid and aggregating information using \( \sum \) results in standard convolution. A detailed discussion on different forms of \( h_{\Theta} \) can be found in \([17]\).

We have adapted the EdgeConv operation from DGCNN in our network to learn both global shape structure, captured by the center coordinates \( c_i \), and local neighborhood information, captured by \( |c_i - c_j| \).
The overall formulation is as follows:

$$e'_{ij} = \text{leakyrelu}(\theta_m \cdot PE(|v_i - v_j|) + \phi_m, c_i),$$

(5.3)

where $\theta_m$ and $\phi_m$ are learnable parameter for local and global information, respectively. $PE$ is the positional embedding to represent relative distances along each edge starting at $v_i$.

### 5.3 Point Pair Attention Network:

Thus far, we have built the graph and defined edge embeddings in terms of strictly spatial features. If we followed DGCNN’s approach, we would simply concatenate the categorical features into the EdgeConv operations. However, the importance of interactions between vertices of categorical features $v_i$ and $v_j \in \mathcal{V}_i$ would not be learned in this way; the model would be limited to only learning individual category features. As a better inductive bias, the representation learning model should learn how to appropriately weight different point pair relationships. To this end, we propose a point pair attention layer to learn categorical pairwise interaction weights, followed by an average pooling layer to extend this concept to $N$-way spatial relationships. As a whole, this layer is analogous to a weighted average pooling function, where the weights correspond to the importance of the categorical interaction.

The input to this layer is an edge embedding $e''_{ij}$, which is the output from the EdgeConv layer. In the attention layer, we first derive $\hat{e}_{ij}$, an edge embedding augmented by the categorical pairwise attention coefficient:

$$\hat{e}_{ij} = \tilde{a}_{f_i, f_j} \cdot W \cdot e''_{ij}$$

(5.4)

where $W$ is a learnable linear transformation on the original embedding to aid attention expressivity, and $\tilde{a}_{f_i, f_j}$ is our learned pairwise attention coefficient vector for categorical point pair features $(f_i, f_j)$.

In this formulation, we have included $f_j \in \mathcal{V}_i$, where $\tilde{a}_{f_i, f_j}$ is a learned self-weighting based only on the categorical feature of $v_i$. We also note that interactions are assumed invariant with respect to the ordering of the categories; for example, $\alpha_{c_1, c_2} \equiv \alpha_{c_2, c_1}$.

Similar to other attention layers, we then apply a LeakyReLU (LR) activation, followed by a softmax function, resulting in the normalized attention coefficient:

$$\alpha_{f_i, f_j} = \frac{\exp(\text{LR}(\hat{e}_{ij}))}{\sum_{k \in \mathcal{V}_i} \exp(\text{LR}(\hat{e}_{ij}))},$$

(5.5)

where $\tilde{a}_{f_i, f_j} \in \mathbb{R}^K$ is the learned categorical pairwise attention coefficient for each neighbor. With this normalized attention coefficient, $\alpha_{f_i, f_j}$, we can calculate the weighted average pooling and produce the final vertex embedding:

$$\tilde{v}_i = \sigma \left( \frac{1}{|\mathcal{V}_i|} \sum_{j \in \mathcal{V}_i} \alpha_{f_j, f_j} \cdot W \cdot e''_{ij} \right),$$

(5.6)

This formulation can be extended to $K$ attention heads, following other attention networks such as GAT [16], where each attention head learns a separate categorical pairwise attention coefficient $\alpha_{f_i, f_j}^k$ and linear transformation weight $W^k$ followed by an aggregation operation (AGG) over the different attention head outputs:

$$\tilde{v}_i = \text{AGG}_K^{k=1} \sigma \left( \frac{1}{|\mathcal{V}_i|} \sum_{j \in \mathcal{V}_i} \alpha_{f_j, f_j}^k \cdot W^k \cdot e''_{ij} \right),$$

(5.7)

where $\sigma$ is a non-linear activation function such as LeakyReLU and the aggregation operation can take the form of an average or a concatenation.

Since this layer preserves the identity of the center vertex, it can also be extended to multiple layers of the network by maintaining the categorical features of vertices between layers with a skip connection. In this way, we do so, adding point pair attention after each layer’s EdgeConv operation. In the context of hierarchical feature learning, our network is therefore effectively capable of learning the importance of categorical $N$-way interactions in a hierarchical feature space.

Finally, we note that the choice of aggregation is not limited to average pooling; for example, one may choose to select a large number of k-nearest neighbors when building the graph, while taking only a top-$n$ subset of highest features to pool, in order to filter out an overpowering amount of weak interactions. Max pooling can be demonstrated as a special case of this concept, where only the top-1 of a neighbor’s features is selected.

### 6 Validation

We validated our proposed approach with (1) a comparative analysis to evaluate the proposed MC-DGCNN against state-of-the-art DNN point set classification methods, and (2) a sensitivity analysis to evaluate the impact of key element building blocks (i.e., location representation and point pair attention network) along with key parameters on selected performance metrics (e.g., classification accuracy).

#### 6.1 Dataset

Experiments were conducted on two real-world multi-category point datasets from MxIF images. The first dataset was used for two distinct classification tasks, (1) tumor-margin classification and (2) tumor-core classification. The second dataset was used for a (3) disease classification task.

In the tumor-margin classification task, we used 43
FOV point sets, 21 of which were labeled as disrupted margins by pathology analysis, and 22 labeled as intact. In the tumor-core classification task, we used 103 FOVs indicating two different clinical outcomes of ICI therapy, 30 of which were labeled as responders and 73 non-responders. In the disease classification task, we used 143 point sets of chronic pancreatitis (i.e., class 1) and 53 pancreatic ductal adenocarcinomas (PDAC) (i.e., class 2). In each classification task, we divided data into 80% training and 20% testing. Due to the limited number of learning samples, we leveraged a data augmentation technique to increase learning samples—whereby—each learning sample was rotated 12 degrees clockwise only five times during the training procedure. We restricted rotation to only five times due to potential overfitting issues. For each candidate method, 2,048, 1,024, and 2,048 points were uniformly sampled from each point set for the tumor-margin, tumor-core, and disease classification tasks, respectively.

6.2 Model Architecture: Fig. 4 shows the network architecture. The proposed MC-DGCNN was implemented in Pytorch, where the number of k nearest neighbors, the grid scale of location representation, the minimal grid cell size, and the maximal grid cell size were set to 6, 10, 1, and 100, respectively. We followed the same settings for the 4 EdgeConv layers, residual block connection, batch normalization, activation functions, and dropout as described in [17]. We used the Adam optimization algorithm with a learning rate of $10^{-4}$ and cross-entropy loss for 100 epochs to train the MC-DGCNN. The batch size was 6, and the momentum was 0.9. All other candidate methods tested with the same setting described above.

6.3 Experimental Results

Comparative Analysis: To validate the proposed model, we asked three questions: CA-1. How does the candidate methods compare on classification accuracy? CA-2. How does the candidate methods compare on F1 score? CA-3. Which candidate method achieves the best trade-off in both computational complexity and classification accuracy? We tested four candidate DNN architectures for our classification tasks:

- **PointNet** [13]: neural network architectures that directly consume point sets for applications ranging from object classification to part segmentation;
- **DGCNN** [17]: a dynamic neural network architecture integrated with a simple design of EdgeConv operation that is ideal for CNN-based high-level point cloud tasks such as classification and segmentation;
- **SRNet** [10]: a DNN architecture for binary multi-category point set data.
- **MC-DGCNN**: MC-DGCNN is tested using both PE and one-head point pair attention networks. This illustrates our entire proposed network architecture.

CA-1. We had candidate approaches perform on the three classification tasks described in Section 6.1 Table 1 and Fig. 5a show the superiority of MC-DGCNN over existing DNN competition (i.e., PointNet, DGCNN, SRNet). Most notably, we were able to improve accuracy over SRNet by a margin of 10.0%, 7.0%, and 4.0% on tumor-margin, tumor-core, and disease classifications, respectively.

CA-2. We also computed F1-score to validate the performance of our proposed with other candidate methods (e.g., PointNet, DGCNN, SRNet). F1-score metric is widely used to measure the performance of classification in imbalanced datasets. As shown in Table 2 and Fig. 5b again our MC-DGCNN has the best performance.

CA-3. Lastly, we used the forward pass time on three classification tasks as measure of the model’s computational time complexity and examined the trade-off between between time-complexity and classification accuracy for the four approaches. MC-DGCNN is not only more accurate than state-of-the-art SRNet, but it also runs faster from 12 to 68 times faster on the three classification tasks. Table 3 provides the detailed on each classification task.

It is evident that our proposed architecture significantly outperforms competing DNN point set classification methods (e.g., PointNet, DGCNN, and SRNet). In addition, these results suggest that positional embedding along with specifying different importance (i.e., attention co-efficient) to nodes of the same neighborhood with the different categorical attribute are beneficial.

| Model     | Overall accuracy |
|-----------|------------------|
|           | tumor-margin | tumor-core | disease  |
| PointNet  | 0.69         | 0.80       | 0.75     |
| DGCNN     | 0.78         | 0.81       | 0.73     |
| SRNet     | 0.88         | 0.92       | 0.88     |
| MC-DGCNN  | **0.98**     | **0.99**   | **0.92** |

| Model     | F1-Score |
|-----------|----------|
|           | tumor-margin | tumor-core | disease  |
| PointNet  | 0.69      | 0.76       | 0.71     |
| DGCNN     | 0.74      | 0.79       | 0.71     |
| SRNet     | 0.90      | 0.90       | 0.93     |
| MC-DGCNN  | **0.99**  | **0.98**   | **0.98** |
Sensitivity Analysis: To evaluate the impact of different building blocks in our proposed model, we asked 2 questions: SA-1. What is the effect of the position embedding? SA-2. What is the effect of a point pair attention network?

In addition, we also tested our proposed model in terms of key parameters on selected performance metrics (e.g., classification accuracy).

SA-1. To evaluate the effectiveness of the location representation layer for classification accuracy, we first tested MC-DGCNN using only position embedding along the edge. Table 4 shows that MC-DGCNN + PE provides competing results in many cases, but lack of point pair attention is evident. In addition, we tested PE in a new form, where we separately embed the center point and its k-nearest neighbor. Lastly, we computed the distance between the embedded space (i.e., $|PE(\text{center}) - PE(\text{neighbor})|$), and this was then fed into the EdgeConv layer. We refer to this new form of position embedding as $PE_{\text{sep}}$, where Table 4 shows that this new form is not as beneficial as embedding the distance between the center point and its neighbors. MC-DGCNN + PE/PE$_{\text{sep}}$ could be thought of as a generalizable form of DGCNN with position embedding (i.e., DGCNN + PE), where it significantly improves accuracy over DGCNN. Position embedding is only using spatial coordinates, and all categorical attributes have dropped.

SA-2. Next, we evaluated MC-DGCNN using only a one-head pair attention network without position embedding. Table 4 shows that the impact of the point pair attention network to distinguish between different points with different categories, which results in improving classification accuracy over existing DNN methods (i.e., PointNet, DGCNN). However, this result suggests that the proposed model performs the best when the location representation layer is integrated with the point pair attention network.

We further evaluated our proposed model by varying the key parameters, namely grid scale count, attention head, and k-nearest neighborhood.

As shown in Fig. 6a, the trends show that classification accuracy is sensitive to the choice of positional embedding representation, where increasing grid-scale does not guarantee better performance. While we did not thoroughly test all feasible grid-scale counts, but our intuition is modeling relative distance to an embedded space higher than that first multi-layer perceptron (i.e., EdgeConv) makes it challenging to approximate local information.

We also tested MC-DGCNN with a different number of attention heads, $H = \{1, 2, 4\}$, at each layer. As shown in Fig. 6b, our point pair convincingly learned point pair interactions between various categorical attributes. This also implies that compared to a multi-head attention network, our one-head attention network provides the best trade-off in model complexity, computational time complexity, and classification performance in terms of learning fewer parameters and taking less time.

Lastly, we tested our model with different sizes of k nearest neighborhoods. As shown in Fig. 6c, a large k neighbor size results in deteriorating the classification performance. It shows that beyond a certain threshold density, the locally connected neighborhood graph fails to approximate geodesic distance and destroys the geometry of each patch, as discussed in [17]. In addition, this confirms the hypothesis that a large size of k allows an overpowering amount of weak point pair interactions contributing to the overall representation of the center node. Hence, as discussed in Section 5.3, one may investigate more in a combination of different aggregation operations as the size of K increases.

7 Conclusion and Future Work.

In this paper, we propose MC-DGCNN, neural network architecture with location representation and point pair attention network. MC-DGCNN provides a promising way to help understand the spatial configuration of
multi-category point sets in N-way spatial relationships. Experimental evaluation shows that the proposed model significantly outperforms existing DNN techniques.

In the future, we plan to investigate a dynamic location representation layer to learn spatial distribution of embedded space between a given point and its neighbor. We also plan to identify a multi-category public benchmark dataset for a larger and broader evaluation of the proposed method. We plan to extend this work to consider spatial variability by learning an attention co-efficient based on density and distribution of multi-category points in different sub-regions.

Acknowledgments.

This material is based upon work supported by the NSF under Grants No. 2040459, 1737633, 1901099, 1218168, and 1916518. We also thank Kim Koffolt and the Spatial Computing Research Group for valuable comments and refinements.

References

[1] A. Abbott and E. Callaway. Nobel prize for decoding brain’s sense of place, Nature News. 514(7521):153, 2014.
[2] C. Cecotti et al. Grape detection with convolutional neural networks, Expert Systems with Applications., 2020.
[3] M. Farhadloo. Twitter Sentiment on Affordable Care Act using Score Embedding, arXiv preprint arXiv:1908.07061 2019.
[4] R. Gao et al. Learning grid cells as vector representation of self-position coupled with matrix representation of self-motion, ICRL., 2019.
[5] T. Gide et al. Close proximity of immune and tumor cells underlies response to anti-PD-1 based therapies in metastatic melanoma patients, Oncoimmunology., 2020.
[6] J. Golmohammadi, et al. An Introduction to Spatial Data Mining., The Geographic Information Science & Technology Body of Knowledge., 2020.
[7] G. Grace et al. Geo-aware networks for fine-grained recognition, International Conference on Computer Vision Workshops, 2019.
[8] K. Hargadon et al. Immune checkpoint blockade therapy for cancer: an overview of FDA-approved immune checkpoint inhibitors,International immunopharmacolgy., pp. 29–39, 2018.
[9] Y. Li et al. Pointcnn: Convolution on x-transformed points, Advances in neural information processing systems., pp. 820–830, 2018.
[10] Y. Li et al. SRNet: A spatial-relationship aware point-set classification method for multiplexed pathology images, (DeepSpatial ’21)., 2021.
[11] G. Mai et al. Multi-scale representation learning for spatial feature distributions using grid cells, ICLR., 2020.
[12] C. Qi et al. Pointnet: Deep learning on point sets for 3d classification and segmentation, IEEE international conference on computer vision and pattern recognition., pp. 652–660, 2018.
[13] C. Qi et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, arXiv preprint arXiv:1706.02413., 2017.
[14] S. Shekhar and Y. Huang. Discovering spatial co-location patterns: A summary of results, SSTD., pp. 236–256, 2001.
[15] K. Tang et al. Improving image classification with location context, IEEE international conference on computer vision., pp. 1008–1016, 2015.
[16] P. Velickovic et al. Graph attention networks, ICLR., 2018.
[17] Y. Wang et al. Dynamic graph cnn for learning on point clouds, Acm Transactions On Graphics (tog)., pp. 1–12, 2019.
[18] W. Wang et al. PD1 blockade reverses the suppression of melanoma antigen-specific CTL by CD4+ CD25Hi regulatory T cells, International immunology., pp. 1065–1077, 2009.
[19] A. Worrich et al. Associational effects in the microbial neighborhood, The ISME journal., pp. 2143–2149, 2019.
[20] T. Wu and Y. Dai Tumor microenvironment and therapeutic response, Cancer letters., pp. 61–68, 2017.
[21] Y. Xu et al. Spidercnn: Deep learning on point sets with parameterized convolutional filters, European Conference on Computer Vision., pp. 87–102, 2018.
help understand the spatial interactions of a multi-
categorical point set by identifying a subset of points
frequently located in close spatial proximity to one
another. Co-location patterns are commonly associated
with symbiotic interactions, such as the Nile Crocodile
and Egyptian Plover [14], which often reside in close
proximity. In the following, we describe two com-
monly used spatial statistics and spatial data mining
to understand the spatial relationships between multi-
categorical point sets.

**Cross-K Function:** Spatial statistics [6] uses the
cross-K function, a generalization of Ripley’s K func-
tion, to detect spatial relationships between point pat-
terns with more than one feature. The cross-K function
\( k(h) \) for binary spatial features is defined as
\[ K_{ij}(h) = \lambda_j^{-1} E|\# \text{ type } j \text{ instances within distance } h \]
of a randomly chosen type \( i \) instance], where \( i \) and \( j \)
represents two category types, \( \lambda_i \) is the density of
type \( j \) instances, \( h \) is the distance, and \( E|.| \) is the
expectation. The cross-k function could be estimated in
the form of
\[ K_{ij}(h) = \frac{1}{\lambda_j} \sum_{k} \sum_{l} d(i_k, j_l), \]
where \( d(i_k, j_l) \) is the distance between the \( i_k \) instance and
the \( j_l \) instance, \( I_k \) is an indicator function, and \( W \)
the study area [13]. The value of cross-k is a func-
tion of neighborhood distance \( h \), which implies the
spatial relationship between categorical points at
scales.

**Participation Index:** The co-location pattern inter-
est measure most related to the cross-k function is the
participation index. The participation index, an upper-
bound approximation of the cross-K function, possesses
an anti-monotone property that can be used for compu-
tational efficiency. Before defining participation index,
we need to define another interest measure, participation
ratio. The participation ratio \( Pr(C, f_i) \) of feature
\( f_i \) in a co-location pattern \( C = \{f_1, \ldots, f_K\}, 1 \leq i \leq K \)
is the fraction of spatial objects of feature \( f_i \) in the
neighborhood of instances of co-location C. Then, par-
ticipation index \( Pi(C) \) is defined as the minimum par-
ticipation ratio of the features in a co-location pattern,
that is \( Pi(C) = \min_{f_i \in C} \{ Pr(C, f_i) \} \).
The overall formulation of participation ratio is as
follows:

\[
Pr(C, f_i) = \frac{\text{Number of distinct } f_i \text{ in instances of } C}{\text{Number of } f_i}
\]

From equation above, it can be observed that the
value of the participation index is between 0 and 1.
A large \( Pi(C) \) value shows that events of \( f_i \) tend
to be located in close spatial proximity of other events
of features in \( C \). We used the cross-k function and
participation index to quantify the spatial relationship
between categorical point sets. For example, given a
point set containing points belonging to \( g \) categories
and a set of neighborhood distance threshold \( H =
\{h_1, \ldots, h_s\}, 1 \leq i \leq s \), there will be \( g(g-1) * s \) cross-
k functions or participation index pairs. Specifically,
we represent each point sets as a feature vector, \( v \),
of the cross-k function and participation index of a
co-location pattern containing two features with a
spatial neighborhood distance \( h = 50 \) pixels. This
feature vector is human-expert selected spatial features
that can be used for the down streaming two-class
classification problem.

For example, we used human-expert selected spatial
features derived from the spatial association measure
(e.g., participation index) to model intracellular inter-
action within each FOV. A simple decision tree clas-
sifier (See Fig. 7a) using 80% of data for training and
20% of data for testing provided 88% classification ac-
curacy. The decision tree uses a threshold line of 0.07
on the participation index “PI ((Helper T Cells, NK
Cells), distance = 50)” with the dashed black line
for 0.07 threshold.

**Figure 7:** (a). A simple decision tree to distinguish
between intact and disrupted in the tumor-margin
(learned from 34 learning sample FOVs) (b). Frequency
distribution of participation index “PI ((Helper T Cells,
NK Cells), distance = 50)” with the dashed black line
for 0.07 threshold.
tial relationships such as surrounded by, which may be of biological significance. In addition, these techniques mainly depend on co-location pairs (e.g., cell-1, cell-2) and do not consider the skew distribution of participating cells in spatial association calculation in a given FOV.