Abstract
Sparse incidence tensors can represent a variety of structured data. For example, we may represent attributed graphs using their node-node, node-edge, or edge-edge incidence matrices. In higher dimensions, incidence tensors can represent simplicial complexes and polytopes. In this paper, we formalize incidence tensors, analyze their structure, and present the family of equivariant networks that operate on them. We show that any incidence tensor decomposes into invariant subsets. This decomposition, in turn, leads to a decomposition of the corresponding equivariant linear maps. We characterize these linear maps as a combination of efficient pooling-and-broadcasting operations. We demonstrate the effectiveness of this family of networks by reporting state-of-the-art on graph learning tasks for many targets in the QM9 dataset.

1. Introduction
Many interesting data structures can be represented with sparse incidence tensors. For example, we can represent graphs using both node-node and node-edge sparse incidence matrices. We can extend this incidence representation to data defined on simplicial complexes and polytopes of arbitrary dimension, such as mesh, polygons, and polyhedra. The goal of this paper is to design deep models for these structures.

We represent an attributed geometric structure using its incidence tensor, which models the incidence pattern of its faces. For example, rows and columns in a node-edge incidence matrix are indexed by faces of size one (nodes) and two (edges). Moreover each edge (column) is incident to exactly two nodes (rows). The sparsity pattern of the incidence tensor has important information about the geometric structure. This is because sparsity preserving permutation of nodes often match the automorphism group of the geometric object; see Fig. 1(a,b).

We are interested in designing models that are informed by the symmetry of the underlying structure. We do so by making the model equivariant to symmetry transformations. When using the incidence tensor representation, a natural choice of symmetry transformations is the automorphism group of the geometric object. However, when working with a dataset comprising of different instances (e.g., different graphs or polyhedra), using individual automorphism groups is not practical. This is because each symmetry group dictates a different equivariant model, and we cannot train a single model on the whole dataset. A solution is to use the symmetric group (the group of all permutations of nodes) for all instances, which implicitly assumes a dense structure where all faces are present, e.g., all graphs are fully connected; see Fig. 1(c,d).

We show that under the action of the symmetric group, any incidence tensor decomposes into invariant subsets, or orbits, where each orbit corresponds to faces of particular size. For example, a node-node incidence matrix decomposes into: 1) diagonals, that can encode node attributes, and; 2) off-diagonals, corresponding to edge attributes. This is because any permutation of nodes, (i.e., simultaneous permutation of rows and columns) moves an (off-) diagonal entry to another (off-) diagonal entry in the node-node incidence. We can vectorize the diagonal and off-diagonal entries to get a node vector and an edge vector. These are examples of face-vectors in the general setting, and this example shows how and incidence tensor decomposes into face-vectors.

This decomposition into face-vectors also breaks up the design of equivariant linear maps for arbitrary incidence tensors into design of such maps between face-vectors of different size. We show that any such linear map can be written as a linear combination of efficient pooling-and-broadcasting operations. These equivariant linear maps replace the linear layer in a feedforward neural network to create an incidence network. We provide an extensive...
Incidence Networks for Geometric Deep Learning

2. Related Works

Deep learning with structured data is a very active area of research. Here, we briefly review some of the closely related works in graph learning and equivariant deep learning.

**Graph Learning.** The idea of graph neural networks goes back to the work of (Scarselli et al., 2009). More recently, Gilmer et al. (2017) introduced the message passing neural networks and showed that they subsume several other graph neural network architectures (Li et al., 2015; Duvenaud et al., 2015; Kearnes et al., 2016; Schütt et al., 2017), including the spectral methods that follows. Another body of work in graph deep learning extends convolution to graphs using the spectrum of the graph Laplacian (Bronstein et al., 2017; Bruna et al., 2014). While principled, in its complete form, the Fourier bases extracted from the Laplacian are instance-dependent and the lack of any parameter or function sharing across the graphs limits their generalization. Following (Henaff et al., 2015; Defferrard et al., 2016), Kipf & Welling (2016) propose a single-parameter simplification of spectral method that addresses this limitation and it is widely used in practice. Some notable extensions and related ideas include (Veličković et al., 2017; Hamilton et al., 2017; Xu et al., 2018; Zhang et al., 2018; Ying et al., 2018; Morris et al., 2018; Maron et al., 2019a).

**Equivariant Deep Learning.** Equivariance constrains the predictions of a model \( \phi : \mathbb{X} \rightarrow \mathbb{Y} \) under a group \( \mathcal{G} \) of transformations of the input, such that

\[
\phi(\pi \cdot x) = \pi \cdot \phi(x), \quad \forall x \in \mathbb{X}, \, \forall \pi \in \mathcal{G}. 
\]

Here \( \pi \cdot x \) is a consistently defined transformation of \( x \) parameterized by \( \pi \in \mathcal{G} \), while \( \pi \cdot \phi(x) \) denotes the corresponding transformation of the output. For example, in a convolution layer (LeCun et al., 1998), \( \mathcal{G} \) is the group of discrete translations, and (1) means that any translation of the input leads to the same translation of the output. When \( \phi : x \mapsto \sigma(Wx) \) is a standard feed-forward layer with parameter matrix \( W \), the equivariance property (1) enforces parameter-sharing in \( W \) (Shawe-Taylor, 1993; Ravanbakhsh et al., 2017).

Most relevant to our work, are equivariant models proposed for geometric deep learning that we review next. Covariant compositional network (Kondor et al., 2018) extends the message passing framework by considering basic tensor operations that preserve equivariance. While the resulting architecture can be quite general, it comes at the cost of efficiency. ¹ Hartford et al. (2018) propose a linear map equivariant to independent permutation of different dimensions of a tensor. Equivariant graph networks of (Maron et al., 2018) model the interactions within a set of nodes. We will further discuss this model as a special type of incidence network. These equivariant layers for interactions between and within sets are further generalized to multiple types of interactions in (Graham & Ravanbakhsh, 2019). Several recent works investigate the universality of such equivariant networks (Maron et al., 2019b; Keriven & Peyré, 2019; Chen et al., 2019). A flexible approach to equivariant and geometric deep learning where a global symmetry is lacking is proposed in (Cohen et al., 2019).

3. Graphs

In this section we discuss graphs and later generalize the arguments to a broader set of geometric objects in Section 5. Without loss of generality, we assume a fully-connected graph \( G = ([N], E) \in \mathbb{N} \times \mathbb{N} \), where \([N] = \{1, \ldots, N\}\) denotes a set of nodes, and \( E \) a set of edges.

¹Possibly due to the complexity of CCN architecture, experiments in (Kondor et al., 2018) do not use all attributes in QM9 dataset and their results are not comparable to state-of-the-art.
Alternatively, one can represent G with a node-edge incidence matrix $X_{δ_1,δ_2}$ indexed by $δ_1, δ_2 ∈ [N]$. In this representation, node and edge features are encoded as diagonal and off-diagonal entries of $X$, respectively. Here, we assume single input and output channel (i.e., scalar node and edge attributes) for simplicity; results trivially generalize to multiple channels.

Consider the group of all $N!$ permutations of nodes $δ_N$ and its action on $X$, which simultaneously permutes the rows and columns of $X$. Let $\mathcal{W} : \mathbb{R}^{N×N} → \mathbb{R}^{N×N}$ be a linear map equivariant to this action of $δ_N$,

$$\mathcal{W}(\pi X \pi^T) = \pi \mathcal{W}(X) \pi^T, \quad \forall \pi ∈ δ_N, \forall X. \quad (2)$$

The map $\mathcal{W}$ is constrained so that permuting the rows and columns of the input will have the same effect on the output. As shown in (Maron et al., 2018) this condition constrains the number of independent parameters in $\mathcal{W}$ to fifteen, regardless of the size of the graph.

Alternatively, one can represent $G$ with a node-edge incidence matrix $Y_{δ_1,\{δ_2, δ_3\}}$ where $δ_1 ∈ [N]$ labels nodes and the unordered pair $\{δ_2, δ_3\}$ with $δ_2, δ_3 ∈ [N]$ labels edges. $Y$ has a special sparsity pattern: $Y_{δ_1,\{δ_2, δ_3\}} ≠ 0$ iff node $δ_1$ is incident to the edge $\{δ_2, δ_3\}$. We identify this sparsity pattern implicitly by writing the matrix as $Y_{δ_1,\{δ_1, δ_2\}}$, so that we only index non-zero entries (note the repeated $δ_1$). $Y$ naturally encodes edge features at $Y_{δ_1,\{δ_1, δ_2\}}$ and $Y_{δ_2,\{δ_1, δ_2\}}$ for two different edge-directions of the edge $\{δ_1, δ_2\}$.

The action of $δ_N$ on $Y$ is also a simultaneous permutation of rows and columns, where the permutation of columns is defined by the action on the node pair that identifies each edge. $\pi \cdot \{δ_1, δ_2\} = \{π \cdot δ_1, π \cdot δ_2\}$. This action preserves the sparsity pattern of $Y$ defined above (note that even a fully connected graph has a sparse node-edge incidence matrix.) The maximal equivariant linear map acting on $Y$ is constrained to have seven independent parameters (assuming a single input and output channel). More alternatives beside node-node and node-edge representation for graph exist – e.g., one may use an edge-edge incidence matrix.

Since both $X$ and $Y$ represent the same graph $G$, and the corresponding linear maps are equivariant to the same group $δ_N$, one expects a relationship between the two representations and maps. This relationship is due to decomposition of $X$ into orbits under the action of $δ_N$. In particular, $X$ decomposes into two orbits: diagonal elements ($X_{\{δ_1\}}$) and off-diagonal elements ($X_{\{δ_1, δ_2\}}$), where each subset is invariant under the action of $δ_N$ – that is simultaneous permutations of rows and columns do not move a diagonal element to off-diagonal or vice-versa. We write this decomposition as

$$X = X_{\{δ_1\}} ⊥ X_{\{δ_1, δ_2\}}. \quad (3)$$

where the diagonal orbit is isomorphic to the vector of nodes $X_{\{δ_1\}}$ and the off-diagonal orbit is isomorphic to the vector of edges $X_{\{δ_1, δ_2\}}$ with $δ_1 ≠ δ_2$. Consider the map $\mathcal{W}$ defined above, in which both input and target decompose in this way. It follows that the map itself also decomposes into four maps

$$\mathcal{W}(X) = \bigcup_{m' = 1}^{2} \left( W^{1−m'}(X_{\{δ_1\}}) + W^{2−m'}(X_{\{δ_1, δ_2\}}) \right), \quad (4)$$

where $W^{m−m'}$ maps a face-vector of faces of size $m$ to a face-vector of faces of size $m'$. Equivariance to $δ_N$ for each of these maps constrains the number of independent parameters: $W^{1−1}$ is the equivariant layer used in DeepSets (Zaheer et al., 2017), and has two parameters. $W^{1−2}$ and $W^{2−1}$ each have three parameters, and $W^{2−2}$ has seven.
Incidence Networks for Geometric Deep Learning

unique parameter, and maps input edge features into target edge features. One key point is that the edge-vector $X_{\{d_1, d_2\}}$ is isomorphic to the node-edge incidence matrix $Y_{\delta_1, \delta_2}$ and thus the seven-parameters equivariant map for $Y$ is exactly $W_2^{-2}(X_{\{d_1, d_2\}})$ of (4). The parameter-sharing in two linear maps for the node-node and node-edge incidence matrices are visualized in Fig. 2 (left, middle). 2

![Parameter Sharing Diagram](image)

Figure 3: The 15 parameter equivariant map for node-node incidence matrix is decomposed into four maps between two orbits, diagonals and off-diagonals, where each orbit represents features of faces of a specific size. Each of these four maps performs pooling and broadcasting operations listed here: for example, the 3 parameter map from diagonals to off-diagonals is a linear combination of: (1) broadcasting the diagonal over rows; (2) broadcasting over columns, and; (3) first pooling the diagonal to get a scalar and broadcasting it over the off-diagonals. This approach generalizes: any $\delta_N$-equivariant linear map between face-vectors of any size can be written as a weighted sum of all “viable” pool-broadcast combinations.

Rather than using explicit parameter-sharing in $W$, we show that $W$ is a linear combination of pooling-and-broadcasting operations. In particular, the combination includes pooling the node indices of the input face-vector in “all possible ways”, and broadcasting the resulting collection of pooled vectors to the target face-vector, again in all ways possible. Each operation is associated with a learnable parameter; see Fig. 3 for the graph example.

3.2. Sparse Tensors and Non-Linear Layers

So far we have discussed equivariant linear layers for a fully connected graph. This means dense input/output node-node incidence $X$, or equivalently a node-edge incidence $Y$ with the sparsity pattern described in the previous section (which is maintained by the action of $\delta_N$). To avoid the cost of a dense representation, one may apply a sparsity mask after the linear map, while preserving equivariance:

$$W_{sp} : X \mapsto W(X) \circ s(X),$$

where $W$ is the equivariant linear map of Section 3.1, $s(X)$ is the sparsity mask, and $\circ$ is the Hadamard product. For example, assuming the layer output has the same shape as the input, we can choose to preserve the sparsity of the input. In this case, $s(X)$ will have zero entries where the input $X$ has zero entries, and ones otherwise. However, the setting of (5) is more general as input and output may have different shapes. Since the sparsity mask $s(X)$ depends on the input, the map of (5) is now non-linear. In practice, rather than calculating the dense output and applying the sparsity mask, we directly produce and store only the non-zero values. We consider this setup in our experiments, and observe competitive performance despite its significant computational advantage.

3.3. Further Relaxation of the Symmetry Group

The neural layers discussed so far are equivariant to the group $G = \delta_N$, where $N$ is the number of nodes. A simplifying alternative is to assume independent permutations of rows and columns of the node-node or node-edge matrix. This is particularly relevant for the node-edge matrix, where one can consider node and edges as two interacting sets of distinct objects. The corresponding $(\delta_N \times \delta_{N_2})$-equivariant layer, where $N_2$ is the number of edges, was introduced in (Hartford et al., 2018), has 4 unique parameters, and it is substantially easier to implement compared to the layers introduced so far. In Appendix A we show how to construct a sparsity-preserving (and therefore non-linear) layer for this case. Even though a single layer is over-constrained by these symmetry assumptions, we prove that two such layers generate exactly the same node and edge features as a single $\delta_N$-equivariant linear layer for a node-node incidence. These results are corroborated by comparable performance of this type of model in experiments.

4. Experiments

Many deep models for graphs have been applied to the task of predicting molecular properties (Gilmer et al., 2017; Schütt et al., 2018, 2017; Jørgensen et al., 2018; Morris et al., 2018; Unke & Meuwly, 2019; Kondor et al., 2018; Anderson et al., 2019); interestingly, most, if not all of these methods are considered message passing methods.3 Note that, in general, a drawback of a fully-fledged message passing scheme compared to incidence networks is its scalability. We test multiple models within our incidence networks framework on the QM9 dataset (Ramakrishnan et al., 2014)

3We were not able to compare our experimental results to (Morris et al., 2018; Maron et al., 2019a) and the results reported in (Wu et al., 2018) due to their choice of using a larger training split. Moreover, the raw QM9 dataset used by (Morris et al., 2018) contains 133,246 molecules, which has 639 fewer molecules than the dataset used in our experiments.
which contains 133,885 small organic molecules. Since the molecules are small, scalability is less of an issue in this context.

Our architecture for all models is a simple stack of equivariant layers:

$$\text{Pool}_{(0,1)} W^{(\ell)}(\text{ReLU}(W^{(\ell-1)} \ldots \text{ReLU}(W^{(1)}X)) \ldots),$$

where the final layer has a single channel followed by pooling, which produces a scalar value for the target. For details on the dataset, architecture, and training procedure, see Appendix F. 4

Table 1 reports previous state-of-the-art, as well as our results using various members of the incidence network family. The abbreviation used for the results include: Sparse (S); directed vs. undirected (D/D) and node-edge vs. node-node (E/E). For example, SD denotes sparse non-linear layers that operate on directed node-node incidence and produce directed asymmetric outputs. Finally (H) identifies the layer that uses the larger $\delta_N \times \delta_N$ symmetry. See Table 2 for more details on each incidence network model. The SD model is equivalent to the equivariant graph networks of (Maron et al., 2018).

All models match or outperform state-of-the-art in 7/12 targets (bold values). They also show a similar performance despite using different representations, supporting our theoretical analysis regarding the comparable expressiveness of node-node and node-edge representation. Dense models generally perform slightly better at the cost of 3× run-time for training. Finally, we note that the 4 parameter model (HSDE) of Section 3.3 performs almost as well, despite using an over-constraining symmetry group, further supporting our theoretical results outlined in Section 3.3 and explained in Appendix A.

### 5. Higher Order Geometric Structures

In this section we generalize the results of Section 3 to geometric structures beyond graphs. In Section 5.1 we provide a definition of incidence tensors, which generalize node-node and node-edge matrices of graphs. We discuss several examples, showing how they can represent geometric structures such as polytopes and simplicial complexes. In Section 5.2 we generalize the orbit decomposition of Section 3.1. Finally, in Section 5.3, we show how to build equivariant layers using a linear combination of simple pooling-and-broadcasting operations for arbitrary incidence tensors.

#### 5.1. Incidence Tensors

Recall that $[N] = \{1, \ldots, N\}$ denotes a set of nodes. A directed face of size $M$ is an ordered tuple of $M$ distinct
We have also denoted \( \delta \in [N]^M \) | \( \delta_i \neq \delta_j \forall i \neq j \). Following a similar logic, an undirected face \( \delta \subseteq [N] \), is a subset of \([N]\) of size \( M \). We use \( \delta^{(M)} \) when identifying the size of the face – i.e., \( |\delta^{(M)}| = M \). For example, \( \delta^{(2)} \) identifies an edge in a graph, or a mesh, while \( \delta^{(3)} \) is a triangle in a triangulated mesh.

An incidence tensor \( \mathbf{X}_{\delta_1, \ldots, \delta_D} \) is a tensor of order \( D \), where each dimension is indexed by all faces of size \( M_d = |\delta_d| \). For example, if \( \delta_1 = \{\delta_{1,1}\} \) indexes nodes, and \( \delta_2 = \{\delta_{2,1}, \delta_{2,2}\} \) identifies an edge, \( \mathbf{X}_{\delta_1, \delta_2} \) becomes a node-edge incidence matrix. An incidence tensor has a sparsity structure, identified by a set of constraints \( \Sigma = \{\sigma_1, \ldots, \sigma_C\} \), where all the indices \( \sigma_m, \ldots, \sigma_m \in \sigma_c \), \( \forall \sigma_c \in \Sigma \) are equal for any non-zero entry of \( \mathbf{X} \). For example, we have \( \mathbf{X}_{\{\delta_{1,1}\},\{\delta_{1,2},\delta_{2,2}\}} \neq 0 \), only if \( \delta_{1,1} = \delta_{2,1} \). Therefore \( \Sigma = \{\sigma_1 = \{\delta_{1,1}, \delta_{2,1}\}\} \).

While in general the pair \( (\mathbf{X}_{\delta_1, \ldots, \delta_D}, \Sigma) \) defines the incidence tensor, whenever it is clear from context, we will only use \( \mathbf{X}_{\delta_1, \ldots, \delta_D} \) to denote it. This formalism can represent a variety of different geometric structure as demonstrated in the following sections.

5.1.1. Simplicial Complexes

Before discussing general simplicial complexes let us review graphs as an example of incidence tensors.

The node-node incidence matrix is an incidence tensor \( \mathbf{X}_{\{\delta_1\},\{\delta_2\}} \) indexed by a pair of nodes, with no sparsity constraints. We denoted it with \( \mathbf{X}_{\delta_1, \delta_2} \) in Section 3.1 for simplicity. The node-edge incidence matrix is denoted by the pair \( (\mathbf{X}_{\{\delta_1\},\{\delta_2, \delta_3\}}, \{\delta_1, \delta_2\}) \). It is indexed by nodes \( \{\delta_1\} \) and edges \( \{\delta_2, \delta_3\} \). The entries can be non-zero only when \( \delta_1 = \delta_2 \), meaning that the edge \( \{\delta_1, \delta_3\} \) is adjacent to the node \( \{\delta_1\} \). An alternative notation is \( \mathbf{X}_{\{\delta_1\},\{\delta_1, \delta_2\}} \). Again, we denoted it simply with \( \mathbf{X}_{\delta_1, \{\delta_1, \delta_2\}} \) in Section 3.1. We have also denoted node and edge vectors with \( \mathbf{X}_{\{\delta_1\}} \) and \( \mathbf{X}_{\{\delta_1, \delta_2\}} \), respectively. As a final example, \( \mathbf{X}_{\{\delta_1, \delta_2\},\{\delta_1, \delta_3\}} \) would denote an edge-edge incidence matrix whose entries are non-zero wherever two edges are incident.

Let us now move to the definition of a general (undirected) simplicial complex. An abstract simplicial complex \( \Delta \subseteq 2^{[N]} \) is a collection of faces, closed under the operation of taking subsets – that is \( \{\delta_1 \in \Delta \text{ and } \delta_2 \subseteq \delta_1 \} \Rightarrow \delta_2 \in \Delta \). Dimension of a face \( \delta \) is its size minus one. Maximal faces are called facets and the dimension of \( \Delta \) is the dimension of its largest facet. For example, an undirected graph is a one-dimensional simplicial complex. Each dimension of an incidence tensor \( \mathbf{X}_{\delta_1, \ldots, \delta_D} \) may be indexed by faces of specific dimension. Two undirected faces of different dimension \( \delta, \delta' \in \Delta \) are incident if one is a subset of the other. This type of relationship as well as alternative definitions of incidence between faces of the same dimension can be easily accommodated in the form of inequality constraints in \( \Sigma \).

Although not widely used, a directed simplicial complex can be defined similarly. The main difference is that faces are sequences of the nodes, and \( \Delta \) is closed under the operation of taking a subsequence. As one might expect, the incidence tensor for directed simplicial complexes can be built using directed faces in our notation.

Example 1. A zero-dimensional simplicial complex is a set of points that we may represent using an incidence vector. At dimension one, we get undirected graphs, where faces of dimension one are the edges. Triangulated mesh is an example of two-dimensional simplicial complex; see figure below.

![Triangulated Mesh](image)

The triangular bi-pyramid of Fig. 1 is an example of 3-dimensional simplicial complex with 5 nodes, 9 edges, 7 faces of size 3, and two faces of size 4. The node-face incidence matrix in Fig. 1(a) is expressed by \( \mathbf{X}_{\{\delta_1\},\{\delta_1, \delta_2, \delta_3\}} \) in our formalism.

5.1.2. Polygons, Polyhedra, and Polytopes

Another family of geometric objects with incidence structure is polytopes. A formal definition of abstract polytope and its representation using incidence tensors is given in Appendix D. A polytope is a generalization of polygone and polyhedron to higher dimensions. The structure of an (abstract) polytope is encoded using a partially ordered set (poset) that is graded, i.e., each element of the poset has a rank. For example, Fig. 4, shows the poset for a cube, where each level is a different rank, and subsets in each level identify faces of different size (nodes, edges, and squares). The idea of using incidence tensor representation for a polytope, is similar to its use for simplicial complexes. Each dimension of \( \mathbf{X}_{\delta_1, \ldots, \delta_D} \) indexes faces of different rank. Two faces of the same dimension may be considered incident if they have a face of specific lower rank in common. We may also define two faces of different dimension incident if one face is a subset of the other – i.e., \( \delta^{(m)} < \delta^{(m')} \) in the partial order.

5.2. Symmetry & Decomposition

The automorphism group \( \text{Aut}(\mathbf{X}) \leq S_N \) associated with an incidence tensor is the set of all permutations of nodes that maps every face to another face, and therefore preserve the sparsity

\[
(\mathbf{X}_{\pi \delta_1, \ldots, \pi \delta_D} \neq 0 \iff \mathbf{X}_{\delta_1, \ldots, \delta_D} \neq 0) \iff \pi \in \text{Aut}(\mathbf{X})
\]
Figure 4: Representation of a cube as a (graded) partially ordered set. The incidence structure of the poset as well as face attributes is encoded in an incidence tensor.

where the action of \( \text{Aut}(X) \) on the faces is naturally defined as

\[
\pi \cdot (\delta_1, \ldots, \delta_M) = (\pi \cdot \delta_1, \ldots, \pi \cdot \delta_M). \tag{6}
\]

See Fig. 1(a,b) for an example. We may then construct \( \text{Aut}(X) \)-equivariant linear layers through parameter-sharing. However, the constraints on this linear operator varies if our dataset has incidence tensors with different sparsity patterns. For example, a directed graph dataset may be decomposed into a node-vector (the main diagonal of the adjacency cube), three edge-vectors (isomorphic to the three diagonal planes of the cube adjacency, with the main diagonal removed), and one hyper-edge-vector (isomorphic to the adjacency cube, where the main diagonal and diagonal planes have been removed). Here, \( \kappa_1 = \{3\} \), \( \kappa_2 = \{2\} \), and \( \kappa_3 = \{3\} \).

5.3. Equivariant Maps for Incidence Tensors

As shown in the previous section, any incidence tensor can be decomposed into disjoint union of face-vectors, that are invariant sets under the action of the symmetric group. An implication is that any equivariant map from an incidence tensor to another also decomposes into equivariant maps between face-vectors.

Let \( W^{M \rightarrow M'} \) be a linear function (here represented as a tensor) that maps a vector of faces of size \( M \) to a vector of faces of size \( M' \),

\[
W^{M \rightarrow M'} : \mathbb{R}^{N \times N \times \cdots \times N} \rightarrow \mathbb{R}^{N \times N \times \cdots \times N} \tag{8}
\]

\[
X(\delta_1, \ldots, \delta_M) \mapsto W^{\delta_1, \ldots, \delta_M}_{\delta'_1, \ldots, \delta'_M} X(\delta_1, \ldots, \delta_M),
\]

where \( \delta_1, \ldots, \delta_M \) identifies faces of size \( M \), and (using Einstein notation) repeated indices on are summed over. Equivariance to \( S_N \) is realized through a symmetry constraint on \( W \),

\[
W_{\pi\delta_1, \ldots, \pi\delta_M}^{\delta'_1, \ldots, \delta'_M} = W_{\delta'_1, \ldots, \delta'_M}^{\delta_1, \ldots, \delta_M}, \quad \forall \pi \in S_N, \tag{9}
\]

which ties the elements within each orbit of the so called diagonal \( S_N \)-action on \( W \); see Fig. 2 (left, middle) for a graph example.

5.3.1. Pool & Broadcast Interpretation

Each unique parameter in the constrained \( W \) corresponds to a linear operation that has a pool and broadcast interpretation – that is any linear equivariant map between two incidence tensors can be written as a linear combination of pooling- and broadcasting operations. Moreover, this interpretation allows for a linear-time implementation of the equivariant layers, as we avoid the explicit construction of \( W \).
Definition 1 (Pooling). Given a face vector \( \mathbf{X}_{\{\delta_1, \ldots, \delta_M\}} \), for \( \mathcal{P} = \{p_1, \ldots, p_L\} \subseteq [M] \), the pooling operation sums over the indices in \( \mathcal{P} \):

\[
\text{Pool}_\mathcal{P}(\mathbf{X}_{\{\delta_1, \ldots, \delta_M\}}) = \sum_{\delta_{p_1} \in [N]} \cdots \sum_{\delta_{p_L} \in [N]} \mathbf{X}_{\{\delta_1, \ldots, \delta_M\}},
\]

In practice, the summation in the definition may be replaced with any permutation-invariant aggregation function. We use mean-pooling in our graph experiments.

Definition 2 (Broadcasting). \( \text{Bcast}_{\mathcal{B}, M'}(\mathbf{X}) \) broadcasts \( \mathbf{X} \), a faces vector of size \( M \), over a target vector of faces of size \( M' \geq M \). We identify \( \mathbf{X} \) with a sequence of node indices of the target face-vector, \( \mathcal{B} = (b_1, \ldots, b_M) \) with \( b_m \in [M'] \), and we broadcast across the remaining \( M' - M \) node indices – that is

\[
(\text{Bcast}_{\mathcal{B}, M'}(\mathbf{X}))_{\{\delta_1, \ldots, \delta_{M'}\}} = \mathbf{X}_{\{\delta_1, \ldots, \delta_M\}}.
\]

For example, given an edge-vector \( \mathbf{X} = \mathbf{X}_{\{\delta_1, \delta_2\}} \), \( \text{Bcast}_{\{0,1\}, 3}(\mathbf{X}) \) broadcasts \( \mathbf{X} \) to a triangle-vector (i.e., vector of faces of size 3), where \( \mathbf{X} \) is mapped to the first two node indices and broadcasted along the third. Note that operations defined through pooling-and-broadcasting are equivariant to permutation of nodes. In fact, it turns out that an equivariant \( \mathcal{W} \) can only linearly combine pooling and broadcasting of input incidence tensor into an output tensor.

Theorem 5.2. Any equivariant linear map \( \mathcal{W}^{M \to M'} \) between face-vectors of size \( M \) and \( M' \), defined in (8), can be written as

\[
\mathcal{W}^{M \to M'}(\mathbf{X}) = \sum_{\mathcal{P} \subseteq [M]} \mathbf{w}_{\mathcal{B}, \mathcal{P}} \text{Bcast}_{\mathcal{B}, M'}(\text{Pool}_\mathcal{P}(\mathbf{X})).
\]

(10)

The proof appears in Appendix B. The sum of the pooling-and-broadcasting operations in (10) includes pooling the node indices of the input face-vector in all possible ways, and broadcasting the resulting collection of face-vectors to the target face-vector, again in all ways possible; \( \mathbf{w}_{\mathcal{B}, \mathcal{P}} \in \mathbb{R} \) is the parameter associated with each unique pooling-and-broadcasting combination. More details are discussed in Appendix C.

The number of operations in (10), is given by

\[
\tau^{M \to M'} = \min(M, M') \binom{M}{M'} (M')! m!.
\]

(11)

This counts the number of possible choices of \( m \) indices out of \( M \) input indices in (8) and \( m \) indices out of \( M' \) output indices to for pool and broadcast. Once this set is fixed there are \( m! \) different ways to match input indices to output indices.

5.3.2. Decomposition of Equivariant Maps

Let \( \mathcal{W} : (\bigcup_m \kappa_m \mathcal{X}_{\delta_m}) \rightarrow (\bigcup_m \kappa'_m \mathcal{X}_{\delta'_m}) \) be an equivariant map between arbitrary incidence tensors, where both input and output decompose according to (7). Using the equivariant maps \( \mathcal{W}^{m \to m'} \) of (10), we get a decomposition of \( \mathcal{W} \) into all possible combination of input-output face vectors

\[
\mathcal{W}(\mathcal{X}_{\delta_1, \ldots, \delta_D}, \Sigma) = \bigcup \bigcup \sum_{m} \sum_{k=1}^{\kappa'_m} \sum_{k'=1}^{\kappa'_m} \mathcal{W}^{k,m \to k',m'}(\mathcal{X}_{\delta_m}),
\]

(12)

where for each copy (out of \( \kappa'_m \) copies) of the output face of size \( m' \), we are summing over all the maps produced by different input faces having different multiplicities. Use of \( k \) and \( k' \) in the map \( \mathcal{W}^{k,m \to k',m'} \) is to indicate that for each input-output copy, the map \( \mathcal{W}^{m \to m'} \) uses a different set of parameters. The upshot is that input and output multiplicities \( \kappa, \kappa' \) play a role similar to input and output channels.

The total number of independent parameters in a layer is

\[
\tau = \sum_{m,m'} \kappa'_m \kappa_m \tau^{m \to m'},
\]

(13)

where \( \tau^{m \to m'} \) is given by (11).

Example 3 (Node-adjacency tensors). This example, is concerned with the incidence representation used in equivariant graph networks of (Maron et al., 2018) and derives their model as a special case, using our pooling-and-broadcasting layer and face-vectors decomposition. For an equivariant layer that maps a node-node-...-node incidence tensor \( \mathbf{X} \) of order \( D \) to the same structure, the decomposition in terms of face-vectors reads

\[
\mathbf{X}_{\delta_1, \ldots, \delta_D} = \bigcup \binom{D}{m} \mathbf{X}_{\delta_m},
\]

where \( \left\{ \begin{array}{c} D \\ m \end{array} \right\} \) is the Stirling number of the second kind; see Example 2. The total number of operations according to (13) is then given by

\[
\tau = \sum_{m,m' = 1}^{D} \binom{D}{m} \binom{D}{m'} \sum_{l=0}^{\min(m,m')} \binom{m}{l} \binom{m'}{l} l! = \text{Bell}(2D).
\]

In the last line, \( \text{Bell}(2D) \) is the Bell number and counts the number of unique partitions of a set of size \( 2D \). To see the logic in the final equality: first divide \( \left\{ \begin{array}{c} D \\ m \end{array} \right\} \) in
Figure 5: Decomposition of the 52 = 2 + 3 + 3 × 3 + 7 × 3 + 4 + 13 parameter equivariant map from a node-node incidence matrix to a node-node-node incidence tensor. This figure is similar to the example of Fig. 3, with the only difference that the output tensor has a higher rank. The input has two orbits and the output has five orbits described in Example 2. In our notation, the equivariant map from each orbit (face-vector) to another is \( W_{M \rightarrow M'} \) for \( M = 1, 2 \) and \( M' = 1, 2, 3 \). Each map \( W_{M \rightarrow M'} \) consists of different ways the source orbit can be pooled and broadcasted to the target orbit. For example, the 4 operations from a matrix diagonal to all the off-diagonals in the cube consists of broadcasting the diagonal over the cube in three different ways, plus an operation that pools over the diagonal and broadcasts the resulting scalar over the entire cube. The number of pool-broadcast operations \( \tau_{M \rightarrow M'} \) for each map is highlighted by the pink square, and the number agrees with (11).

Next, partition each half into subsets of different sizes \( (0 \leq m, m' \leq D) \) and choose \( l \) of these partitions from each half and merge them in pairs. The first two terms count the number of ways we can partition each half into \( m \) (or \( m' \)) partitions and select a subset of size \( l \) among them. The \( l! \) term accounts for different ways in which \( l \) partitions can be aligned. This result agrees with the result of (Maron et al., 2018). Therefore one may implement the hyper-graph networks using efficient pooling-and-broadcasting operations outlined in (10).

Recall that when discussing equivariant layers for graphs, we also considered independent permutations of rows and columns in a node-edge incidence matrix, and claimed that despite having only 4 parameters, stacking two such layers (with additional channels) is equivalent to the 15 parameter model. In Appendix E a similar result is given for higher dimensions, showing that one may use \( S_N^{1 \times \ldots \times D} \) as the symmetry group of an incidence tensor, where the equivariant model has \( 2^D \) parameters.

6. Conclusion and Future Work

This paper introduces a general approach to learning equivariant models for a large family of structured data through their incidence tensor representation. In particular, we showed various incidence tensor representations for graphs, simplicial complexes, and abstract polytopes.

The proposed family of incidence networks are 1) modular: they decompose to simple building blocks; 2) efficient: they all have linear-time pooling-and-broadcasting implementation, and; 3) effective: various members of this family achieve state-of-the-art performance for graphs using a simple architecture.

In our systematic study of this family, we discussed implications of 1) added symmetry due to undirected faces; 2) sparsity preserving equivariant maps, and; 3) the successive relaxation of the symmetry group \( \text{Aut}(X) \) ≤ \( S_N^{1 \times \ldots \times D} \) from the automorphism group, to a direct product of symmetric groups that independently permutes each dimension of the incidence tensor. Here, moving to a larger group simplifies the neural layer by reducing the number of unique parameters (and linear operations), while increasing its bias.

Application of incidence networks to different domains, such as learning on triangulated mesh, is a natural extension of this work. Note that for a comprehensive study with nodes, edges, and triangles, as they appear in triangulated mesh, one could investigate multiple order three representations (node-node-node, node-edge-face, etc), as well as some order two representations (node-face, face-face, etc.), or simply work with face-vectors. For each of these, one potentially needs a different incidence network using a different set of sparse pool-broadcast operations. We hope to explore this direction in a follow-up work.
Acknowledgements

We thank the reviewers for helpful comments on the manuscript. The computational resources for this work were mainly provided by Compute Canada. This research was in part supported by Canada CIFAR AI Chair Program and NSERC Discovery Grant.

References


