Lovász Convolutional Networks

Prateek Yadav
IISc Bangalore

Madhav Nimishakavi
IISc Bangalore

Naganand Yadati
IISc Bangalore

Shikhar Vashishth
IISc Bangalore

Arun Rajkumar
IIT Madras

Partha Talukdar
IISc Bangalore

Abstract

Semi-supervised learning on graph structured data has received significant attention with the recent introduction of Graph Convolution Networks (GCN). While traditional methods have focused on optimizing a loss augmented with Laplacian regularization framework, GCNs perform an implicit Laplacian type regularization to capture local graph structure. In this work, we propose *Lovász Convolutional Network* (LCNs) which are capable of incorporating global graph properties. LCNs achieve this by utilizing Lovász's orthonormal embeddings of the nodes. We analyse local and global properties of graphs and demonstrate settings where LCNs tend to work better than GCNs. We validate the proposed method on standard random graph models such as stochastic block models (SBM) and certain community structure based graphs where LCNs outperform GCNs and learn more intuitive embeddings. We also perform extensive binary and multi-class classification experiments on real world datasets to demonstrate LCN's effectiveness. In addition to simple graphs, we also demonstrate the use of LCNs on hyper-graphs by identifying settings where they are expected to work better than GCNs.

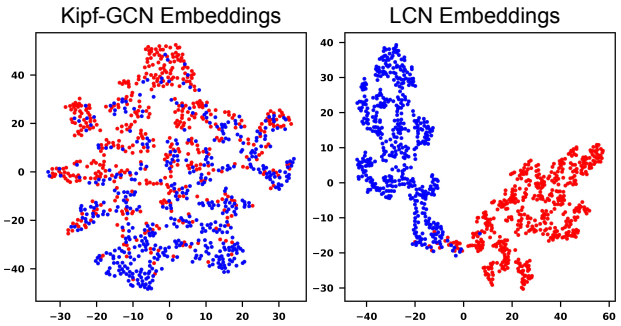
1 Introduction

Learning on structured data has received significant interest in recent years (Getoor and Taskar, 2007; Subramanya and Talukdar, 2014). Graphs are ubiquitous, several real world data-sets can be naturally represented as graphs; knowledge graphs (Suchanek et al., 2007; Auer et al., 2007; Bollacker et al., 2008), protein interaction graphs (Zitnik and Leskovec, 2017), social network graphs (Leskovec et al., 2010b,a; McAuley and Leskovec, 2012), citation networks (Giles et al., 1998; Lu and Getoor, 2003; Sen et al., 2008) to name a few. These graphs typically have a large number of nodes and manually labeling them as belonging to a certain class is often prohibitive in terms of resources needed. A common approach is to pose the classification problem as a semi-supervised graph transduction problem where one wishes to label all the nodes of a graph using the labels of a small subset of nodes.

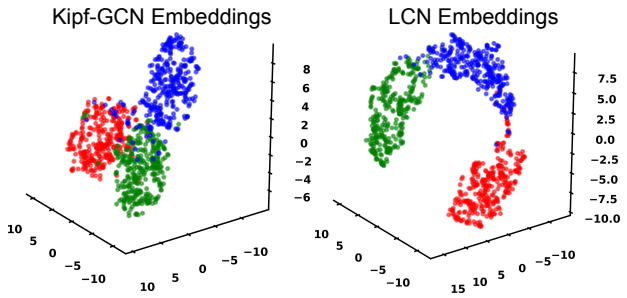
Recent approaches to the graph transduction problem rely on the assumption that the labels of nodes are related to the structure of the graph. A common approach is to use the Laplacian matrix associated with a graph as form of a structural regularizer for the learning problem. While the Laplacian regularization is done explicitly in (Agarwal, 2006; Zhu et al., 2003; Zhou et al., 2004; Belkin et al., 2006; Yang et al., 2016), more recent deep learning based Graph Convolution Network (GCN) approaches do an implicit Laplacian type regularization (Atwood and Towsley, 2016; Kipf and Welling, 2017; Li et al., 2018; Zhuang and Ma, 2018). While these traditional methods work reasonably well for several real world problems, our extensive experiments show that they may not be the best methods for tasks involving communities and there is a scope for significant improvement in such cases.

In this work, we propose a graph convolutional network based approach to solve the semi-supervised learning problem on graphs that typically have a community structure. An extensively studied model for communities is the Stochastic block model (SBM) which is a random graph model where the nodes of a graph exhibit community structure i.e., the nodes belonging to same community have a larger probability of having an edge between them than those in different communities. In this work, we propose the *Lovász Convolutional Network* (LCN) which, instead of the

Proceedings of the 22[nd] International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.



(a) Binary classification



(b) 3-class classification

Figure 1: Node embeddings for SBM Experiments. Note that these figures are obtained by projecting higher-dimensional embeddings to lower-dimensional space using t-sne (van der Maaten and Hinton, 2008).

traditional Laplacian, uses the embeddings of nodes that arise from Lovász's orthogonal representations as an implicit regularizer. The Lovász regularization, as we will see, is tightly coupled to the coloring of the complement graph of a given graph and hence often produce remarkably superior embeddings than those obtained using the Laplacian regularization for graphs which have a community structure. Intuitively, the optimal coloring of the complement of a graph can be viewed as a way to associate same color to nodes belonging to a same community. As Lovász embeddings also tend to embed nodes with same colors to similar points in Euclidean space, the proposed model performs much well in practice. Figure 1a and Figure 1b illustrate this phenomenon using examples for a binary and a three class classification problem where the graph is generated using a stochastic block model. As can be seen, the average distance between embeddings learnt using the LCN is much better than using traditional graph based convolution networks. We make the following contributions in this work:

- We propose the *Lovász Convolutional Network (LCN)* for the problem of semi-supervised learning on graphs. LCN combines the power of using the Lovász embeddings with GCNs.
- We analyze various types of graphs and identify the classes of graphs where LCN performs much better than existing methods. In particular, we demonstrate that by keeping the optimal coloring, a global property of the graph, fixed and increasing the number edges to the graph, LCNs outperforms traditional GCNs.
- We carry out extensive experiments on both synthetic and real world datasets and show significant improvement using LCNs than state of the art algorithms for semi-supervised graph transduction.

The Source code for our model can be found at https://github.com/malllabiisc/lcn.

2 Related Work

The work that is most closely related to ours is (Shivanna et al., 2015) which proposes a spectral regularized orthogonal embedding method for graph transduction (SPORE). While they use a Lovász embedding based kernel for explicit regularization, the focus is on computing the embedding efficiently using a special purpose optimization routine. Our work on the other hand proposes a deep learning based Lovász convolutional network which differs from the traditional loss plus explicit regularizer approach of (Shivanna et al., 2015) and our experimental results confirm that the proposed LCN approach performs significantly better than SPORE. The use of explicit Laplacian regularizer for semi-supervised learning problems on graphs has been explored in (Ando and Zhang, 2007; Agarwal, 2006), where the focus is to derive generalization bounds for learning on graphs. However, as we will discuss in the sequel, there are settings where Lovász embeddings are more natural in capturing the global property of graphs than the Laplacian embeddings and this reflects in our experimental results as well. More recently (Zhuang and Ma, 2018) propose a dual convolution approach to capture global graph property using positive pointwise mutual information (PPMI). We differ from this approach in defining global property in terms of coloring of the complement graph as opposed to computing semantic similarity using random walks on the graph as done in (Zhuang and Ma, 2018). Lovász based kernels for graphs have been explored in the context of other machine learning problems such as clustering in (Johansson et al., 2014). Jethava et al. (2013) show an interesting connection between Lovász ϑ function and one class SVMs.

There has been considerable amount of work in extending well established deep learning architectures for graphs. Bruna et al. (2014); Henaff et al. (2015); Duvenaud et al. (2015); Defferrard et al. (2016) extend Convolutional Neural Networks (CNN) for graphs, while Jain et al. (2016) propose Recurrent Neural Net-

works (RNN) for graphs. Kipf and Welling (2017) propose Graph Convolutional Networks which achieve promising results for the problem of semi-supervised classification on graphs. Most recently, a faster version of GCN, for inductive learning on graphs, has been proposed by (Chen et al., 2018). An extension to GCNs based on graph partition is proposed recently by (Liao et al., 2018). Recently, GCNs with confidence scores for embeddings has been proposed by (Vashishth et al., 2019). GCNs have been shown to be effective for several tasks Marcheggiani and Titov (2017); Vashishth et al. (2018a,b,c); Ray et al. (2018). However, as we show in our experiments, there are several natural settings where the proposed LCN performs much better than the state of the art GCNs in various problems of interest.

3 Problem Setting and Preliminaries

We work in the semi-supervised graph transduction setting where we are given a graph $G(V, E)$, where V denotes the set of vertices with cardinality n and E is the edge set. We are given the labels ($\{0, 1\}$ in the case of binary classification) of a subset of nodes ($m < n$) of V and the goal is to predict the labels of the remaining nodes as accurately as possible. Given a graph $G(V, E)$, $\alpha(G)$ denotes the maximum independence number of the graph i.e., the size of the set containing the maximum number of non-adjacent nodes in G. A coloring of G corresponds to an assignment of colors to nodes of the graph such that no two nodes with the same color have an edge between them. $\chi(G)$ denotes the chromatic number of G which is the minimum number of colors needed to color G. We denote the complement of a graph by $\bar{G}(V, \bar{E})$. An edge (u, v) is present in $\bar{G}$ if and only if it is not present in G. It is easily seen that for any graph G, $\alpha(G) \leq \chi(\bar{G})$. A clique is a fully connected graph which has edges between all pairs of nodes. We assume that there is a natural underlying manner in which the graph structure is related to the class labels. In what follows, we recall certain classes of graphs and a certain type of graph embedding which will be of interest in the rest of the paper.

SBM Graphs: The Stochastic Block Model (SBM) (Holland et al., 1983; Condon and Karp, 1999) is a generative model for random graphs. They are a generalization of the Erdos-Renyi graphs where the edges between nodes of the same community are chosen with a certain probability (p) while the edges across communities are chosen with a certain other probability (q where $q < p$). SBMs tend to have community structure and hence are used to model several applications including protein interactions, social network analysis and have been extensively studied in machine learning, statistics, theoretical computer science and network science literature.

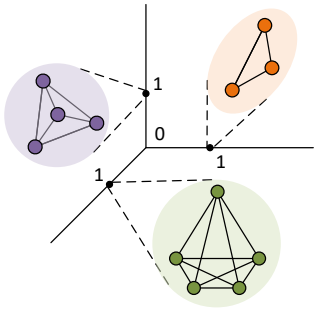


Figure 2: Lovász embeddings for a graph consisting of set cliques are mapped orthogonal dimensions. Refer Section 3 for more details.

Perfect Graphs: Perfect graphs are a class of graphs whose chromatic number $\chi(G)$ equals the size of the largest clique for every induced subgraph. Several important class of graphs including bipartite graphs, interval graphs, chordal graphs, caveman graphs etc. are all perfect graphs. We refer to (Ramírez-Alfonsín and Reed, 2001) for graph theoretical analysis of perfect graphs. Our interest in these graphs is due to the fact that the the chromatic number of these graphs can be computed in polynomial time (Lovász, 2009) and they coincide with the Lovász ϑ number of the graph which we discuss next.

Lovász Embeddings: Lovász (Lovász, 1979) introduced the concept of *orthogonal embedding* in the context of the problem of embedding a graph $G = (V, E)$ on a unit sphere $\mathcal{S}^{d-1}$.

Definition 3.1 (Orthogonal embedding (Lovász, 1979; Lovász and Vesztergombi, 1999))**.** An orthogonal embedding of a graph $G(V, E)$ with $|V| = n$, is a matrix $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_n] \in \mathbb{R}^{d \times n}$ such that $\mathbf{u}_i^\top \mathbf{u}_j = 0$ whenever $(i, j) \notin E$ and $\mathbf{u}_i \in \mathcal{S}^{d-1}$ $\forall i \in [n]$.

Let $Lab(G)$ denote the set of all possible orthogonal embeddings of the graph G, given by $Lab(G) = \{\mathbf{U} | \mathbf{U}$ is an orthogonal embedding$\}$. The Lovász theta function is defined as:

$$\vartheta(G) = \min_{\mathbf{U} \in Lab(G)} \min_{\mathbf{c} \in \mathcal{S}^{d-1}} \max_i (\mathbf{c}\top \mathbf{u}_i)^{-2}.$$

The famous sandwich theorem of Lovász (Lovász, 1979) states that $\alpha(G) \leq \vartheta(G) \leq \chi(\bar{G})$, where $\alpha(G)$ is the independence number of the graph and $\chi(\bar{G})$ is the chromatic number of the complement of G. Perfect graphs are of interest to us as both the above inequalities are equalities for them (Lovász, 2009).

A few examples are helpful to gain intuition about the relation of Lovász embeddings to community structures. For a complete graph, the complement can be colored using just one color, the Lovász embeddings of all the nodes are trivial and in 1-dimension. These embeddings are exactly the same as there are no orthogonal

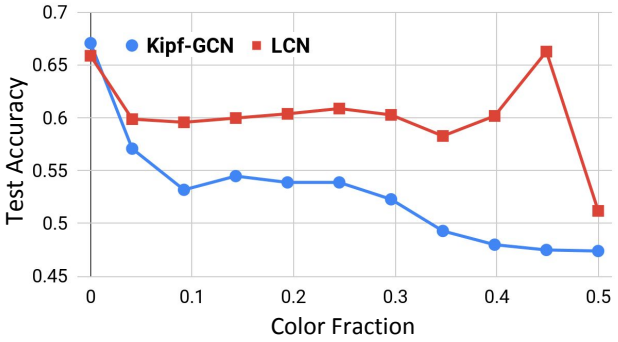


Figure 3: Variation of test accuracy (higher is better) for GCN and LCN- with variation in the graph structure. GCN fails to perform as the number color fraction increases. Refer Section 4 for more details.

constraints imposed by the edges. As a generalization of this example (Figure 2), for a graph that is a disjoint union of k cliques of possible variable number of nodes in each clique, the complement is a complete k partite graph and hence can be colored using k colors where each partition corresponds to a single color. It turns out the Lovász embeddings for this graph are a set of orthonormal vectors in $\mathbb{R}^k$. In practice, the communities that occur are not exactly cliques i.e., not all edges in a community are connected to each other. However, the Lovász embeddings still capture the necessary structure as we will see in our experiments.

Graph convolutional networks (GCN): GCNs (Kipf and Welling, 2017) extend the idea of Convolutional Neural Networks (CNNs) for graphs. Let $G(V, E)$ be an undirected graph with adjacency matrix $\mathbf{A}$ and let $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ be the adjacency with added self-connections and $\widetilde{\mathbf{D}}_{ii} = \sum_j \widetilde{\mathbf{A}}_{ij}$. Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ represent the input feature matrix of the nodes. A simple two-layer GCN for the problem of semi-supervised node classification assumes the form : $f(\mathbf{X}, \mathbf{A}) = \text{softmax}(\hat{\mathbf{A}}\ \text{ReLU}(\hat{\mathbf{A}}\mathbf{X}\mathbf{W}^{(0)})\mathbf{W}^{(1)})$. Where, $\hat{\mathbf{A}} = \widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}$, $\mathbf{W}^{(0)} \in \mathbb{R}^{d \times h}$ is an input-to-hidden weight matrix for a hidden layer with h units and $\mathbf{W}^{(1)} \in \mathbb{R}^{h \times F}$ is hidden-to-output weight matrix. The softmax activation function, defined as $\text{softmax}(x_i) = \frac{1}{Z}\exp(x_i)$ with $Z = \sum_i \exp(x_i)$ is applied row-wise.

For semi-supervised multi-class classification, cross-entropy loss over the labeled examples is given by

$$\mathcal{L} = \sum_{l \in \mathcal{Y}_L} \sum_{f=1}^{F} Y_{lf} \ln Z_{lf}, \tag{1}$$

where, $\mathcal{Y}_L$ is the set of labeled nodes. The weights $\mathbf{W}^{(0)}$ and $\mathbf{W}^{(1)}$ are learnt using gradient descent.

4 Motivating Example

In this section we present a motivating example to demonstrate the use of the Lovász orthogonal embeddings in the semi-supervised graph transduction task. In particular, we want to show how the embeddings learnt using the Lovász kernel results in improved accuracy as a parameter called *coloring fraction*, which we define below, varies. To illustrate our hypothesis, we consider a bipartite graph as input to the problem. The reason for this choice is that bipartite graphs are perfect and hence optimal coloring of both the graph G and its complement $\bar{G}$ (which is also perfect by the perfect graph theorem (Chudnovsky et al., 2006)) are easy to compute in polynomial time. Before explaining the experiment, we start with the following definition.

Coloring Fraction: Given a graph $G = (V, E)$, consider the optimal coloring of the complement graph $\bar{G}$. According to this coloring scheme of the nodes, let n_d represent the number of edges in G such that the pair of nodes each edge connects have different colors. And let n_t represent the total number of pairs of nodes in G such that the nodes in each pair have different colors. Then the *coloring fraction* is defined as n_d/n_t.

As an example, for a complete bipartite graph $G = K(n, n)$ on $2n$ nodes, the complement graph is the union of 2 disjoint cliques of n nodes each and hence the graph can be colored using n colors. The coloring fraction is then $\frac{n(n-1)}{2(n-1)(n)} = 0.5$. The following proposition establishes how coloring fraction varies with removal of edges from a graph.

Proposition 1. *Let $G(V, E)$ be a graph where $\chi(\bar{G})$ is the chromatic number of the complement of G. Let $\beta(G)$ be the coloring fraction of G. Let G' be the graph obtained from G by removing a set of edges whose nodes have different colors with respect to the optimal coloring of $\bar{G}$. Then $\chi(\bar{G'}) = \chi(\bar{G})$ whereas $\beta(G') < \beta(G)$.*

Proof. It should be observed that the optimal coloring for $\bar{G}$ is also a *valid* coloring of $\bar{G'}$ as the edges removed from G are only from nodes with different colors with respect to coloring of $\bar{G}$. To see why it is also an *optimal* coloring, we use contradiction. If there exists a coloring of $\bar{G'}$ with strictly smaller number of colors than $\chi(\bar{G})$, then we can remove edges to form $\bar{G'}$ to obtain $\bar{G}$ such that it is also a valid coloring of $\bar{G}$ as removing edges does not affect the validity of a coloring. However, this contradicts the optimality of the original coloring for $\bar{G}$. Thus $\chi(\bar{G}) = \chi(\bar{G'})$. Moreover, since we are removing edges from G, the coloring fraction increases by definition and hence $\beta(G') < \beta(G)$. □

The above proposition says that by removing edges carefully, a local property of the graph (coloring fraction) changes whereas a global property (chromatic number of complement graph) does not change. If the labels of nodes depends on the global property of the

Algorithm 1: Lovasz Kernel Matrix Computation

Input: $\mathbf{A}$, Adjacency matrix of Graph G
Output: $\mathbf{K}$: Lovasz Kernel
$[SDP]\mathbf{Y} \leftarrow$minimize t, subject to:

$$\mathbf{Y} \succeq 0,\ \mathbf{Y}_{ij} = -1,\ \forall (i,j) \notin E,\ \mathbf{Y}_{ii} = t - 1$$

$\mathbf{P} \in \mathbb{R}^{n \times n} \leftarrow Cholesky(\mathbf{Y})$;
if $rank(\mathbf{P}) < n$ **then**
  $c \leftarrow$ random basis element from $Null(\mathbf{P})$;
  $u_i = \frac{c+p_i}{\sqrt{t}}$ where p_i is i^{th} column of $\mathbf{P}$;
end
if $rank(\mathbf{P}) = n$ **then**
  $p_i = [p_i\ 0] \in \mathbb{R}^{n+1}\ \forall\ i \in \{1, 2, \ldots, n\}$;
  $u_i = \frac{e_{n+1}+p_i}{\sqrt{t}}$ where $e_{n+1} \in \mathbb{R}^{n+1}$ is the standard basis element;
end
$\mathbf{U} = [u_1 u_2 \ldots u_n]$;
$\mathbf{K} = \mathbf{U}^\top \mathbf{U}$;

graph, then a natural question of interest is to study the sensitivity of algorithms to change in the local property while keeping the global property fixed. This is precisely what we do as we explain below.

We begin with a complete bipartite graph $K(n, n)$ whose coloring fraction as computed above is 0.5. We remove m edges in each step where the nodes of removed edges have different colors (w.r.t optimal coloring of $\bar{G}$). In each case, the labels are assigned such that nodes with half the colors are assigned to class 0 and remaining to class 1. We compute the accuracy of a Laplacian based GCN model vs the proposed LCN model. In our experiment we set $n = 50$ and $m = 250$. The results averaged over 10 random splits of $20\% - 20\% - 60\%$ train-validation-test are presented in Figure 3. It is clear that as the color fraction increases, the accuracy of the standard GCN drops while that of Lovász does not. This is because the standard GCN depends on local connectivity property of the graph whereas the orthogonal labeling is done in accordance to the global coloring of the complement graph and is better captured by the proposed LCN.

The above example motivates our study of Lovász based embeddings in cases where the global structure of the graph is related to the class labels. With this motivation, we propose the Lovász convolution network in the following section.

5 LCN: Proposed Model

In this section, we present our proposed method, the Lovász Convolution Network (LCN), for semi-supervised graph transduction. As motivated in the previous section, when the class labels depend on the coloring (a global property) of the given graph, it is natural to start training a graph based convolution network which incorporates this property into learning. Let $Lab(G)$, as defined in Section 3, represent the set of all possible orthonormal embeddings for a given graph G. The set of graph kernel matrices is defined as

$$\mathcal{K}(G) \coloneqq \{\mathbf{K} \in \mathcal{S}_n^+ | \mathbf{K}_{ii} = 1, \forall i \in [n]; \mathbf{K}_{ij} = 0. \forall (i,j) \notin E\},$$

where $\mathcal{S}_n^+$ is the set of all positive semidefinite matrices. Jethava et al. (2013) showed the equivalence between $Lab(G)$ and $\mathcal{K}(G)$. Since $\mathbf{K} \in \mathcal{K}(G)$ is positive semidefinite, there exists a $\mathbf{U} \in \mathbb{R}^{d \times n}$ such that $\mathbf{K} = \mathbf{U}^\top \mathbf{U}$. It should be noted that $\mathbf{K}_{ij} = \mathbf{u}_i^\top \mathbf{u}_j$, where $\mathbf{u}_i$ is the i-th column of $\mathbf{U}$, which implies $\mathbf{U} \in Lab(G)$. Similarly, it can be shown that for any $\mathbf{U} \in Lab(G)$, $\mathbf{K} = \mathbf{U}^\top \mathbf{U} \in \mathcal{K}(G)$. Given a graph G, we follow the procedure described in (Lovász and Vesztergombi, 1999, Proposition 9.2.9) for computing the Lovász orthonormal embedding $\mathbf{U}$ and the associated kernel matrix $\mathbf{K}$ optimally. The procedure is summarized in Algorithm 1. Similar to the normalized Laplacian of a graph, the kernel matrix is also positive semidefinite.

The kernel computation explained in Algorithm 1 requires solving a Semi Definite Program (SDP), the computational complexity of which is $O(n^6)$. This becomes a huge bottle-neck for large scale datasets. Therefore, for large scale datasets, we exploit the following characterization of $\vartheta(G)$ given by Luz and Schrijver (2005):

Theorem 5.1 (Luz and Schrijver (2005))**.** *For a graph $G = (V, E)$ with $|V| = n$, and let $\mathbf{C} \in \mathbb{R}^{n \times n}$ be any non-null symmetric matrix with $\mathbf{C}_{ij} = 0$ whenever $(i, j) \notin E$. Then,*

$$\vartheta(G) = \min_{\mathbf{C}} \nu(G, \mathbf{C}), \quad \textit{where}$$

$$\nu(G, \mathbf{C}) = \max_{\mathbf{x} \geq 0} 2\mathbf{x}^\top \mathbf{e} - \mathbf{x}^\top \left(\frac{\mathbf{C}}{-\lambda_{\min}(\mathbf{C})} + \mathbf{I}\right)\mathbf{x},$$

where $\mathbf{e} = [1, 1, \ldots, 1]^\top$ and $\lambda_{\min}(\mathbf{C})$ is the minimum eigen value of $\mathbf{C}$.

Note that the matrix $\mathbf{K}_{LS} = \frac{\mathbf{A}}{-\lambda_{\min}(\mathbf{A})} + \mathbf{I}$ obtained by fixing $\mathbf{C} = \mathbf{A}$ in Theorem 5.1 is positive semidefinite. Therefore, there exists a labeling $\mathbf{U} \in \mathbb{R}^{d \times n}$ such that $\mathbf{U}^\top \mathbf{U} = \mathbf{K}_{LS}$, which is referred to as LS labeling (Jethava et al., 2013). From Theorem 5.1, for any graph G, we have

$$\vartheta(G) \leq \nu(G, \mathbf{A}),$$

an upper bound on $\vartheta(G)$, and the equality holds for a class of graphs called $\mathcal{Q}$ graphs (Luz, 1995). Computation of $\mathbf{K}_{LS}$ has a complexity of only $O(n^3)$, hence for large scale datasets we approximate the Lovász kernel by $\mathbf{K} = \mathbf{K}_{LS}$.

We propose to use the following two layered architecture for the problem of semi-supervised classification,

$$f(\mathbf{X}, \mathbf{K}) = \text{softmax}(\mathbf{K}\ \text{ReLU}(\mathbf{K}\mathbf{X}\mathbf{W}^{(0)})\mathbf{W}^{(1)}). \quad (2)$$

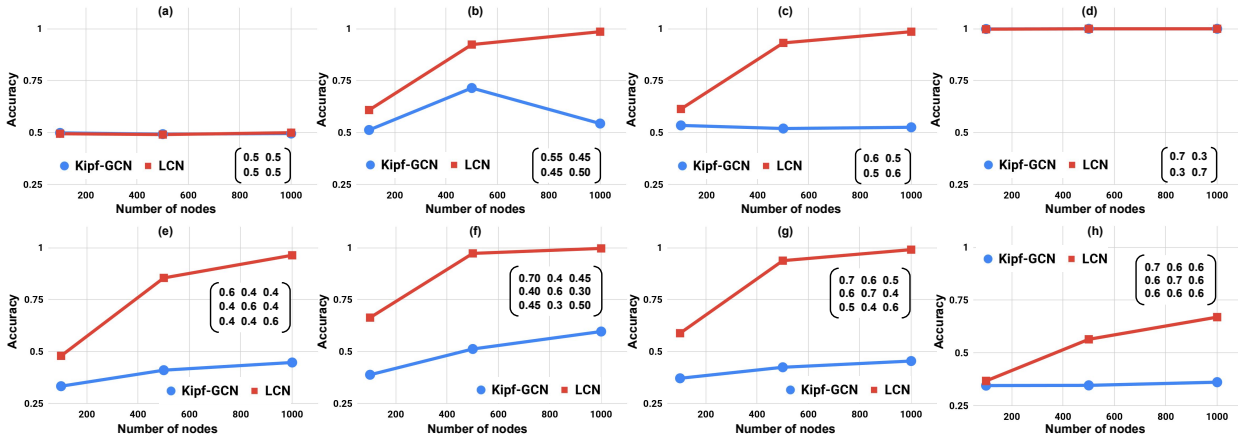


Figure 4: Test accuracy plots for various synthetically generated graphs from stochastic block model. The matrix in plot denotes the connection probabilities between classes.

Similar to GCN, we minimize the cross-entropy loss given in Equation (1) for semi-supervised multi-class classification. We use batch gradient descent for learning the weights $\mathbf{W}^{(0)}$ and $\mathbf{W}^{(1)}$.
We note that when the class labels are a non-linear mapping of the optimal coloring of $\bar{G}$, LCN with Lovász kernel $\mathcal{K}(G)$ tunes the weights of the network to learn the mapping.

6 Experimental Results

In this section, we report the results of our experiments on several synthetic and real world datasets. We demonstrate the usefulness of the embeddings learnt using the Lovász convolution networks over several state of the art methods including GCNs, SPORE, normalized and unnormalized laplacian based regularization along with other embeddings such as KS labelings that are described in (Shivanna et al., 2015). We demonstrate our results on Stochastic block models, real world MNIST datasets (binary and multiclass) and several real world UCI datasets. We also run experiments on large scale real world datasets Citeseer, Cora and Pubmed which are standard in GCN literature. In addition to this, we test the goodness of Lovász based embeddings in certain perfect graphs called caveman graphs which have been used to model simple social network communities. In addition to simple graphs, we also experiment with hypergraphs with clique expansion to see how the proposed method performs.

Stochastic Block Model: Synthetic Data Experiments: We start by describing our experiments on synthetically generated stochastic block model graphs. We perform the experiment on binary as well as three class classification problem. We report several settings of inter cluster and intra cluster probabilities in Figure 4 and corresponding embeddings in Figure 5. In each of the experiments, the input features are fixed to identity. We varied the number of nodes from 100 to 1000 and used a $20\% - 10\% - 70\%$ train-validation-test split where we use early stopping during training (Kipf and Welling, 2017). The test accuracy is compared against the standard GCNs (denoted by Kipf-GCN). We make several observations from the results in Figure 4. Firstly, as the inter and intra cluster probabilities get closer, it becomes much harder for GCN to classify well whereas LCN outperforms GCN by a significant margin. Secondly, as the size of the graph increases, the differences in connections become more critical and this is reflected in the increased accuracy with increase in nodes for LCN, whereas accuracy of GCN is almost agnostic to the number of nodes. Finally, as the graph becomes denser i.e., as connection probabilities tend towards 1, LCN performs much better than GCN in the three class setting. These results demonstrate the advantage of using LCNs over GCNs for semi-supervised classification tasks for SBM models.

Real World Data Experiments: We run several experiments on real world datasets including MNIST and UCI datasets. To make a fair comparison with state of the art, we first run the same set of binary classification experiments as in (Shivanna et al., 2015) and compare it with GCNs (Kipf and Welling, 2017), Graph Partition Neural Networks (GPNN) (Liao et al., 2018) and our proposed LCNs. These include experiments on 6 UCI datasets (breast-cancer $n = 683$, diabetes $n = 768$, fourclass $n = 862$, heart $n = 270$, ionosphere $n = 351$ and sonar $n = 208$) and experiments on certain pair of classes from a subsampled set of images from the MNIST datasets. For both the UCI and MNIST datasets features and labels are available, we used an RBF kernel on features to construct the graph. Table 1 reports the results for these experiments with various input embeddings including Laplacian (normalized, unnormalized), KS embedding and others as considered

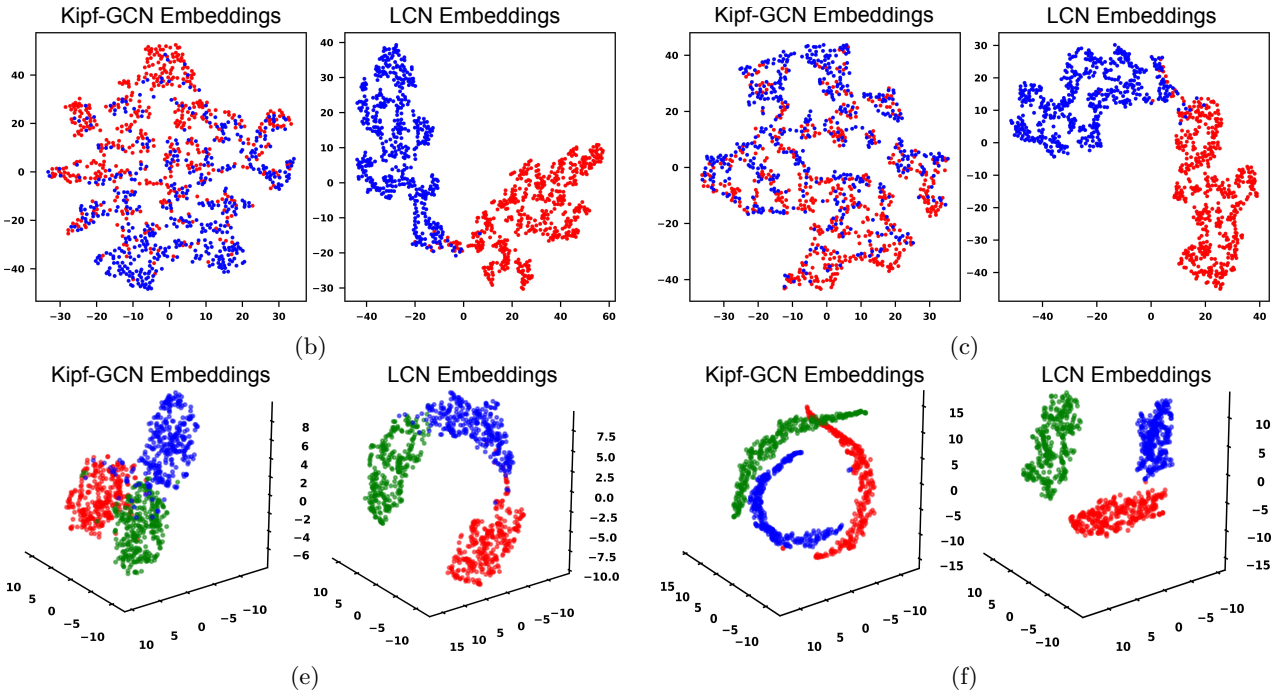


Figure 5: Embeddings learnt for settings corresponding to Figure 4 (b), (c), (e), (f) for n=1000

in (Shivanna et al., 2015). For MNIST datasets, the results are averaged over five randomly sampled graphs and for UCI datasets, the results are averaged over five random splits. As can be seen, LCN performs significantly better than SPORE and performs much better than GCNs in all datasets except two. In addition to the binary classification experiment, we also conducted three class classification on 500 and 2000 images from MNIST (randomly subsampled from classes 1, 2 and 7) and also 10 class classification where we randomly subsample 2000 images from all classes. The results are reported in Table 2 As the classes increase, LCN significantly outperforms other GCN baselines.

To be consistent with the GCN literature, we also run experiments on large scale datasets Citeseer, Cora and Pubmed. All these datasets are citation networks, where each document is represented as a node in the graph with an edge between nodes indicating the citation relation. The aim is to classify the documents into one of the predefined classes. We use the same splits as in (Yang et al., 2016). Table 3 shows the results on these large scale datasets, as explained in Section 5 we use the approximate K_{LS} kernel for these datasets and LCN (LS) refers to this setting. LCN outperforms other state-of-the-art baselines on all three datasets, Citeseer, Cora and Pubmed. Node2vec (Grover and Leskovec, 2016) is an unsupervised method for learning node representations for a given graph using just the structure of the graph. In table 3, *Node2vec* refers to the model when the kernel is obtained from normalized Node2vec embeddings, which achieves a significantly

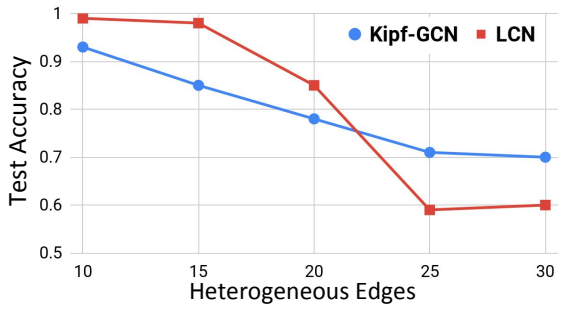


Figure 6: (c) Behavior of test accuracy with increase in the heterogeneous edges in the hypergraph.

poor performance.

Caveman graph: Goodness of Embeddings Experiment: A connected caveman graph of size (n, k) is formed by modifying a set of isolated k-cliques (or *caves*) by removing one edge from each clique and using it to connect to a neighboring clique along a central cycle such that all n cliques form a single unbroken loop (Watts, 1999). Caveman graphs are perfect graphs and are used for modeling simple communities in social networks (Kang and Faloutsos, 2011). We run our experiments on various synthetic caveman graphs. For every caveman graph, we compute the optimal coloring of the complement graph. We consider a binary classification setting and randomly assign nodes corresponding to half of the colors to class 0 and the other half to the class 1. We set initial features to be identity and we work with a $20\% - 20\% - 60\%$ train-validation-test splits, for every graph we run the experiments on

Dataset	Un-Lap	N-Lap	KS	SPORE	Kipf-GCN	GPNN	LCN
breast-cancer	88.2	93.3	92.8	96.7	**97.6**	95.5	97.2
diabetes	68.9	69.3	69.4	73.3	71.4	68.0	**76.3**
fourclass	70.0	70.0	70.4	78.0	80.5	73.9	**81.7**
heart	72.0	75.6	76.4	82.0	**85.1**	81.1	82.5
ionosphere	67.8	68.0	68.1	76.1	76.1	70.0	**87.9**
sonar	58.8	59.0	59.3	63.9	71.4	64.8	**73.2**
mnist-500 1 vs 2	75.6	80.6	79.7	85.8	98.0	96.2	**99.0**
mnist-500 3 vs 8	76.9	81.9	83.3	86.1	92.3	83.1	**93.7**
mnist-500 4 vs 9	68.4	72.0	72.2	74.9	**89.4**	88.5	83.3
mnist-2000 1 vs 2	83.8	96.2	95.0	96.7	99.0	97.5	**99.2**
mnist-2000 3 vs 8	55.2	87.4	87.4	91.4	94.7	89.6	**95.7**
mnist-2000 1 vs 7	90.7	96.8	96.6	97.3	**98.8**	96.4	98.7

Table 1: Binary Classification with Random label-to-color assignment in UCI and MNIST datasets.

Dataset	Kipf-GCN	GPNN	LCN
mnist 500 127	96.1	93.8	**97.5**
mnist 2000 127	97.2	94.4	**97.4**
mnist all 2000	84.4	56.9	**85.1**

Table 2: Multi class Classification in MNIST dataset

Dataset	Node2vec	Kipf-GCN	GPNN	LCN
Citeseer	23.1	70.3	69.7	**73.5**
Cora	31.9	81.5	81.8	**82.6**
Pubmed	42.3	79	79.3	**79.7**

Table 3: Performance for semi-supervised on Citeseer, Cora, Pubmed datasets

10 random splits. Table 4 shows the test accuracy of Kipf-GCN and LCN. In Table 4, *Avg_same* stands for average inner product of the representations of the nodes with same color and *Avg_diff* stands for that of the nodes with different colors. As we see, LCN performs better on all cases considered. Also, the average dot products of nodes with same color is high and those with different colors is close to zero showing that the representations are as well separated as possible for nodes with different colors.

Hypergraphs: Homogeneous vs Heterogeneous Edges Experiment: Though our main focus is on simple graphs, we also experiment with synthetic hypergraphs. A hypergraph is a generalized version of a graph where a hyper edge consists of a set of nodes. However, for hypergraphs, to the best of our knowledge, orthogonal embeddings and Lovász theta function are not defined. Therefore, we consider the clique expansion of the hypergraphs (Zhou et al., 2006). Clique expansion creates a simple graph from a hypergraph by replacing every hyperedge with a clique. In our experiments, we generated a hypergraph of 100 nodes with every hyperedge containing 35 nodes. We consider a binary classification setting and assign randomly 50 nodes to one class and the other 50 to a different class. We randomly create 20 hyperedges such that all the nodes in the hyperedge belong to same class, we call these edges homogeneous edges. We also create m random hyperedges such that the label distribution of the nodes in the hyperedge is 2:3, we call these edges heterogeneous. We vary m between 10 and 30 and create multiple hypergraphs. We set the initial features to identity and work with a 20%-20%-60% train-validation-test split and average across ten runs per each hypergraph. Figure 6 shows the behavior of test accuracy with increase in the number of heterogeneous edges. As one can see LCN performs much better than GCN when the number of heterogeneous edges are small (and hence the clique expansion has a community-like structure) whereas GCNs tend to perform better with increase in the number of heterogeneous edges.

(n, k)	Kipf-GCN	LCN	Avg_same	Avg_diff
(50, 10)	0.92	**0.93**	0.83	-0.008
(75, 6)	0.77	**0.80**	0.80	-0.005
(100, 5)	0.71	**0.73**	0.79	-0.003
(100, 7)	**0.81**	**0.81**	0.80	-0.003

Table 4: Caveman graph experiment: Average test accuracy of Kipf-GCN and LCN on caveman graphs.

7 Conclusion

We propose Lovász Convolution Networks for the problem of semi supervised learning on graphs. Our analysis shows settings where LCNs perform much better than GCNs. Our results on real world and synthetic datasets demonstrate the superior embeddings learnt by LCNs and show that they significantly outperform GCNs. Future work includes detailed analysis of Lovász embeddings for hypergraphs, use of LCNs for community detection and clustering.

References

Agarwal, S. (2006). Ranking on graph data. In *International Conference on Machine Learning*.

Ando, R. K. and Zhang, T. (2007). Learning on graph with laplacian regularization. In *Advances in neural information processing systems*.

Atwood, J. and Towsley, D. (2016). Diffusion-convolutional neural networks. In *Advances in neural information processing systems*.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference*.

Belkin, M., Niyogi, P., and Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*.

Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. (2014). Spectral networks and locally connected networks on graphs. In *ICLR*.

Chen, J., Ma, T., and Xiao, C. (2018). Fastgcn: Fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247*.

Chudnovsky, M., Robertson, N., Seymour, P., and Thomas, R. (2006). The strong perfect graph theorem. *Annals of mathematics*.

Condon, A. and Karp, R. M. (1999). Algorithms for graph partitioning on the planted partition model. In *Randomization, Approximation, and Combinatorial Optimization. Algorithms and Techniques*.

Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*.

Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. (2015). Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*.

Getoor, L. and Taskar, B. (2007). *Introduction to statistical relational learning*.

Giles, C. L., Bollacker, K. D., and Lawrence, S. (1998). Citeseer: An automatic citation indexing system. In *Proceedings of the Third ACM Conference on Digital Libraries*.

Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM.

Henaff, M., Bruna, J., and LeCun, Y. (2015). Deep convolutional networks on graph-structured data. *CoRR*.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social networks*.

Jain, A., Zamir, A. R., Savarese, S., and Saxena, A. (2016). Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Jethava, V., Martinsson, A., Bhattacharyya, C., and Dubhashi, D. (2013). Lovasz theta function, svms and finding dense subgraphs. *Journal of Machine Learning Research*.

Johansson, F., Jethava, V., Dubhashi, D., and Bhattacharyya, C. (2014). Global graph kernels using geometric embeddings. In *Proceedings of the 31st International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*.

Kang, U. and Faloutsos, C. (2011). Beyond'caveman communities': Hubs and spokes for graph compression and mining. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *ICLR*.

Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010a). Predicting positive and negative links in online social networks. In *Proceedings of the 19th International Conference on World Wide Web*.

Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010b). Signed networks in social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Li, Q., Han, Z., and Wu, X.-M. (2018). Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*.

Liao, R., Brockschmidt, M., Tarlow, D., Gaunt, A. L., Urtasun, R., and Zemel, R. (2018). Graph partition neural networks for semi-supervised classification. *arXiv preprint arXiv:1803.06272*.

Lovász, L. (1979). On the shannon capacity of a graph. *IEEE Transactions on Information theory*.

Lovász, L. (2009). A characterization of perfect graphs. In *Classic Papers in Combinatorics*, pages 447–450. Springer.

Lovász, L. and Vesztergombi, K. (1999). Geometric representations of graphs. In *IN PAUL ERDÖS, PROC. CONF.*

Lu, Q. and Getoor, L. (2003). Link-based classification. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*.

Luz, C. J. (1995). An upper bound on the independence number of a graph computable in polynomial-time. *Operations Research Letters*, 18(3):139–145.

Luz, C. J. and Schrijver, A. (2005). A convex quadratic characterization of the lovász theta number. *SIAM Journal on Discrete Mathematics*, 19(2):382–387.

Marcheggiani, D. and Titov, I. (2017). Encoding sentences with graph convolutional networks for semantic role labeling. In *EMNLP*.

McAuley, J. and Leskovec, J. (2012). Learning to discover social circles in ego networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*.

Ramírez-Alfonsín, J. L. and Reed, B. A. (2001). *Perfect graphs*, volume 44. Wiley.

Ray, S. N., Dasgupta, S. S., and Talukdar, P. (2018). Ad3: Attentive deep document dater. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1871–1880. Association for Computational Linguistics.

Sen, P., Namata, G. M., Bilgic, M., Getoor, L., Gallagher, B., and Eliassi-Rad, T. (2008). Collective classification in network data. *AI Magazine*.

Shivanna, R., Chatterjee, B. K., Sankaran, R., Bhattacharyya, C., and Bach, F. (2015). Spectral norm regularization of orthonormal representations for graph transduction. In *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc.

Subramanya, A. and Talukdar, P. P. (2014). Graph-based semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(4):1–125.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*.

van der Maaten, L. and Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

Vashishth, S., Dasgupta, S. S., Ray, S. N., and Talukdar, P. (2018a). Dating documents using graph convolution networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1605–1615. Association for Computational Linguistics.

Vashishth, S., Joshi, R., Prayaga, S. S., Bhattacharyya, C., and Talukdar, P. (2018b). RESIDE: Improving distantly-supervised neural relation extraction using side information. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1257–1266. Association for Computational Linguistics.

Vashishth, S., Yadav, P., Bhandari, M., Rai, P., Bhattacharyya, C., and Talukdar, P. (2018c). Graph convolutional networks based word embeddings. *CoRR*, abs/1809.04283.

Vashishth, S., Yadav, P., Bhandari, M., and Talukdar, P. (2019). Confidence-based graph convolutional networks for semi-supervised learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Watts, D. J. (1999). Networks, dynamics, and the small-world phenomenon. *American Journal of sociology*.

Yang, Z., Cohen, W. W., and Salakhutdinov, R. (2016). Revisiting semi-supervised learning with graph embeddings. In *International Conference on Machine Learning*.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. (2004). Learning with local and global consistency. In *Advances in neural information processing systems*.

Zhou, D., Huang, J., and Schölkopf, B. (2006). Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in neural information processing systems*.

Zhu, X., Ghahramani, Z., and Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *International Conference on Machine Learning*.

Zhuang, C. and Ma, Q. (2018). Dual graph convolutional networks for graph-based semi-supervised classification. In *TheWebConf*.

Zitnik, M. and Leskovec, J. (2017). Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*.