IMPROVING INDUCTIVE LINK PREDICTION THROUGH LEARNING GENERALIZABLE NODE REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Link prediction is a core task in graph machine learning, as it is useful in many application domains from social networks to biological networks. Link prediction can be performed under different experimental settings: (1) transductive, (2) semi-inductive, and (3) inductive. The most common setting is the transductive one, where the task is to predict whether two observed nodes have a link. In the semi-inductive setting, the task is to predict whether an observed node has a link to a newly observed node, which was unseen during training. For example, cold start in recommendation systems requires suggesting a known product to a new user. We study the inductive setting, where the task is to predict whether two newly observed nodes have a link. The inductive setting occurs in many real-world applications such as predicting interactions between two poorly investigated chemical structures or identifying collaboration possibilities between two new authors. In this paper, we demonstrate that current state-of-the-are techniques perform poorly under the inductive setting, i.e., when generalizing to new nodes, due to the overlapping information between the graph topology and the node attributes. To address this issue and improve the robustness of link prediction models in an inductive setting, we propose new methods for designing inductive tests on any graph dataset, accompanied by unsupervised pre-training of the node attributes. Our experiments show that the inductive test performances of the state-of-the-art link prediction models are substantially lower compared to the transductive scenario. These performances are comparable, and often lower than that of a simple multilayer perceptron on the node attributes. Unsupervised pre-training of the node attributes improves the inductive performance, hence the generalizability of the link prediction models.

1 INTRODUCTION

Graph datasets are ubiquitous. Examples include friendship networks (Ball and Newman, 2013), collaboration networks (Wang et al., 2020), protein interaction networks (Qi et al., 2006), power grids (Pagani and Aiello, 2011), and transportation networks (Lordan and Sallan, 2020). Real-world graphs are often sparse and partially observed, which makes predicting the unobserved links a problem of great interest (Liben-Nowell and Kleinberg, 2007). Applications of link prediction include predicting unknown protein interactions, exploring drug responses (Stanfield et al., 2017), recommending products to users (Lakshmi and Bhavani, 2021), completing knowledge graphs (Nickel et al., 2016), and suggesting friends in social networks (Adamic and Adar, 2003).

Link prediction is a well-researched problem in graph machine learning, and numerous methods have been developed which include similarity-based indices, probabilistic methods, dimensionality reduction approaches etc. (Kumar et al., 2020). Latent representations of nodes and edges are often used for link prediction (Cao et al., 2015; Grover and Leskovec, 2016; Perozzi et al., 2014; Tang et al., 2015; Wang et al., 2016). These embeddings encode the graph topology in low-dimensional feature vectors, which are then used for training on the observed links. Simple features representing the graph topology can also achieve similar performance as these deep learning-based latent features (Ghasemian et al., 2020). In a recent work, disentangling the topological structure of the graph and the node attributes improved performance in link prediction (Ai et al., 2022).

With the increased interest in graph machine learning, the research community has developed multiple benchmark datasets for evaluating the performance of the machine learning models. The Deep Graph Library (DGL) includes benchmark graph datasets from multiple domains including citation networks, collaboration networks, biological networks, co-buy networks, etc (Wang et al., 2019). DGL's custom data loader programs support efficient loading and manipulation of large graph datasets, and help in optimizing the computational patterns of graph neural network (GNN) (Scarselli et al., 2009) models. The Open Graph Benchmark (OGB) has a diverse set of benchmarks containing biological networks, molecular graphs, source code ASTs, and knowledge graphs (Hu et al., 2020). OGB also standardizes the train-test splits for out-of-distribution generalization under realistic data splits and well-defined performance metrics.

However, the majority of the link prediction models and the benchmarks have focused on transductive link prediction, where the training and the test graphs share the same set of nodes. Many real world applications require making inductive link predictions on newly observed nodes, unseen during training. For example, cold-start is an extensively explored problem in recommendation systems, which requires suggesting a known product to a new user (Maksimov et al., 2020). This resembles a semi-inductive setting, where one node of the test link is seen during training, while the other one is a newly arrived node. When both the nodes are unseen, the prediction task is further complicated and resembles an inductive test scenario. For example, AI-Bind (Chatterjee et al., 2021; Menichetti, 2022) solves inductive link prediction in protein-ligand interaction networks, where both the protein and the ligand in the test set are unseen during training. Inductive link prediction in knowledge graphs has also got much attention recently (Bonner et al., 2022).

Present link prediction benchmarks like OGB do not focus on inductive link prediction. The majority of the complex neural networks developed on the benchmark datasets perform well in transductive tests, but fail in an inductive setting. These state-of-the-art models learn mainly from the topology of the training graph, which can also be achieved via much simpler models that only use the topological features to make link prediction. In this work, we first define inductive tests on the benchmark graph datasets. Thereafter, we propose a method for selecting the node attributes which are the most informative for inductive tests, and helps the link prediction models to learn beyond the graph topology.

OUR CONTRIBUTION

1. A method for designing inductive tests on any graph dataset. We introduce OGB-Inductive, an extension of the Open Graph Benchmark (Hu et al., 2020) for inductive learning.
2. A method to quantify the information contained in the graph topology and the node attributes.
3. We show the importance of learning beyond graph topology, and using the node attributes in inductive link prediction.
4. We propose unsupervised pre-training of the node attributes, which improves the transferablity of the link prediction models.

2 BACKGROUND

Let $G = (V, E, X)$ be a graph, where V is the set of vertices (or nodes), E is the set of edges (or links), and the X matrix contains the node attributes. Here we consider an undirected unipartite graph. This formulation can be extended to directed graphs, bipartite graphs, and multilayered graphs. For example, G may be a protein-protein interaction network, where the nodes are the proteins, the links are the interactions between the proteins, and the node metadata are the embeddings of the molecular structures of the proteins (Asgari and Mofrad, 2015).

A link prediction model may be constructed using supervised learning. We partition the edge set into observed and unobserved edges $E = E_o \cup E_u$. We learn a function that maps the observed nodes and the node attributes to the observed edges $\{V_o, X_o\} \rightarrow E_o$, and hope that it will generalize to the unobserved edges E_u. Furthermore, we define three types of link prediction scenarios based on the observed and the unobserved nodes V_o and V_u, respectively:

- Transductive: Predicting $(a, b) \in E_u$, where $a, b \in V_o$,

- Semi-inductive: Predicting $(a, b) \in E_u$, where $a \in V_o$ and $b \in V_u$ or vice-versa,
- Inductive: Predicting $(a, b) \in E_u$, where $a, b \in V_u$.

Transductive tests are the easiest among the three, and simple algorithms such as configuration models (Barabási, 2016; Chatterjee et al., 2021) or models using stacked topological features can often achieve performances comparable to complex deep learning models (Ghasemian et al., 2020). In semi-inductive tests, where one node is unseen during training, the complexity of the prediction task increases. In this scenario, the model is required to learn from both the training graph topology and the node attributes to make accurate predictions. Inductive tests represent the most complex link prediction scenario, where both the nodes associated with the test links are unseen during training.

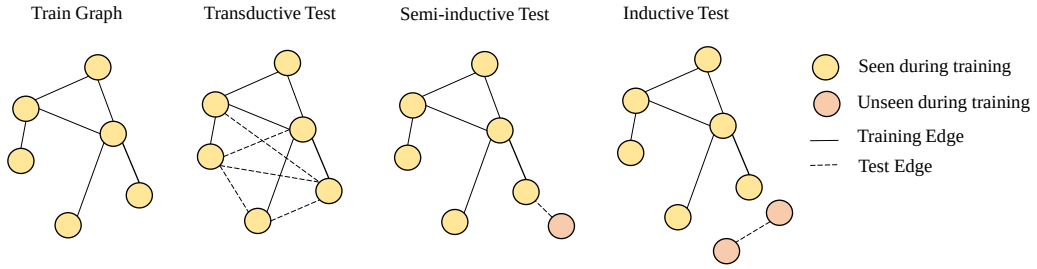

Figure 1: Visual representation of transductive, semi-inductive, and inductive tests in link prediction.

3 Related work

Machine learning models use both the graph topology and the node attributes in link prediction (Ai et al., 2022). When the training and the test graphs share similar topology, the topological information is of more importance compared to the node attributes. Existing inductive link prediction methods like GraphSAGE (Hamilton et al., 2017) perform well in inductive tests when the links associated with the newly arrived nodes are known, and the test graph shares similar topological features as the graph used in training. As a random split creates the training and the test graphs with similar topologies, using simple topological features on the nodes and the links (like degree, centrality etc.) are sufficient to predict the unobserved links (Ghasemian et al., 2020). AI-Bind shows how the degree information in DTIs are used for predicting new protein-ligand interactions, circumventing the molecular structures (Chatterjee et al., 2021).

(Joachims, 1999) first mentioned about the difficulty of inductive tests compared to transductive ones in a text classification setting. This understanding is valid for other domains of machine learning, and can be extended to graph machine learning, particularly link prediction. Planetoid (Yang et al., 2016), GraIL (Teru et al., 2019), and GraphSAGE (Hamilton et al., 2017) are among the most recognized methods for inductive link prediction which learn from the graph topology. GraphSAGE learns the neighborhood information for each node in training to make predictions from the neighborhoods of the newly arrived nodes in inductive tests. DEAL uses both the topological information and the node attributes in making link prediction in both transductive and inductive settings (Hao et al., 2020). Structure Enhanced Graph neural network (SEG) uses a simple one-layer GCN to encode the topology of the training graph, which combined with a simple multilayer perceptron (MLP) on the node attributes significantly improves the transductive test performance (Ai et al., 2022). A recently proposed open challenge for inductive link prediction on Knowledge Graphs (Galkin et al., 2022) has inspired many state-of-the-art inductive link prediction models.

4 Limitations of the current methods for Link Prediction

4.1 Benchmark Graph Datasets are insufficient for Inductive Tests

The Open Graph Benchmark (OGB) dataset provides a useful benchmark for comparing link prediction methods (Hu et al., 2020), but is limited to the transductive setting, and the provided train-test

splits are inadequate for inductive testing. OGB provides large scale graph datasets from various domains like social networks, biological networks, and molecular graphs. The train-validation-test splits are specifically tailored to test generalization based on specific properties associated with each graph. For example, in ogbl-ppa (protein-protein interaction network), the training graph consists of the interactions obtained via high throughput technology, or even text-mining. This method of obtaining the interactions is cost effective, but produces low confidence data. The validation and the test datasets in this setting are obtained from low throughput and resource intensive experiments. Thus, the validation and tests datasets are of high confidence and provide a challenging generalization scenario.

However, the OGB benchmark tasks are limited to transductive tests only. Large overlap between the nodes in the train, the validation, and the test graphs limits us from creating node-disjoint train and test datasets for an inductive scenario. Table 6 in Appendix A shows the node overlaps between the train-validation-test splits in the OGB datasets. We lose the majority of the training edges when we discard the edges from the training graph which share nodes with the test dataset. The OGB data splits are thus insufficient for inductive testing.

4.2 Topological Shortcuts in Transductive Tests

We perform a simple experiment showing that the performance of the current state-of-the-art deep models on the link prediction benchmarks can be obtained by simple configurations models which ignore the node attributes and rely only on the topology of the training graph. This is strong evidence that these deep models are reliant on shortcuts exploiting the graph topology and ignoring node attributes. This is problematic since the graph topology cannot be used for generalization to unseen nodes in inductive tests.

Shortcuts in transductive learning have been investigated in multiple previous studies. (Ghasemian et al., 2020) showed that link prediction models stacking topological features have sub-optimal link prediction performance in graph data from different domains such as social networks, biology networks, information networks, and transportation networks. Configuration models use only the degree sequence of the training graph, and through an entropy maximization algorithm produce the probabilities associated with the unobserved links. In Chatterjee et al. (2021), a duplex configuration model, which uses only the degree sequences of the proteins and the ligands in the training DTI, achieves excellent test performance in predicting protein-ligand interactions, comparable to state-of-the-art deep neural networks.

Here we consider two types of configuration models. The first one is a traditional soft configuration model (van der Hoorn et al., 2017), which takes as input the number of nodes n and the degree sequence of the training graph $\mathbf{k}$. This model depends only on the degree sequence of the observed positive edges and does not require any information on the negative edges. The traditional configuration model is represented by $\mathrm{SCM}(n, \mathbf{k})$, where $\mathbf{k} = \{k_i\}_{i=1}^n$, $k_i \geq 0$.

The SCM is an exponential random graph model (Fronczak, 2012; Menichetti and Remondini, 2014; Menichetti et al., 2015), where $P(G) = B(G)/Z$, the Boltzmann factor is $B(G) = \exp(-\sum_{i=1}^n \lambda_i d_i(G))$, and $d_i(G) = \sum_{j=1}^n G_{ij}$ is the degree of the node i, and $\{G_{ij}\}_{i,j=1}^n$ is the adjacency matrix of the training graph G. By entropy maximization, the link probability between the nodes i and j is

$$p_{ij} = \frac{1}{e^{\lambda_i + \lambda_j} + 1}, \tag{1}$$

where the Lagrange multipliers $\{\lambda_i\}$ are such that $\langle d_i \rangle = \sum_{j=1}^n p_{ij} = k_i$ for all nodes $i \in \{1, 2, \ldots, n\}$.

We also use a duplex configuration model from Menichetti et al. (2014) which takes into account both the positive and negative edges for making the link predictions. We use the same negative sampling strategy as OGB. We randomly sample the negative edges from the unobserved links in the training graph, and keep the number of the positive and the negative edges the same in training. The duplex configuration model represents the positive and the negative training graphs as a two-layered network. Thereafter, it runs an entropy maximization algorithm jointly on both the layers.

Using the link probabilities in Eq. 1 for the traditional configuration model, and the formulation developed in Menichetti et al. (2014) for the unipartite duplex configuration model, we compute the

transductive test performances in terms of the metrics Area Under the Receiver Operating Characteristic (AUROC), Area Under the Precision-Recall Curve (AUPRC), and Hit@Top K on the benchmark ogbl-ddi dataset (drug-drug interaction network). In Table 1, we compare the performances of the traditional and the duplex configuration models with Adaptive Graph Diffusion Networks (AGDN, (Sun et al., 2020)), Path-aware Siamese Graph Neural Network (PSG, (Lv et al., 2022)), and Pairwise Learning for Neural Link Prediction (PLNLP, (Wang et al., 2021)) in the transductive setting. AGDN, PSG, and PLNLP are state-of-the-art link prediction models, achieving top performances on the ogbl-ddi benchmark. We observe that the configuration models outperform the state-of-the-art neural network models on the OGB benchmark performance metric Hits@Top K but not on AUROC and AUPRC. In the transductive link prediction setting, the fact that simple configuration models learned from purely topological data can outperform recent deep learning based models trained on both topological data and node attributes provides evidence that these deep learning models derive much of their predictive power from topological features alone. Similar to our observation, a recent work (Stolman et al., 2022) shows that classic graph structural features outperform graph embedding-based methods in another downstream task on graphs (namely, community labeling) when the performance measure is Hits@Top K.

Table 1: Link prediction results in transductive tests. The traditional and duplex configuration models outperform the state-of-the-art neural network models for Hits@Top K on the benchmark ogbl-ddi dataset. For calculating the Hits@Top K, we use K=20 as recommended by the OGB benchmark.

Model	**AUROC**	**AUPRC**	**Hits@Top K(%)**
Traditional Configuration Model	0.87 ± 0.01	0.90 ± 0.00	$\mathbf{0.99 \pm 0.00}$
Duplex Configuration Model	0.87 ± 0.01	0.90 ± 0.00	$\mathbf{0.99 \pm 0.00}$
AGDN	$\mathbf{0.97 \pm 0.03}$	$\mathbf{0.98 \pm 0.02}$	0.95 ± 0.01
PSG	$\mathbf{0.97 \pm 0.04}$	0.96 ± 0.05	0.93 ± 0.01
PLNLP	0.92 ± 0.04	0.95 ± 0.02	0.91 ± 0.03

5 Experiments on Inductive Link Prediction

We first address the limitations of the existing benchmarks by designing methods for inductive tests on the OGB link prediction benchmarks. We use a random node split to define inductive tests on the OGB benchmark datasets. Next, we show that the state-of-the-art link prediction models perform poorly in inductive tests, and achieve lower performances than a simple MLP model trained on node attributes. Non-overlapping topological information between the training and the test graphs in the inductive setting compels the models to use only the node attributes for making meaningful link predictions on the nodes unseen during training. Next, we quantify the information contained in different node features and show that the pre-trained node attributes accommodate the most information beyond the graph topology. Finally, we run inductive tests using different node features and validate the hypothesis that pre-trained node attributes are the most effective node features in inductive tests.

5.1 Designing Inductive tests

Traditionally the train-validation-test split in link prediction tasks is obtained via random edge split (Hamilton et al., 2017; Galkin et al., 2022). Creating an inductive test scenario by combining the random edge split with the removal of the overlapping nodes between the training and the test graphs eliminates the majority of the links from the training dataset (see Appendix B1). Inspired by GraIL (Teru et al., 2019), we use a random node split, which can create inductive test scenarios for any graph dataset. The algorithm for random node split is as follows (visualized in Figure 2):

1. Randomly split the nodes of the original graph V into three groups V_{train}, $V_{validation}$, and V_{test}. We divide the nodes at a 80:10:10 ratio for the train, the validation, and the test datasets.
2. Obtain the subgraphs G_{train}, $G_{validation}$, and G_{test} induced by the node sets V_{train}, $V_{validation}$, and V_{test}, respectively.

We observe that the number of edges lost in the random node split is significantly less compared to the random edge split method. We use random node split to design inductive tests on the OGB link prediction datasets, creating *OGB-Inductive*, an extension of the OGB benchmark for inductive link prediction. The number of nodes and edges for each of the undirected link prediction dataset is summarized in Table 2 and Table 3, respectively.

Table 2: Number of nodes in different graph datasets in OGB-Inductive.

Dataset	Train Nodes	Validation Nodes	Test Nodes
ogbl-ppa	461,031	57,629	57,629
ogbl-collab	188,694	23,587	23,587
ogbl-ddi	3,413	427	427

Table 3: Edges in different graph datasets in OGB-Inductive, and the edges lost in random node split.

Dataset	Train Edges	Validation Edges	Test Edges	Edges lost
ogbl-ppa	19,460,915	296,195	303,563	10,265,600
ogbl-collab	821,974	12,739	12,826	437,926
ogbl-ddi	864,478	13,869	11,433	445,109

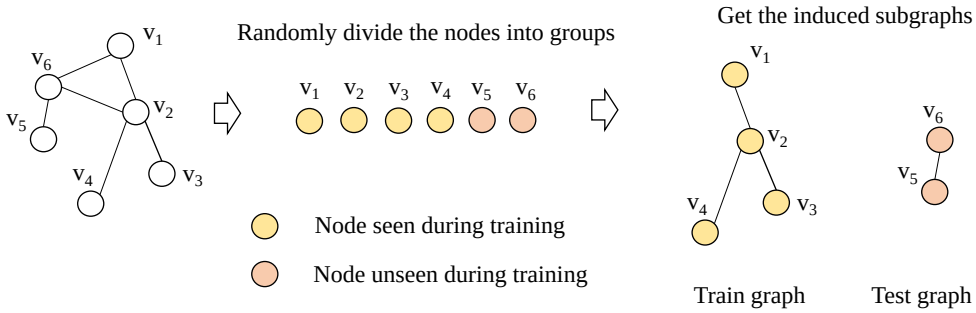

Figure 2: Creating train and test graphs for inductive tests using the random node split method.

5.2 State-of-the-art link prediction models fail in inductive tests

Transductive link prediction is largely driven by topological features. To show the importance of node attributes in inductive link prediction, we design a simple MLP which takes the concatenated attributes of two nodes as input and compare its performance with PLNLP (Wang et al., 2021) on OGB-Inductive. The MLP contains three dense layers. For ogbl-ppa, we use the OGB defined 58-dimensional one-hot feature vectors as the node attributes. These indicate the species associated with the proteins. For ogbl-collab, we use the 128-dimensional features obtained by averaging the word embeddings of papers published by the authors. For ogbl-ddi, we use 300-dimensional Mol2vec embeddings of the drug structures. The performances are summarized in Table 4.

5.3 Quantifying the information in the node attributes

In this section, we develop a method to measure the information contained in different types of node attributes in order to identify the attributes most suitable for inductive link prediction.

Table 4: Inductive test performance of PLNLP is significantly lower than its transductive test performance, and is lower than an MLP on the node attributes. For Hits@Top K, we use the default K=100, 50, and 20 for ogbl-ppa, ogbl-collab, and ogbl-ddi, respectively.

Dataset	PLNLP in Transductive Test Hits@Top K(%)	PLNLP in Inductive Test Hits@Top K(%)	MLP on node attributes Hits@Top K(%)
ogbl-ppa	32.38 ± 2.58	0.09 ± 0.03	0.39 ± 0.03
ogbl-collab	70.59 ± 0.29	11.56 ± 0.93	36.44 ± 3.11
ogbl-ddi	90.88 ± 3.13	0.01 ± 0.02	0.39 ± 0.02

5.3.1 METHODOLOGY

Many real world graphs are naturally organised into community structures. In social networks, the communities are often based on different interest groups. Protein-protein interaction networks organize the protein structures into communities based on their functions in metabolism (Radicchi et al., 2004). Community detection is a well-explored problem in network science (Newman, 2006). Various community detection algorithms cluster nodes into meaningful communities based on graph topology and node attributes. Locally dense subgraphs often form communities, which can be detected by the traditional community detection algorithms. Unsupervised clustering on the node attributes helps in detecting communities beyond the topology of the graphs (Yang et al., 2013). In this subsection, we use the Davies-Bouldin score (Davies and Bouldin, 1979), a cluster separation measure, to quantify the quality of node clusters obtained from unsupervised clustering using only node attributes. Furthermore, we show that when the node attributes are pre-trained in an unsupervised fashion on a dataset different from the training graph, these node attributes produce the best unsupervised clusters and improve inductive test performance.

We start by observing the quality of the unsupervised node clusters under these scenarios:

- Using Node2vec (Grover and Leskovec, 2016) as the nodes feature, which encodes the topology of the graph.
- Pre-train the node attributes independently of the training graph used in link prediction.
- Randomly shuffle the pre-trained node attributes for each node individually.
- Replace the pre-trained node attributes with random entries drawn from a uniform distribution, removing any information contained in them.

We run the K-means (Macqueen, 1967) algorithm on the node features to obtain unsupervised node clusters. We then compute the Davies-Bouldin score for each of the node features listed above. A lower Davies-Bouldin score implies that the average similarity between clusters is lower and the quality of the clustering is better. Furthermore, we measure adjusted mutual information (AMI) (Vinh et al., 2009) between the clusters obtained using the pre-trained node attributes and the ones obtained using Node2Vec. Low AMI values indicate that the node attributes contains information disjoint to the graph topology, and are thus suitable for inductive tests. The proposed method is visualized in Figure 3.

5.3.2 OBSERVATIONS

We now use the above mentioned methodology to explore different node features in the OGB link prediction benchmark datasets. We train the node attributes of the graphs in OGB-Inductive in an unsupervised manner. For ogbl-ppa, we use the pre-trained 100-dimensional ProtVec (Asgari and Mofrad, 2015) vectors as the pre-trained node attributes. The corpus used in the training of the ProtVec model includes 546,790 amino acid sequences obtained from the Swiss-Prot database (Bairoch, 1996). For ogbl-collab, we use 128-dimensional Word2vec embeddings on the papers for each author. The Word2Vec training corpus uses Google news articles with approximately 6B tokens (Mikolov et al., 2013). For ogbl-ddi, we use the pre-trained 300-dimensional Mol2vec (Jaeger et al., 2018) embeddings as the node attributes. The Mol2vec training corpus includes 19.9M chemicals obtained from the ZINC (Irwin et al., 2012) and ChEMBL (Gaulton et al., 2011) libraries.

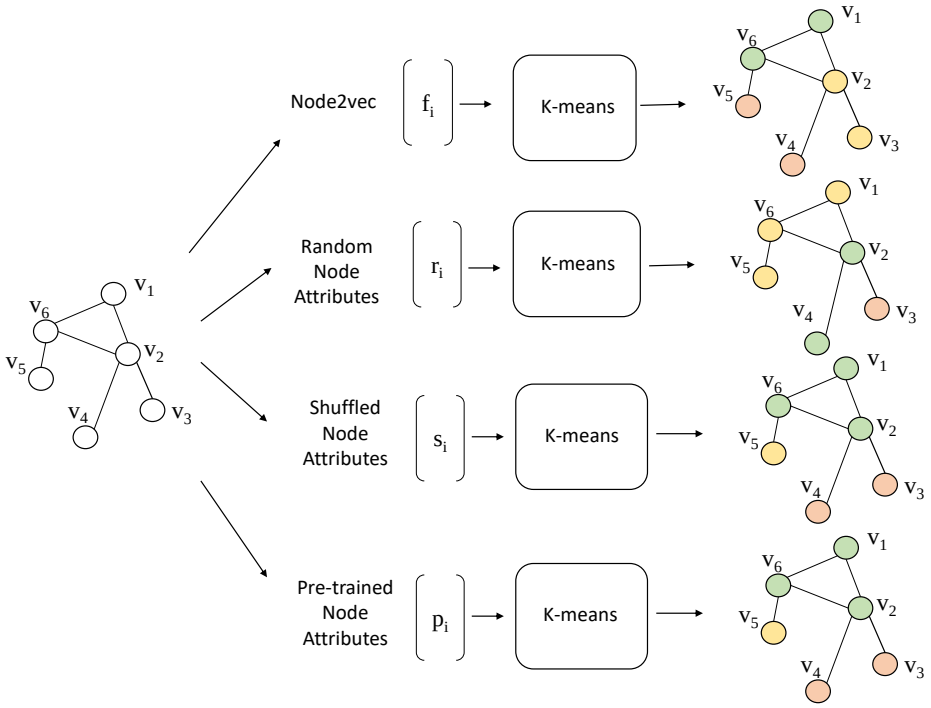

Figure 3: Unsupervised clustering of nodes using different node attributes. We use the resultant clusters to quantify how disjoint the information in the node attributes is from the information in the graph topology.

We consider the effect of shuffling the components of each node attribute vector for each node individually to disrupt the information contained in them. This process keeps the distribution of the attributes unchanged for each node. In another experiment, we replace the node attributes with uniform random entries. Comparing the Davies-Bouldin score for all these scenarios shows us that the pre-trained node attributes produce the most meaningful node clustering (see Figure 4). Hence, the pre-trained node attributes are the most informative node features beyond the graph topology, and the most suited for inductive tests. Moreover, in Table 5 we observe a significant reduction in the adjusted mutual information (AMI) for the unsupervised clusters obtained using the pre-trained node attributes with respect to the clusters obtained using only Node2vec. This confirms that the pre-trained node attributes accommodate minimal topological information from the training graph.

Table 5: We compare the adjusted mutual information between the unsupervised clusters obtained using only Node2vec and the pre-trained node attributes. Node2vec uses the graph topology to create the node clusters. Low AMI values indicate that the pre-trained attributes share minimal information with the graph topology.

Dataset	Node2vec	Pre-trained
ogbl-ppa	1.0	0.17
ogbl-collab	1.0	0.08
ogbl-ddi	1.0	0.22

5.4 Improved generalizability

We use the node features described above for inductive link prediction on OGB-Inductive. We use the same MLP architecture described in subsection 5.2. In Figure 5, we observe that the pre-trained node attributes, for which the Davies-Bouldin scores are the lowest, (i.e., the node features which create the best unsupervised node clustering) perform the best in inductive tests. Hence, we

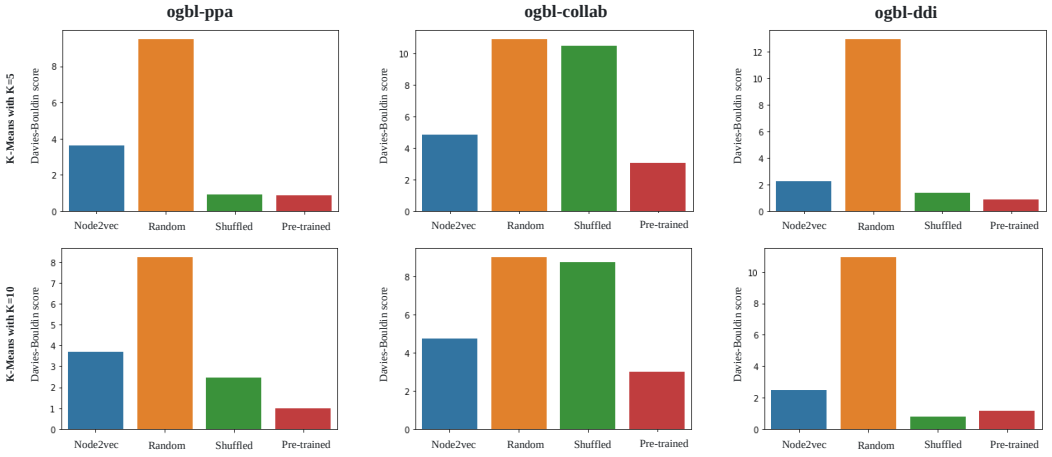

Figure 4: Comparing the quality of the unsupervised clusters using Davies-Bouldin score for various node features in the OGB link prediction benchmarks. The pre-trained node attributes show the lowest Davies-Bouldin score in all of the graph datasets, and hence are the most suitable for inductive link prediction tasks. Here we run the K-means algorithm with K=5 and K=10.

validate our proposed method that selecting the most informative node attributes leads to the most improvement in generalizability of link prediction models.

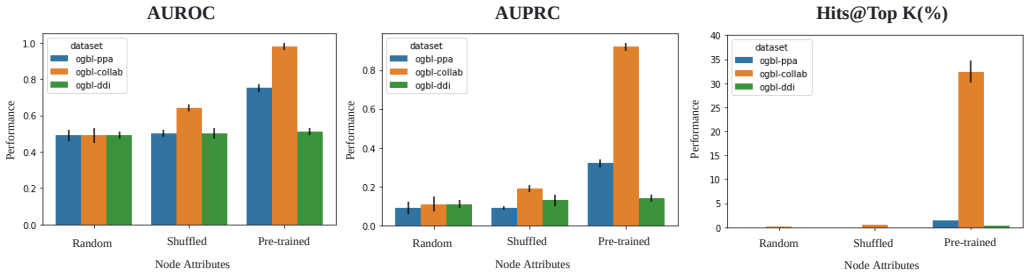

Figure 5: We compare AUROC, AUPRC and Hits@Top K performances for different node attributes in inductive tests. The pre-trained node attributes show the highest performance across the OGB link prediction datasets.

6 Conclusion and Future Work

In this work, we have shown how the link prediction models leverage the topology of the training graph to achieve excellent transductive test performance. We have developed a pipeline to design inductive tests on any graph dataset. We observe that the performance of the state-of-the-art link prediction models reduce significantly in inductive tests compared to the transductive scenario. These performances are comparable, and sometimes lower than that of a simple MLP on the node attributes. Furthermore, we have developed a method to quantify the goodness of node attributes, and experimentally shown that the pre-trained node attributes are the most suitable for improving the generalizability in link prediction tasks. As a future work, we plan to explore multiple questions regarding the choice of node attributes for better performances in inductive tests. Our pipeline can be used for selecting the best-suited corpus for unsupervised pre-training of the node attributes in order to achieve optimal inductive performance. We also plan to incorporate the pre-trained node attributes in multiple state-of-the-art link prediction models for improved inductive test performance. Further-

more, we plan to incorporate the Davies-Bouldin score on the node attributes in the pre-training objective function for obtaining the optimal set of node attributes. Overall, our work develops a method for selecting the best node features for improving the transferability of the link prediction models for newly arrived nodes.

REFERENCES

Brian Ball and M.E.J. Newman. Friendship networks and social status. *Network Science*, 1(1): 16–30, apr 2013. doi: 10.1017/nws.2012.4. URL https://doi.org/10.1017%2Fnws.2012.4.

Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies*, 1(1):396–413, February 2020. doi: 10.1162/qss_a_00021. URL https://doi.org/10.1162/qss_a_00021.

Yanjun Qi, Ziv Bar-Joseph, and Judith Klein-Seetharaman. Evaluation of different biological data and computational classification methods for use in protein interaction prediction. *Proteins: Structure, Function, and Bioinformatics*, 63(3):490–500, January 2006. doi: 10.1002/prot.20865. URL https://doi.org/10.1002/prot.20865.

Giuliano Andrea Pagani and Marco Aiello. The power grid as a complex network: a survey, 2011. URL https://arxiv.org/abs/1105.3338.

Oriol Lordan and Jose M. Sallan. Dynamic measures for transportation networks. *PLOS ONE*, 15 (12):e0242875, December 2020. doi: 10.1371/journal.pone.0242875. URL https://doi.org/10.1371/journal.pone.0242875.

David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 2007. doi: 10.1002/asi.20591. URL https://doi.org/10.1002/asi.20591.

Zachary Stanfield, Mustafa Coşkun, and Mehmet Koyutürk. Drug response prediction as a link prediction problem. *Scientific Reports*, 7(1), January 2017. doi: 10.1038/srep40321. URL https://doi.org/10.1038/srep40321.

T. Jaya Lakshmi and S. Durga Bhavani. Link prediction approach to recommender systems, 2021. URL https://arxiv.org/abs/2102.09185.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, jan 2016. doi: 10.1109/jproc.2015.2483592. URL https://doi.org/10.1109%2Fjproc.2015.2483592.

Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, July 2003. doi: 10.1016/s0378-8733(03)00009-1. URL https://doi.org/10.1016/s0378-8733(03)00009-1.

Ajay Kumar, Shashank Sheshar Singh, Kuldeep Singh, and Bhaskar Biswas. Link prediction techniques, applications, and performance: A survey. *Physica A: Statistical Mechanics and its Applications*, 553:124289, September 2020. doi: 10.1016/j.physa.2020.124289. URL https://doi.org/10.1016/j.physa.2020.124289.

Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM, pages 891–900. ACM, 2015.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks, 2016. URL https://arxiv.org/abs/1607.00653.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: Online Learning of Social Representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 701–710. ACM, 2014. doi: 10.1145/2623330.2623732. URL https://doi.org/10.1145%2F2623330.2623732.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE: Large-scale Information Network Embedding. In *Proceedings of the 24th International Conference on World Wide Web*, WWW, pages 1067–1077, 2015. doi: 10.1145/2736277.2741093. URL https://doi.org/10.1145%2F2736277.2741093.

Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 1225–1234. ACM, 2016. doi: 10.1145/2939672.2939753. URL https://doi.org/10.1145/2939672.2939753.

Amir Ghasemian, Homa Hosseinmardi, Aram Galstyan, Edoardo M. Airoldi, and Aaron Clauset. Stacking models for nearly optimal link prediction in complex networks. *Proceedings of the National Academy of Sciences*, 117:23393–23400, 2020. URL https://www.pnas.org/doi/pdf/10.1073/pnas.1914950117.

Baole Ai, Zhou Qin, Wenting Shen, and Yong Li. Structure enhanced graph neural networks for link prediction, 2022. URL https://arxiv.org/abs/2201.05293.

Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. Deep graph library: A graph-centric, highly-performant package for graph neural networks, 2019. URL https://arxiv.org/abs/1909.01315.

F. Scarselli, M. Gori, Ah Chung Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, January 2009. doi: 10.1109/tnn.2008.2005605. URL https://doi.org/10.1109/tnn.2008.2005605.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs, 2020. URL https://arxiv.org/abs/2005.00687.

Ivan Maksimov, Rodrigo Rivera-Castro, and Evgeny Burnaev. Addressing cold start in recommender systems with hierarchical graph neural networks, 2020. URL https://arxiv.org/abs/2009.03455.

Ayan Chatterjee, Robin Walters, Zohair Shafi, Omair Shafi Ahmed, Michael Sebek, Deisy Gysi, Rose Yu, Tina Eliassi-Rad, Albert-László Barabási, and Giulia Menichetti. Ai-bind: Improving binding predictions for novel protein targets and ligands, 2021. URL https://arxiv.org/abs/2112.13168.

Giulia Menichetti. An AI pipeline to investigate the binding properties of poorly annotated molecules. *Nature Reviews Physics*, 2022. doi: 10.1038/s42254-022-00471-1. URL https://doi.org/10.1038/s42254-022-00471-1.

Stephen Bonner, Ufuk Kirik, Ola Engkvist, Jian Tang, and Ian P Barrett. Implications of topological imbalance for representation learning on biomedical knowledge graphs. *Briefings in Bioinformatics*, jul 2022. doi: 10.1093/bib/bbac279. URL https://doi.org/10.1093%2Fbib%2Fbbac279.

Ehsaneddin Asgari and Mohammad R. K. Mofrad. Protvec: A continuous distributed representation of biological sequences, 2015. URL https://arxiv.org/abs/1503.05140.

Albert-László Barabási. *Network Science*. Cambridge University Press, 2016. ISBN 9781107076266.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs, 2017. URL https://arxiv.org/abs/1706.02216.

Thorsten Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference on Machine Learning*, ICML, pages 200–209, 1999.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings, 2016. URL https://arxiv.org/abs/1603.08861.

Komal K. Teru, Etienne Denis, and William L. Hamilton. Inductive relation prediction by subgraph reasoning, 2019. URL https://arxiv.org/abs/1911.06962.

Yu Hao, Xin Cao, Yixiang Fang, Xike Xie, and Sibo Wang. Inductive link prediction for nodes having only attribute information. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, IJCAI, pages 1209–1215, 2020. doi: 10.24963/ijcai.2020/168. URL https://doi.org/10.24963%2Fijcai.2020%2F168.

Mikhail Galkin, Max Berrendorf, and Charles Tapley Hoyt. An open challenge for inductive link prediction on knowledge graphs, 2022. URL https://arxiv.org/abs/2203.01520.

Pim van der Hoorn, Gabor Lippner, and Dmitri Krioukov. Sparse maximum-entropy random graphs with a given power-law degree distribution. *Journal of Statistical Physics*, 173(3-4): 806–844, oct 2017. doi: 10.1007/s10955-017-1887-7. URL https://doi.org/10.1007%2Fs10955-017-1887-7.

Agata Fronczak. Exponential random graph models, 2012. URL https://arxiv.org/abs/1210.7828.

Giulia Menichetti and Daniel Remondini. Entropy of a network ensemble: Definitions and applications to genomic data. *Theoretical Biology Forum*, 107(1-2):77–87, 2014. ISSN 00356050.

Giulia Menichetti, Ginestra Bianconi, Gastone Castellani, Enrico Giampieri, and Daniel Remondini. Multiscale characterization of ageing and cancer progression by a novel network entropy measure. *Molecular bioSystems*, 11(7):1824–31, 2015. URL http://pubs.rsc.org/en/content/articlehtml/2015/mb/c5mb00143a.

Giulia Menichetti, Daniel Remondini, Pietro Panzarasa, Raúl J. Mondragón, and Ginestra Bianconi. Weighted multiplex networks. *PLoS ONE*, 9(6):e97857, June 2014. doi: 10.1371/journal.pone.0097857. URL https://doi.org/10.1371/journal.pone.0097857.

Chuxiong Sun, Jie Hu, Hongming Gu, Jinpeng Chen, and Mingchuan Yang. Adaptive graph diffusion networks, 2020. URL https://arxiv.org/abs/2012.15024.

Jingsong Lv, Zhao Li, Hongyang Chen, Yao Qi, and Chunqi Wu. Path-aware siamese graph neural network for link prediction, 2022. URL https://arxiv.org/abs/2208.05781.

Zhitao Wang, Yong Zhou, Litao Hong, Yuanhang Zou, Hanjing Su, and Shouzhi Chen. Pairwise learning for neural link prediction, 2021. URL https://arxiv.org/abs/2112.02936.

Andrew Stolman, Caleb Levy, C. Seshadhri, and Aneesh Sharma. Classic graph structural features outperform factorization-based graph embedding methods on community labeling. In *Proceedings of the 2022 SIAM International on Conference on Data Mining*, SDM, pages 388–396. SIAM, 2022. URL https://arxiv.org/abs/2201.08481.

Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences*, 101(9):2658–2663, February 2004. doi: 10.1073/pnas.0400054101. URL https://doi.org/10.1073/pnas.0400054101.

M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, June 2006. doi: 10.1073/pnas.0601602103. URL https://doi.org/10.1073/pnas.0601602103.

Jaewon Yang, Julian McAuley, and Jure Leskovec. Community detection in networks with node attributes. In *Proceedings of the 13th IEE International Conference on Data Mining*, ICDM, pages 1151–1156. IEEE, 2013. doi: 10.1109/icdm.2013.167. URL https://doi.org/10.1109%2Ficdm.2013.167.

David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, April 1979. doi: 10.1109/tpami.1979.4766909. URL https://doi.org/10.1109/tpami.1979.4766909.

J. Macqueen. Some methods for classification and analysis of multivariate observations. In *In the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML, pages 1073–1080. ACM, 2009. doi: 10.1145/1553374.1553511. URL https://doi.org/10.1145/1553374.1553511.

A Bairoch. The SWISS-PROT protein sequence data bank and its new supplement TREMBL. *Nucleic Acids Research*, 24(1):21–25, January 1996. doi: 10.1093/nar/24.1.21. URL https://doi.org/10.1093/nar/24.1.21.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013. URL https://arxiv.org/abs/1301.3781.

Sabrina Jaeger, Simone Fulle, and Samo Turk. Mol2vec: Unsupervised machine learning approach with chemical intuition. *Journal of Chemical Information and Modeling*, 58(1):27–35, January 2018. doi: 10.1021/acs.jcim.7b00616. URL https://doi.org/10.1021/acs.jcim.7b00616.

John J. Irwin, Teague Sterling, Michael M. Mysinger, Erin S. Bolstad, and Ryan G. Coleman. ZINC: A free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling*, 52 (7):1757–1768, June 2012. doi: 10.1021/ci3001277. URL https://doi.org/10.1021/ci3001277.

A. Gaulton, L. J. Bellis, A. P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani, and J. P. Overington. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, 40(D1):D1100–D1107, September 2011. doi: 10.1093/nar/gkr777. URL https://doi.org/10.1093/nar/gkr777.

MohammadHossein Bateni, Soheil Behnezhad, Mahsa Derakhshan, MohammadTaghi Hajiaghayi, Raimondas Kiveris, Silvio Lattanzi, and Vahab S. Mirrokni. Affinity clustering: Hierarchical clustering at scale. In *Neural Information Processing Systems*, NIPS, pages 6864–6874, 2017. URL https://proceedings.neurips.cc/paper/2017/file/2e1b24a664f5e9c18f407b2f9c73e821-Paper.pdf.

Jaroslav Nešetřil, Eva Milková, and Helena Nešetřilová. Otakar borůvka on minimum spanning tree problem translation of both the 1926 papers, comments, history. *Discrete Mathematics*, 233(1-3):3–36, April 2001. doi: 10.1016/s0012-365x(00)00224-7. URL https://doi.org/10.1016/s0012-365x(00)00224-7.

APPENDIX

A INDUCTIVE TEST USING THE DEFAULT OGB SPLIT

The default train-validation-test splits in the OGB link prediction benchmark have overlapping nodes. Therefore, when we remove the edges from the training graph which share nodes with the test graph, we lose majority of the edges.

Table 6: We lose majority of the edges while creating inductive tests on the default OGB train-validation-test split.

Dataset	Train Nodes	Validation Nodes	Test Nodes	Train − Test Nodes	Test − Train Nodes
ogbl-ppa	576,289	276,199	576,071	0	0
ogbl-collab	235,868	144,942	143,679	0	0
ogbl-ddi	3,967	3,995	1,737	86	0

B EXPLORING METHODS OF INDUCTIVE SPLIT

Here we explore and summarize the advantages and the limitations of different methods of creating inductive train-validation-test splits:

- Random Edge Split
- Spectral Clustering-based split
- k-Inductive Tests
- Affinity Clustering-based Split

B.1 RANDOM EDGE SPLIT

The recently published benchmark (ILPC) (Galkin et al., 2022) uses random edge split in creating the inductive test setting. In this method, we first sample two sets of edges for the validation and the test graphs. The remaining edges comprise the training graph. The ratio of edges in the training, the validation, and the test graphs typically is 80:10:10. Then we remove the edges from the validation graph which include the nodes from the test graph. Finally, we remove the edges from the training graph which contain the nodes from both the test and the validation graphs.

We observe a large overlap between the train-validation-test nodes in the OGB link prediction benchmark datasets under random edge split. This happens because the hubs contribute to most of the links in the graph and are present in all of the three splits. Table 7 summarizes the training edges obtained using this process, and the edges lost in the training dataset after removing the validation and the test nodes.

Table 7: Train-validation-test split using random edge split on the OGB benchmark.

Dataset	Train Nodes	Validation Nodes	Test Nodes
ogbl-ppa	3,991	36,778	461,288
ogbl-collab	36,593	62,561	60,023
ogbl-ddi	0	3,759	508

Dataset	Train Edges	Validation Edges	Test Edges	Edges lost
ogbl-ppa	2,626	25,973	1,213,051	29,084,623
ogbl-collab	27,482	51,419	42,680	1,281,488
ogbl-ddi	0	53,396	5	445,109

B.2 SPECTRAL CLUSTERING-BASED SPLIT

In this approach, since the edges dropped are not based on the node overlap between train-validation-test sets as random edge split, the number of the lost edges is reduced. We use spectral clustering to divide the graph into three connected components. The component with the most number of links is used for training. The component with the second most links is used as the inductive test graph, and the remaining component is used for validation. This process of splitting the data reduces the edges lost in the process, and creates connected components for train, validation, and test. But, the clusters obtained in this process are highly imbalanced. Table 8 summarizes the edge split obtained using this method.

B.3 K-INDUCTIVE TESTS

In this set-up, we randomly sample two sets of edges for the validation and the test graphs. The remaining edges form the training graph. The ratio of edges in the training, the validation, and the test graphs typically is 80:10:10. Next, we obtain the nodes in the train and the validation datasets which are k-hops away from the nodes in the test dataset. Finally, we remove all the edges from the train and the validation datasets involving the nodes derived in the previous step. This method has similar limitations as random edge split. We lose majority of the nodes and the links in both the training and the validation datasets.

Table 8: Train-validation-test split using spectral clustering-based split on the OGB benchmark.

Dataset	Train Nodes	Validation Nodes	Test Nodes
ogbl-ppa	-	-	-
ogbl-collab	4,254	5	8
ogbl-ddi	1,472	293	2,233

Dataset	Train Edges	Validation Edges	Test Edges	Edges lost
ogbl-ppa	-	-	-	-
ogbl-collab	1,334,827	0	13	49
ogbl-ddi	134,936	9,720	136,231	119,580

B.4 AFFINITY CLUSTERING-BASED SPLIT

We use random node split for obtaining the train-validation-tests splits in OGB-Inductive. Random node split minimizes the edges lost in the split process. However, we often obtain disconnected graphs after a random node split. In order to obtain connected graphs in train, validation, and test, we use affinity clustering (Bateni et al., 2017) to divide the graph into three balanced clusters. Affinity clustering is used for distributed processing on large graphs, and uses Boruvka's Minimum Spanning Tree algorithm (Nešetřil et al., 2001) for creating the clusters.