On the Topology Awareness and Generalization Performance of Graph Neural Networks

Junwei Su and Chuan Wu†

Department of Computer Science, the University of Hong Kong
{jwsu,cwu}@cs.hku.hk

Abstract. Many computer vision and machine learning problems are modelled as learning tasks on graphs, where graph neural networks (GNNs) have emerged as a dominant tool for learning representations of graph-structured data. A key feature of GNNs is their use of graph structures as input, enabling them to exploit the graphs' inherent topological properties—known as the topology awareness of GNNs. Despite the empirical successes of GNNs, the influence of topology awareness on generalization performance remains unexplored, particularly for node-level tasks that diverge from the assumption of data being independent and identically distributed (I.I.D.). The precise definition and characterization of the topology awareness of GNNs, especially concerning different topological features, are still unclear. This paper introduces a comprehensive framework to characterize the topology awareness of GNNs across any topological feature. Using this framework, we investigate the effects of topology awareness on GNN generalization performance. Contrary to the prevailing belief that enhancing the topology awareness of GNNs is always advantageous, our analysis reveals a critical insight: improving the topology awareness of GNNs may inadvertently lead to unfair generalization across structural groups, which might not be desired in some scenarios. Additionally, we conduct a case study using the intrinsic graph metric, the shortest-path distance, on various benchmark datasets. The empirical results of this case study confirm our theoretical insights. Moreover, we demonstrate the practical applicability of our framework by using it to tackle the cold start problem in graph active learning.

1 Introduction

Many problems in computer vision and machine learning are modeled as learning tasks on graphs. For example, in semantic segmentation, graphs model the relationships between different image regions, enhancing accuracy and context-aware segmentation. Graph neural networks (GNNs) have emerged as a dominant class of machine learning models specifically designed for learning representations of graph-structured data. They have demonstrated considerable success in addressing a wide range of graph-related problems in various domains such as chemistry [10], biology [41], social networking [6,22,35], scene graph generation [49,54]

†: corresponding authors

and visual relationship detection [24,46,52]. A defining characteristic of GNNs is their use of a spatial approach through message passing on the graph structure for feature aggregation. This enables GNNs to preserve structural information or dependencies (referred to as *topology awareness*) from the underlying graph structure, allowing them to be highly effective in tasks such as node classification. Fig. 1 illustrates the overall learning process of GNNs.

Despite their practicality and potential, there remains a lack of theoretical understanding about GNNs, particularly in the semi-supervised node classification setting where the dependencies among the data differ significantly from other machine learning models [25]. In this setting, the goal is to leverage relations, as captured by the graph structure, among the data and a small set of labelled nodes to predict labels for the remaining nodes. Most of the existing theoretical studies of GNNs have focused on the connection between the message-passing mechanism of GNNs and the Weisfeiler-Lehman isomorphism test [19], aiming to understand the ability of GNNs to differentiate different graph structures in the learned representations, known as the expressive power of GNNs. Inspired by the expressiveness studies, it is commonly believed that increasing topology awareness is universally beneficial and many studies focus on enabling GNNs to preserve more structural properties in the learned representation [28, 32, 51].

However, as GNNs become more reliant on and sensitive (aware) of graph structure as input, they may exhibit different generalization performances towards certain *structural subgroups* (distinct data subsets grouped by structural similarity to the training set) within the data. The quantification of GNN generalization across distinct structural subgroups is termed *structural subgroup generalization* [25]. Such considerations are vital in GNN application and development. For instance, within protein-protein interaction networks, these structural subgroups could represent different molecular complexes, influencing the accuracy of interaction predictions. Similarly, understanding how the topology awareness of GNNs influences generalization is essential when devising sampling strategies for training. The extent to which the generalization performance of GNNs is influenced by specific structural features of graph data is critical in deciding the composition of training datasets. Despite its importance, an understanding of the relationship between the topology awareness of GNNs and its structural subgroup generalization is still lacking. Furthermore, characterizing the topology awareness of GNNs poses a challenge, especially considering that different domains and tasks may prioritize distinct structural aspects. Therefore, a versatile framework is needed to assess the topology awareness of GNNs in relation to various structures.

To address this gap, in this paper, we propose a novel framework based on approximate metric embedding to study the relationship between structural subgroup generalization and topology awareness of GNNs in the context of semi-supervised node classification. The proposed framework allows for the investigation of the structural subgroup generalization of GNNs with respect to different structural subgroups. More concretely, the main contributions of this work are summarized as follows.

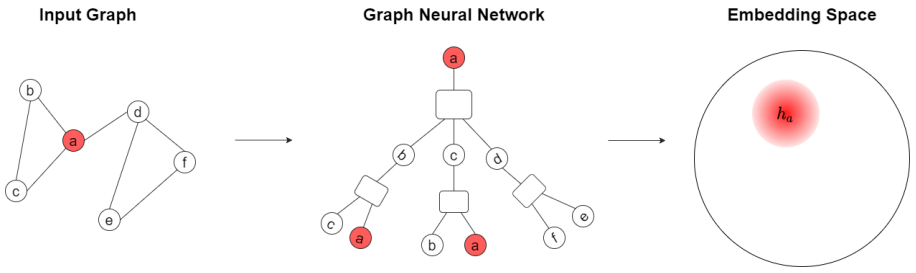


Fig. 1. An illustration of the learning process in a 2-layer GNN. The message-passing mechanism leverages the graph structure to aggregate information, thereby generating the representation/embedding h_a for the target vertex a (highlighted in red).

1. We propose a novel, structure-agnostic framework using approximate metric embedding to examine the interplay between GNNs' structural subgroup generalization and topology awareness. This framework is versatile, accommodating various structural measures like shortest-path distance, and requires only the corresponding structural measure. Its simplicity in estimating key factors makes it applicable and generalizable to a wide range of scenarios.
2. Through formal analysis within our framework, we establish a clear link between GNN topology awareness and their generalization performance (Theorem 1). We also demonstrate that while enhanced topology awareness boosts GNN expressiveness, it can result in uneven generalization performance, favouring subgroups more structurally similar to the training set (Theorem 2). Such structural property can be harmful (causing unfairness issues) or useful (informing design decisions) depending on the scenario. This challenges the prevailing belief that increased topology awareness universally benefits GNNs [28, 32, 51], emphasizing the importance of considering the relationship between topology awareness and generalization performance.
3. We validate our framework through a case study on shortest-path distance, highlighting its practicality and relevance. The results corroborate our theoretical findings, showing that GNNs with heightened awareness of shortest-path distances excel in classifying vertex groups closer to the training set. Moreover, we demonstrate how our findings can be applied to mitigate the cold start problem in graph active learning [11,15], highlighting the practical implications of our framework and results.

2 Related Work

2.1 Topology Awareness of GNNs

Modern GNN architectures take inspiration from the Weisfeiler-Lehman isomorphism test [12,19], which employs the graph structure to propagate information. Consequently, a significant portion of existing literature on GNNs concentrates

on understanding their ability to differentiate various graph structures, known as the expressive power of GNNs [27–29, 47, 51, 53]. Inspired by the expressiveness studies, it is commonly believed that increasing topology awareness is universally beneficial and many studies focus on enabling GNNs to preserve more structural properties [37, 38] in the learned representation [28, 32, 51]. Nevertheless, the precise ramifications of heightened topology awareness on GNNs' generalization performance remain shrouded in ambiguity. The intricate variations within structures across different domains and tasks further compound this complexity, underscoring the necessity for a unified framework—akin to the one proposed in this paper—to shed light on these intricacies.

2.2 Generalization of GNNs

Relatively few studies focus on understanding the generalization performance of GNNs in node-level tasks. [3,9,18,23,25,33,39,40,44,48] extend the conventional statistical learning or information-theoretic framework to GNNs, providing different generalization bounds based on concepts such as Vapnik–Chervonenkis dimension, mutual information, and algorithm stability. In light of this, despite empirical indications highlighting the critical role of GNNs' topology awareness in generalization performance [20,28,51], there remains limited understanding of the relationship between topology awareness of GNNs and their generalization. Our proposed framework addresses this gap by utilizing the distortion concept in the approximate metric embedding to connect the topology awareness of GNNs and their generalization performance.

2.3 Active Learning in Graph

Active Learning (AL) seeks to enhance model performance within a limited labeling budget by identifying the most representative samples for annotation [31]. The AL methodology generally involves two phases: 1) selecting a small, preliminary subset of samples for labeling; 2) utilizing this labeled subset to iteratively discover the most informative sample for future labeling. The initial selection of an effective subset represents the cold-start problem in AL [11]. While AL has proven effective in settings with independent and identically distributed (i.i.d.) data, such as images and text, applying it to Graph Neural Networks (GNNs) poses unique challenges. These challenges stem from the necessity to account for node interdependencies and the computational complexity of iterative sample selection and retraining, especially when considering multi-hop neighbors. Recent AL methodologies for GNNs, such as AGE [5], FeatProp [45], and GraphPart [26], aim to overcome these obstacles by leveraging the graph structure and initial node features to boost efficiency and efficacy, although their success may significantly hinge on the quality of the initial features. In this work, we focus on the cold-start problem of graph active learning and introduce a novel approach (based on our analysis) that relies exclusively on graph structure, thus eliminating the reliance on feature quality at the onset of learning.

3 Framework

In this section, we introduce our proposed framework, which aims to investigate the intricate relationship between topology awareness and generalization performance in GNNs. The main idea of the framework is to use metric distortion to characterize the topology awareness of GNNs. With this characterization, we are able to show the relation between topology awareness of GNNs and their generalization performance.

3.1 GNN Preliminaries

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ be an undirected graph where $\mathcal{V} = \{1, 2, ..., n\}$ is the set of n nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. Let $\mathbf{X} = \{x_1, x_2, ..., x_n\}$ denote the set of node attribute vectors corresponding to all the nodes in $\mathcal{V}$. More specifically, $x_v \in \mathbf{X}$ is an m-dimensional vector which captures the attribute values of node $v \in \mathcal{V}$. In addition, each vertex $v \in \mathcal{V}$ is associated with a node label $y_v \in \{1, ..., c\}$, where c is the total number of classes.

For our analysis, we focus on the transductive node classification setting [50], where the graph $\mathcal{G}$ are observed prior to learning and labels are only available for a small subset of nodes $\mathcal{V}_0 \subset \mathcal{V}$. Without loss of generality, we treat $\mathcal{G}$ as fixed throughout our analysis and the randomness comes from the labels y. Given a small set of the labelled nodes, $\mathcal{V}_0 \subset \mathcal{V}$, the task of node-level semi-supervised learning is to learn a classifier f, such that $\hat{y_v} = f(v)$, from a hypothesis space $\mathcal{F}$ and perform it on the remaining unlabeled nodes. Let $\mathcal{L}(.,.) \mapsto \mathbb{R}_+$ be a bounded continuous loss function. We denote by $R^{\mathcal{L}}(f, \mathcal{V})$ the generalization risk of the classifier f with respect to $\mathcal{L}$ and $\mathcal{V}$ and it is defined as follows,

$$R^{\mathcal{L}}(f, \mathcal{V}) = \mathbb{E}_{v \sim \mathcal{V}}[\mathcal{L}(f(v), y_v)] \tag{1}$$

In our case, the hypothesis space is captured by a GNN and an additional read-out function. GNNs are commonly used as an embedding function and adopt a neighbourhood aggregation scheme, where the representation of a node is learnt iteratively by aggregating representations of its neighbours. After k iterations of aggregation, a node's representation captures the information within its k-hop neighbourhood in the graph. Formally, the k-th layer of a GNN can be written as follows,

$$h_u^{(k)} = \text{Update}^{(k)}\big(\text{Aggregate}^{(k)}(\{h_v^{(k-1)}, \forall v \in \mathcal{N}(u) \cup \{u\}\})\big), \tag{2}$$

where $h_u^{(k)}$ is the representation vector of node u at the k-th layer. $h_u^{(0)}$ is initialized to be x_u, and $\mathcal{N}(u)$ is the set of nodes directly adjacent to u. Different GNNs are compositions of aggregation and update layers of different designs. A GNN is commonly interpreted as an embedding function that combines the graph structure and node features to learn a representation vector h_v of node v. Then the learnt representation is fed into a task-specific read-out function g (commonly a multilayer perceptron) for the prediction. g maps the embedding

vector h_v to a label y_v in the label set $\mathbb{R}^c$. The overall GNN prediction process can be viewed as a composition of $g \circ \mathcal{M} : \mathcal{V} \mapsto \mathbb{R}^c$. The overall GNN prediction process can be viewed as a composition of $g \circ \mathcal{M} : \mathcal{V} \mapsto \mathbb{R}^c$.

3.2 Structural Subgroup

To study the generalization performance of GNNs on different structural subgroups, we assume the existence of a generalized distance (structural measure) function, $d_s(u, v)$, that captures the similarity between two nodes u, v with respect to structure s. Furthermore, we assume that this structural measure function $d_s(u, v)$ and the vertex set $\mathcal{V}$ form a metric space. A metric space is a fundamental mathematical object that allows for studying and measuring the relation among the elements in a given set, and it is formally defined as follows.

Definition 1 (metric space). *A metric space is a pair $(\mathcal{Q}, d)$, where $\mathcal{Q}$ is a set and $d : \mathcal{Q} \times \mathcal{Q} \mapsto \mathbb{R}$ is a metric satisfying the following axioms (x, y, z are arbitrary points of $\mathcal{Q}$):*

$$
\begin{aligned}
&\textit{(M1)}\ d(x, y) \geq 0,\\
&\textit{(M2)}\ d(x, x) = 0,\\
&\textit{(M3)}\ d(x, y) > 0, x \neq y,\\
&\textit{(M4)}\ d(x, y) = d(y, x),\\
&\textit{(M5)}\ d(x, y) + d(y, z) \geq d(x, z).
\end{aligned}
$$

These properties ensure that the metric $d(.)$ can be used to quantify the distance or dissimilarity between any two elements in the set $\mathcal{Q}$. In the context of graph learning, the metric space $(\mathcal{V}, d_s)$ allows us to study the similarity between nodes in terms of the structural property s and measure the structural distance between different subgroups of the test set. One natural and commonly used example in the graph is the metric space formed by the vertex set and the shortest-path distance.

Definition 2 (Structural Group Distance). *Given a metric space $(\mathcal{V}, d_s)$ with respect to structure s and two distinct sets (groups) $\mathcal{V}_1, \mathcal{V}_2 \subset \mathcal{V}$, we define the structural distance D_s between an element $v \in \mathcal{V}$ to a group as*

$$D_s(v, \mathcal{V}_1) = \min_{u \in \mathcal{V}_1} d_s(v, u) \tag{3}$$

and the structural distance between the two groups as

$$D_s(\mathcal{V}_1, \mathcal{V}_2) = \max_{v \in \mathcal{V}_1} \min_{u \in \mathcal{V}_2} d_s(v, u) \tag{4}$$

It should be noted that the distance function in Def. 2 is not symmetrical, and Eq. 4 captures the maximal structural difference of vertices in $\mathcal{V}_1$ to $\mathcal{V}_2$. To provide further insight into the concept of a structural subgroup, let's

consider a scenario where $\mathcal{V}_0 \subset \mathcal{V}$ represents the training set and $d_s(.)$ represents the shortest-path distance. In this context, we can partition the remaining vertices $\mathcal{V}\backslash\mathcal{V}_0$ into different structural subgroups based on their shortest-path distance to the nearest vertex in the training set. Specifically, we can define $\mathcal{V}_k = \{v \in \mathcal{V}\backslash\mathcal{V}_0 | D_s(v, \mathcal{V}_0) = k\}$. However, it is important to emphasize that our analysis does not assume a specific partition scheme, and the framework remains adaptable to various scenarios.

3.3 Topology Awareness

To characterize and formalize the topology awareness of GNNs, we introduce the concept of distortion, which measures the degree to which the graph structure is preserved in the embedding space. Mathematically, the distortion of a function between two general metric spaces is defined as,

Definition 3 (distortion). *Given two metric spaces $(\mathcal{Q}, d)$ and $(\mathcal{Q}', d')$ and a mapping $\gamma : \mathcal{Q} \mapsto \mathcal{Q}'$ ($\mathcal{Q}$ ($\mathcal{Q}'$) is a set and d (d') is a distance function)), γ is said to have a distortion $\alpha \geq 1$, if there exists a constant $r > 0$ such that $\forall u, v \in \mathcal{Q}$,*

$$rd(u, v) \leq d'(\gamma(u), \gamma(v)) \leq \alpha r d(u, v).$$

The definition above captures the maximum expansion and minimum contraction between the two metric spaces. In the extreme case when $\alpha = 1$, the mapping γ is the so-called isometric mapping that perfectly preserves the distance between any two points in the space. In other words, for any two points x and y in the domain of the function, their distance $d(x, y)$ in the original metric space is equal to the distance $d'(\gamma(x), \gamma(y))$ in the target metric space (up to a constant scaling factor). If the distortion rate of the mapping is close to 1, then this mapping is said to have a low distortion rate.

In the context of GNNs, evaluating distortion involves a straightforward comparison of embeddings in the embedding domain with the structural measures on the graph, as demonstrated in the case study. If the distortion rate between the graph domain and the embedding domain is close to 1, the resulting embedding would largely preserve the corresponding structural information from the graph space, thereby indicating high topology awareness. Consequently, distortion can be employed to measure the topology awareness of GNNs concerning the corresponding structural properties (e.g., shortest-path distance) measured in the graph domain. This concept of distortion is agnostic to specific structural configurations, enabling the study of diverse graph structures using appropriate distance functions. Moreover, its estimation can be readily achieved by comparing the embedding with the original structural measure, as illustrated later in the case study.

4 Main Results

In this section, we present the main results derived from our proposed framework for studying the generalization and fairness of GNNs with respect to different

structural groups in a graph. We start by demonstrating the relationship between the generalization performance of the model and the structural distance between the subgroups and the training set (Theorem 1). We then show that such a structural relation can induce disparities in the generalization performance across different subgroups (Theorem 2). Finally, we connect these theoretical results to the recent deep learning spline line theory [2] to provide additional insights. Detailed proof of the results presented in this paper can be found in the supplementary material.

4.1 Generalization Performance and Structural Distance

Theorem 1. *Let $(\mathcal{V}, d_s)$ be a metric space for structure s. Let $\mathcal{V}_0$ be a given labelled training set and $\mathcal{V}_i \subset \mathcal{V} \backslash \mathcal{V}_0$ be an arbitrary test subgroup. Suppose $\mathcal{M}$ is a GNN model with distortion α with respect to the structural measure $d_s(.,.)$, and g and $\mathcal{L}$ are Lipschitz, we have that*

$$R^{\mathcal{L}}(g \circ \mathcal{M}, \mathcal{V}_i) \leq R^{\mathcal{L}}(g \circ \mathcal{M}, \mathcal{V}_0) + \mathcal{O}(\alpha D_s(\mathcal{V}_i, \mathcal{V}_0)). \tag{5}$$

Theorem 1 states that the generalization risk of a classifier with respect to a structural subgroup depends on the empirical training loss of the classifier, the structural distance between the subgroup and the training set, and the distortion (topology awareness) of GNNs. Such a relation provides a new perspective on understanding the link between topology awareness and generalization performance in GNNs. Prior research in this area has mainly focused on the connection between GNNs and the Weisfeiler-Lehman algorithm, emphasizing the ability of GNNs to differentiate various graph structures. However, Theorem 1 highlights a direct link between topology awareness and generalization performance, and a potential issue of fairness in GNNs, as it shows that the model's performance on different structural subgroups is influenced by their distance to the training set. Next, we extend the result of Theorem 1 to analyze the relation between the generalization of different structural subgroups and their distance to the training set.

4.2 Unfair Generalization Performance on Structural Subgroups

Theorem 2. *Let $(\mathcal{V}, d_s)$ be a metric space for structure s. Suppose $\mathcal{M}$ is a GNN model with small training loss on $\mathcal{V}_0$ and distortion α with respect to structural measure $d_s(.,.)$,and $\mathcal{L}$ and g are smooth. Let $\mathcal{V}_i, \mathcal{V}_j \subset \mathcal{V} \backslash \mathcal{V}_0$ be two distinct subgroups in the local neighbourhood of the training set $\mathcal{V}_0$, then there exists constant $\delta > 0$ (independent of the choice of subgroups) such that if*

$$D_s(\mathcal{V}_i, \mathcal{V}_0) \geq \alpha\delta D_s(\mathcal{V}_j, \mathcal{V}_0),$$

we have that,

$$R^{\mathcal{L}}(g \circ \mathcal{M}, \mathcal{V}_i) \geq R^{\mathcal{L}}(g \circ \mathcal{M}, \mathcal{V}_j).$$

Theorem 2 states that when comparing two subgroups, if the difference in their structural distance to the training set is larger than the threshold ($\delta\alpha$), then GNNs exhibit unfair generalization performance with respect to these groups. Specifically, the theorem states that GNNs generalize better on subgroups that are structurally closer to the training set, which leads to accuracy disparities among different subgroups. Additionally, the theorem notes that the topology awareness of GNNs (measured by the distortion rate α) plays a key role in determining the strength of this accuracy disparity. This has important implications for the use of GNNs in critical applications, as it suggests that increased topology awareness in GNNs can lead to improved generalization performance but also a greater potential for unfairness in terms of accuracy disparities among different structural subgroups. As such, it is important to carefully consider the trade-offs between topology awareness and fairness when designing new GNNs and applying them to real-world problems.

4.3 Connection with Deep Learning Spline Theory

Our theoretical findings reveal that when considering the performance of the model from the viewpoint of the embedding space, the training set creates a "valley" effect on generalization performance, with the training vertex at the bottom of the valley. Representations that are closer to the embedding of the training set perform better, while those farther away perform worse. This pattern has an interesting connection with deep learning spline theory [2]. This theory suggests that in DNNs, the learning process for classification divides the input/latent space into small circles (referred to as power diagrams), where representations within each circle are grouped, and representations closer to the centre of the circle are predicted with more confidence. Our findings suggest that in GNNs, the mapping of test data to these power diagrams is related to the structural similarity between test and training data in the graph. This connection provides further validation for our derived results.

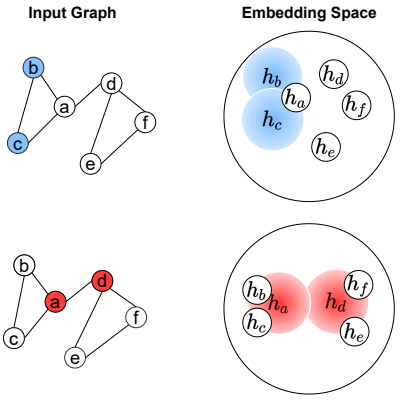


Fig. 2. The loss landscape in embedding space induced by different training sets. When vertices a and d are selected as the training set instead of b and c, the test losses are smaller as the average distance between the training set and the other vertices is smaller (the neighbourhoods of h_a and h_d cover embeddings of more vertices in the embedding space).

5 A Case Study on Shortest-Path Distance

In this section, we present a case study which utilizes the shortest path distance to validate the predictions and effectiveness of our proposed framework. We aim to demonstrate how our framework can be applied to a real-world scenario and how it can be used to understand the generalization performance of GNNs on different structural subgroups. The case study will provide insights and show the potential impact of our results on various critical applications of GNNs. In this case study, we aim to answer empirically the following questions: Given a GNN model that can preserve certain structure s (α closed to 1), is the derived structural result observed on the generalization performance of the GNN model with respect to structural subgroups defined by s? Furthermore, how can our outcomes be harnessed to effectively tackle real-world challenges?

Overall, our case study demonstrates that our proposed framework can be applied to a real-world scenario and provides useful insights into the performance of GNNs in different structural subgroups. It also shows that the results of our framework can be generalized for different train/test pairs. Moreover, we showcase its practical utility by employing the findings to address the cold start problem within graph active learning.

General Setup. We use Deep Graph Library [43] for our implementation and experiment with four representative GNN models, GCN [17], Sage [13], GAT [42] and GCNII [7]. In all experiments, we follow closely the common GNN experimental setup, and multiple independent trials are conducted for each experiment to obtain average results. We adopt five popular and publicly available node classification datasets: Cora [34], Citeseer [21], CoraFull [4], CoauthorCS [36] and Ogbn-arxiv [16]. Due to space limitations, we only show part of the experimental results in the main body of the paper. Detailed descriptions of the datasets, hyperparameters and remaining experimental results can be found in the appendix.

Structural Subgroups. We examine the accuracy disparity of subgroups defined by the shortest path distance (spd). In order to directly investigate the effect of structural distance on the generalization bound (Theorem 1), we first split the test nodes into subgroups by their shortest-path distance to the training set. For each test node v, its distance to the training set $\mathcal{V}_0$ is given by $d_v = D_{\text{spd}}(v, \mathcal{V}_0)$ with the shortest path distance metric as defined in Def. 2. Then we group the test nodes according to this distance.

Distance Awareness of the Selected GNNs. We first verify if the selected GNNs preserve (are aware of) graph distances (with a small distortion rate), by evaluating the relation between the graph distances of test vertices to the training set and the embedding distances (Euclidean distance) of vertex representations to the representations of the training set, i.e., the relation between $d_{\text{spd}}(v, \mathcal{V}_0)$ and $d_{\text{Euclidean}}(\mathcal{M}(v), \mathcal{M}(\mathcal{V}_0))$. In this experiment, we train the GNN models on the default training set provided in the dataset. We extract vertices within the 5-hop neighbourhood of the training set and group the vertices based on their distances to the training set. 5-hop neighbourhood is chosen because the diameter of most real-life networks is around 5 and it gives a better visual

presentation. We then compute the average embedding distance of each group to the training set in the embedding space. Based on Def. 3, if an embedding has low distortion, we should expect to see linear relation with the slope being the scaling factor. As shown in Fig. 3, there exists a strong correlation between the graph distance and the embedding distance. The relation is near linear and the relative orders of distances are mostly preserved within the first few hops. This implies that the distortion α is small with the selected GNNs, i.e., they are distance aware. It also indicates that the premise of the small distortion in our theoretical results can be commonly satisfied in practice.

5.1 Graph Distance and Generalization Performance

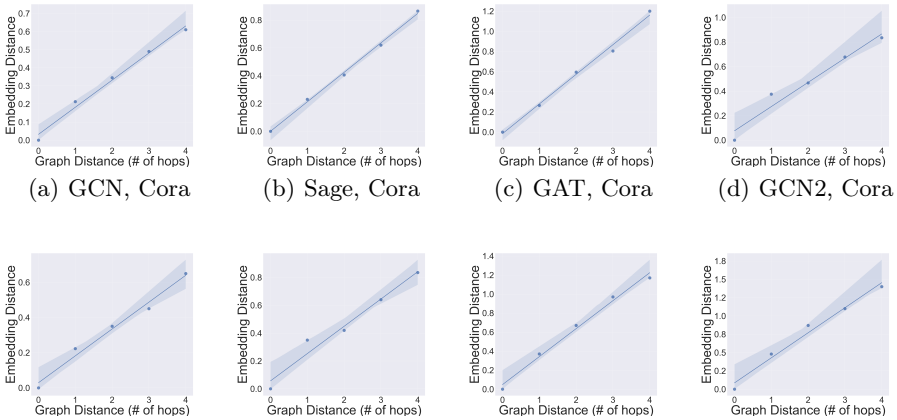
Next, we study the relation between graph distance and generalization performance following the same settings as in the previous experiment. We compute the average prediction accuracy in each group of vertices. In Fig. 4, '0 hop' indicates the training set. We observe that the selected GNNs (with low distortion) indeed tend to generalize better on the group of vertices that are closer to the training set, which validates Theorem 2.

We further validate if the distance between different pairs of the training sets and the test group is related to the generalization performance of GNNs in the test group. We randomly sample $k = 5$ vertices from each class as the training set $\mathcal{V}_0$. Then we evaluate the GNN model on the rest of the vertices $\mathcal{V}\backslash\mathcal{V}_0$. We make k relatively small to enable a larger variance (among different trials) in the distance $D_{\text{spd}}(\mathcal{V}\backslash\mathcal{V}_0, \mathcal{V}_0)$ (as given in Def. 2) between the training set and the rest of the vertices. We conduct 30 independent trials each selecting a different training set and ranking all the trials in descending order of the mean graph distance between the training set and the rest of the vertices. In Table 1, Group 1 contains the 10 trials with the largest graph distance $D_{\text{spd}}(\mathcal{V}\backslash\mathcal{V}_0, \mathcal{V}_0)$ and Group 3 includes the 10 trials with the smallest graph distance $D_{\text{spd}}(\mathcal{V}\backslash\mathcal{V}_0, \mathcal{V}_0)$. The average classification accuracy and the variance (value in the bracket) are computed for each group.

As shown in Table 1, training GNNs on a training set with a smaller graph distance $D_{\text{spd}}(\mathcal{V}\backslash\mathcal{V}_0, \mathcal{V}_0)$ to the rest of the vertices, does generalize better, which validates our result and further enables us to use these insights for designing and developing new GNNs algorithms.

5.2 Application in Cold Start Problem for Graph Active Learning

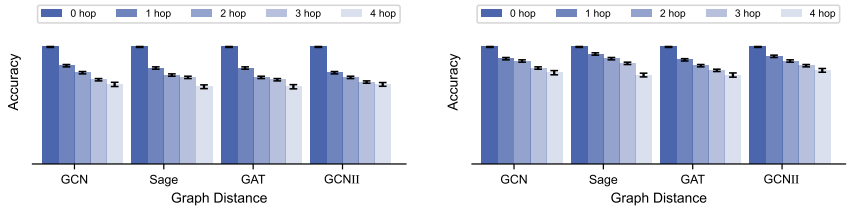
Active learning is a type of machine learning where the model is actively involved in the labelling process of the training data. In active learning, the model is initially trained on a small labelled dataset and then used to select the most informative examples from the unlabeled dataset to be labelled next. Unsatisfactory initial set selection can lead to slow learning progress and unfair performance, which is referred to as the cold start problem [11]. Effective resolution of this issue entails an initial set selection that adheres to the following criteria: 1) exclusively leveraging graph structure; 2) achieving robust generalization; and



(a) GCN, Cora (b) Sage, Cora (c) GAT, Cora (d) GCN2, Cora

(e) GCN, Coauthor CS (f) Sage, Coauthor CS (g) GAT, Coauthor CS (h) GCNII, Coauthor CS

Fig. 3. Graph distance vs. embedding distance. (a)-(d) are the relation between graph distance (shortest-path distance) and embedding distance (Euclidean) of the four representative GNNs on Cora. (e)-(h) are the relation between graph distance and embedding distance of the four representative GNNs on Coauthor CS. The highlighted regions represent the 95% confidence interval for that regression. (a)-(h) all demonstrate a rather linear relation between graph distance and embedding distance.



(a) Cora (b) Coauthor CS

Fig. 4. Graph distance vs. accuracy. (a) is the relation between graph distance (shortest-path distance) and the accuracy of four representative GNNs on Cora. (b) is the relation between graph distance and the accuracy of four representative GNNs on Coauthor CS. Both (a) and (b) show that the graph distance is indeed a good indicator for generalization performance when GNNs are distance aware.

3) ensuring fairness (with minimal discrepancy in generalization performance) across distinct subgroups. We illustrate how our derived findings can be instrumental in the selection of an effective initial labelled dataset.

Our empirical results show that the selected GNNs are highly distance-aware, meaning that their generalization performance is closely related to the coverage of the training set. To ensure that the model is fair and effective, the initial

Table 1. Accuracy of 30 independent trials with different train/test pairs grouped by $D_s(\mathcal{V}\backslash\mathcal{V}_0, \mathcal{V}_0)$. Group 1 contains the 10 trials with the largest graph distance $D_{\text{spd}}(\mathcal{V}\backslash\mathcal{V}_0, \mathcal{V}_0)$ and Group 3 includes the 10 trials with the smallest graph distance $D_{\text{spd}}(\mathcal{V}\backslash\mathcal{V}_0, \mathcal{V}_0)$. The average accuracy and variance of the 10 trials are recorded in the table. The results show that train/test pairs with closer structural similarity indeed have better generalization performance, as predicted by our theoretical results.

	Cora				Citeseer			
	GCN	GAT	Sage	GCNII	GCN	GAT	Sage	GCNII
Group 1	65.42 (5.03)	68.03 (4.62)	67.61 (5.02)	61.42 (4.03)	59.72 (5.41)	56.78 (4.73)	58.52 (4.85)	55.46 (3.69)
Group 2	67.56 (4.23)	70.94 (3.86)	69.54 (4.84)	66.42 (3.76)	61.48 (5.15)	60.33 (4.24)	60.78 (3.60)	58.42 (4.08)
Group 3	**71.40 (3.59)**	**73.52 (3.53)**	**72.60 (4.23)**	**72.18 (4.01)**	**62.93 (4.14)**	**64.63 (3.65)**	**63.03 (4.36)**	**63.23 (5.06)**

Table 2. Initial data labelling: performance comparison of different GNNs sampling strategies based on graph structure. Results are averaged among five trials. For each dataset, we select 0.5% of the total data as the initial set and evaluate on the rest of the vertices. Each entry corresponds to "test accuracy (ACC)| maximum discrepancy (MD)". ↑ means the higher the better and ↓ means the lower the better. **Bold letter** indicates the best sampling strategy for the given dataset and GNN model.

	GCN				Sage			
	Cora	CiteSeer	CoraFull	Coauthor CS	Cora	CiteSeer	CoraFull	Coauthor CS
	ACC(↑)|MD(↓)	ACC(↑)|MD(↓)	ACC(↑)|MD(↓)	ACC(↑)|MD(↓)	ACC(↑)|MD(↓)	ACC(↑)|MD(↓)	ACC(↑)|MD(↓)	ACC(↑)|MD(↓)
R	37.08|13.23	36.35|10.23	8.13|3.23	57.75|16.23	37.69|12.03	37.24|11.20	8.56|4.20	49.72|15.52
Degree	40.87|12.52	36.27|9.28	10.68|2.32	51.50|18.23	38.04|11.87	35.26|12.05	11.56|3.02	42.78|13.02
Centrality	42.75|10.41	38.45|8.15	9.49|2.08	53.48|16.11	42.18|10.15	40.81|9.05	12.56|3.08	45.65|10.11
PageRank	47.87|9.52	42.27|8.25	12.48|2.10	55.50|15.23	50.15|9.14	43.26|9.15	13.15|2.98	55.16|10.74
OURS	**49.03|8.23**	**50.98|7.32**	**15.25|1.23**	**61.75|14.32**	**58.57|8.29**	**48.80|7.03**	**14.76|2.01**	**61.63|8.82**
	GAT				GCNII			
	Cora	CiteSeer	CoraFull	Coauthor CS	Cora	CiteSeer	CoraFull	Coauthor CS
	ACC(↑)|MD(↓)	ACC(↑)|MD(↓)	ACC(↑)|MD(↓)	ACC(↑)|MD(↓)	ACC(↑)|MD(↓)	ACC(↑)|MD(↓)	ACC(↑)|MD(↓)	ACC(↑)|MD(↓)
R	49.36|15.78	43.53|16.27	9.55|3.07	43.25|15.20	31.53|14.02	37.29|17.32	10.98|3.06	52.61|17.02
Degree	54.08|13.52	44.65|15.17	10.50|2.78	47.30|13.04	32.19|13.86	38.88|16.98	10.40|2.79	54.70|15.96
Centrality	56.81|11.52	48.15|13.08	11.15|2.52	48.50|13.18	32.89|12.57	39.98|16.05	12.48|2.52	56.88|15.18
PageRank	60.91|11.08	49.77|12.07	12.94|2.11	50.50|12.23	34.04|12.08	42.77|13.95	14.56|2.19	58.78|14.08
OURS	**62.76|10.54**	**51.11|11.07**	**13.39|1.88**	**51.21|11.02**	**35.61|11.02**	**44.50|13.04**	**15.19|1.96**	**60.76|13.18**

labelled set should consist of vertices that are in close proximity to the rest of the input graph (in graph distance), as illustrated in Fig. 2. This can be formulated into the following optimization problem:

$$\begin{aligned} &\min_{\mathcal{V}_0 \subset \mathcal{V}} D_{\text{spd}}(\mathcal{V}\backslash\mathcal{V}_0, \mathcal{V}_0) \\ &s.t. \quad |\mathcal{V}_0| = k \end{aligned} \tag{6}$$

where k is the given initial set size and $D_s(.)$ is given in Def. 2. Intuitively, Eq.6 aims to select a subset of vertices that exhibit the closest structural similarity (as defined by the graph distance in this case) to the remaining vertices. This problem corresponds to the well-known k-center problem in graph theory [14], which is NP-hard. While finding the optimal solution may be intractable, efficient heuristics exist for tackling Eq. 6. Exploring the best heuristic is beyond the scope of this paper. Instead, for our experiment, we adopt the classic farthest-first traversal algorithm to obtain a solution. We refer to this sample selection procedure as the coverage-based sampling algorithm and provide a detailed description of the method used in the experiment within the supplementary material.

We evaluate the initial set selected by our coverage-based sampling algorithm, comparing it to the existing baselines: (1) **R**: uniform random sampling [30], and importance sampling with graph properties (2) **Degree** [15], (3) **Centrality**, (4) **Density** and (5) **PageRank** [5]. We apply each strategy to select an initial dataset of 0.5% of the total data for labelling (training). We evaluate the performance of the learnt model with the rest of the vertices on two metrics: 1) classification accuracy, which reflects the generalization performance, and 2) the maximum discrepancy among structural groups [1, 8], which reflects the fairness. Let R_i denote the mean classification accuracy of the vertex group that is i-hop away from $\mathcal{V}_0$. Then, the maximum discrepancy is computed as $\max\{|R_i - R_j||i, j \in 1, ..., r\}$, where r is the maximum hop number. We use $r = 5$ to keep consistent with other experiments. The results in Table 2 show that coverage-based sampling achieves a significant improvement in model performance and fairness over the other strategies across different datasets, giving a better kick-start to the active learning process of GNNs.

6 Conclusion and Discussion

In this work, we study and investigate GNNs which are fundamental to many computer vision and machine learning tasks. In particular, we explore the relationship between the topology awareness and generalization performance of GNNs. We introduce a novel framework that connects the structural awareness of GNNs with approximate metric embedding, offering a fresh perspective on their generalization capabilities in semi-supervised node classification tasks. This structure-agnostic framework facilitates a deeper, more intuitive comprehension of GNN generalizability across varied graph structures. Through a case study centered on graph distance, we demonstrate that our theoretical findings regarding structural results are reflected in the practical generalization performance of GNNs. Additionally, our findings shed light on the cold start problem in graph active learning and could influence various significant GNN applications.

6.1 Limitation and Future Works

Our proposed framework introduces a novel angle on GNN generalization performance, yet it is not without limitations. The current approach interprets GNN embeddings in terms of metric mapping and describes the structure awareness of GNNs through distortion. This perspective, however, overlooks the specific dynamics that contribute to reduced distortion within GNNs. Investigating these underlying dynamics would significantly enrich this area of research. Moreover, our study primarily examines the transductive setting. Expanding this analysis to the inductive setting—where the model is trained on one graph but applied to another—would offer a more comprehensive view of GNN generalization capabilities.

Acknowledgement. We would like to thank the anonymous reviewers for their helpful comments. JS and CW are supported by grants from Hong Kong RGC under the contracts HKU 17207621, 17203522 and C7004-22G (CRF).

References

1. Ali, J., Babaei, M., Chakraborty, A., Mirzasoleiman, B., Gummadi, K.P., Singla, A.: On the fairness of time-critical influence maximization in social networks. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE). pp. 1541–1542. IEEE (2022)
2. Balestriero, R., richard baraniuk: A spline theory of deep learning. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 374–383. PMLR (10–15 Jul 2018), https://proceedings.mlr.press/v80/balestriero18b.html
3. Baranwal, A., Fountoulakis, K., Jagannath, A.: Graph convolution for semi-supervised classification: Improved linear separability and out-of-distribution generalization. arXiv preprint arXiv:2102.06966 (2021)
4. Bojchevski, A., Günnemann, S.: Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking (2018)
5. Cai, H., Zheng, V.W., Chang, K.C.C.: Active learning for graph embedding. arXiv preprint arXiv:1705.05085 (2017)
6. Chen, J., Zhu, J., Song, L.: Stochastic training of graph convolutional networks with variance reduction. arXiv preprint arXiv:1710.10568 (2017)
7. Chen, M., Wei, Z., Huang, Z., Ding, B., Li, Y.: Simple and deep graph convolutional networks (2020)
8. Dong, Y., Ma, J., Chen, C., Li, J.: Fairness in graph mining: A survey (2022). https://doi.org/10.48550/ARXIV.2204.09888, https://arxiv.org/abs/2204.09888
9. Esser, P., Chennuru Vankadara, L., Ghoshdastidar, D.: Learning theory can (sometimes) explain generalisation in graph neural networks. Advances in Neural Information Processing Systems **34**, 27043–27056 (2021)
10. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry (2017)
11. Grimova, N., Macas, M., Gerla, V.: Addressing the cold start problem in active learning approach used for semi-automated sleep stages classification. In: 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). pp. 2249–2253. IEEE (2018)
12. Hamilton, W.L.: Graph representation learning. Synthesis Lectures on Artifical Intelligence and Machine Learning **14**(3), 1–159 (2020)
13. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs (2018)
14. Hochbaum, D.S.: Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems. In: Approximation algorithms for NP-hard problems, pp. 94–143 (1996)
15. Hu, S., Xiong, Z., Qu, M., Yuan, X., Côté, M.A., Liu, Z., Tang, J.: Graph policy network for transferable active learning on graphs. Advances in Neural Information Processing Systems **33**, 10174–10185 (2020)
16. Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., Leskovec, J.: Open graph benchmark: Datasets for machine learning on graphs. arXiv preprint arXiv:2005.00687 (2020)
17. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks (2017)

18. Lao, D., Yang, X., Wu, Q., Yan, J.: Variational inference for training graph neural networks in low-data regime through joint structure-label estimation. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 824–834 (2022)
19. Leman, A., Weisfeiler, B.: A reduction of a graph to a canonical form and an algebra arising during this reduction. Nauchno-Technicheskaya Informatsiya **2**(9), 12–16 (1968)
20. Li, P., Wang, Y., Wang, H., Leskovec, J.: Distance encoding: Design provably more powerful neural networks for graph representation learning (2020)
21. Li, Q., Han, Z., Wu, X.M.: Deeper insights into graph convolutional networks for semi-supervised learning. In: Thirty-Second AAAI conference on artificial intelligence (2018)
22. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926 (2017)
23. Liao, R., Urtasun, R., Zemel, R.: A {pac}-bayesian approach to generalization bounds for graph neural networks. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=TR-Nj6nFx42
24. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors (2016)
25. Ma, J., Deng, J., Mei, Q.: Subgroup generalization and fairness of graph neural networks. arXiv preprint arXiv:2106.15535 (2021)
26. Ma, J., Ma, Z., Chai, J., Mei, Q.: Partition-based active learning for graph neural networks. arXiv preprint arXiv:2201.09391 (2022)
27. Morris, C., Ritzert, M., Fey, M., Hamilton, W.L., Lenssen, J.E., Rattan, G., Grohe, M.: Weisfeiler and leman go neural: Higher-order graph neural networks (2020)
28. Nishad, S., Agarwal, S., Bhattacharya, A., Ranu, S.: Graphreach: Position-aware graph neural network using reachability estimations (2021)
29. Oono, K., Suzuki, T.: Graph neural networks exponentially lose expressive power for node classification. arXiv preprint arXiv:1905.10947 (2019)
30. Ren, P., Xiao, Y., Chang, X., Huang, P.Y., Li, Z., Chen, X., Wang, X.: A survey of deep active learning (2020)
31. Ren, P., Xiao, Y., Chang, X., Huang, P.Y., Li, Z., Gupta, B.B., Chen, X., Wang, X.: A survey of deep active learning. ACM computing surveys (CSUR) **54**(9), 1–40 (2021)
32. Sato, R.: A survey on the expressive power of graph neural networks. arXiv preprint arXiv:2003.04078 (2020)
33. Scarselli, F., Tsoi, A.C., Hagenbuchner, M.: The vapnik–chervonenkis dimension of graph and recursive neural networks. Neural Networks **108**, 248–259 (2018)
34. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T.: Collective classification in network data. AI magazine **29**(3), 93–93 (2008)
35. Sheng, G., Su, J., Huang, C., Wu, C.: Mspipe: Efficient temporal gnn training via staleness-aware pipeline. arXiv preprint arXiv:2402.15113 (2024)
36. Sinha, A., Shen, Z., Song, Y., Ma, H., Eide, D., Hsu, B.J., Wang, K.: An overview of microsoft academic service (mas) and applications. In: Proceedings of the 24th international conference on world wide web. pp. 243–246 (2015)
37. Su, J., Marbach, P.: Structure of core-periphery communities. In: International Conference on Complex Networks and Their Applications. pp. 151–161. Springer (2022)
38. Su, J., Marbach, P.: Structural properties of coreâ€“periphery communities. Advances in Complex Systems (ACS) **26**(06), 1–43 (2023)

39. Su, J., Zou, D., Wu, C.: Pres: Toward scalable memory-based dynamic graph neural networks (2024), https://arxiv.org/abs/2402.04284
40. Su, J., Zou, D., Zhang, Z., Wu, C.: Towards robust graph incremental learning on evolving graphs. In: International Conference on Machine Learning. pp. 32728–32748. PMLR (2023)
41. Tsubaki, M., Tomii, K., Sese, J.: Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. Bioinformatics **35**(2), 309–318 (2019)
42. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
43. Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., et al.: Deep graph library: A graph-centric, highly-performant package for graph neural networks. arXiv preprint arXiv:1909.01315 (2019)
44. Wu, Q., Zhang, H., Yan, J., Wipf, D.: Handling distribution shifts on graphs: An invariance perspective (2022)
45. Wu, Y., Xu, Y., Singh, A., Yang, Y., Dubrawski, A.: Active learning for graph neural networks via node feature propagation. arXiv preprint arXiv:1910.07567 (2019)
46. Xu, D., Zhu, Y., Choy, C.B., Fei-Fei, L.: Scene graph generation by iterative message passing (2017)
47. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? arXiv preprint arXiv:1810.00826 (2018)
48. Yang, C., Wu, Q., Wang, J., Yan, J.: Graph neural networks are inherently good generalizers: Insights by bridging gnns and mlps. arXiv preprint arXiv:2212.09034 (2022)
49. Yang, J., Ang, Y.Z., Guo, Z., Zhou, K., Zhang, W., Liu, Z.: Panoptic scene graph generation (2022)
50. Yang, Z., Cohen, W.W., Salakhutdinov, R.: Revisiting semi-supervised learning with graph embeddings (2016). https://doi.org/10.48550/ARXIV.1603.08861, https://arxiv.org/abs/1603.08861
51. You, J., Ying, R., Leskovec, J.: Position-aware graph neural networks (2019)
52. Zellers, R., Yatskar, M., Thomson, S., Choi, Y.: Neural motifs: Scene graph parsing with global context (2018)
53. Zhou, K., Huang, X., Li, Y., Zha, D., Chen, R., Hu, X.: Towards deeper graph neural networks with differentiable group normalization (2020)
54. Zhu, G., Zhang, L., Jiang, Y., Dang, Y., Hou, H., Shen, P., Feng, M., Zhao, X., Miao, Q., Shah, S.A.A., Bennamoun, M.: Scene graph generation: A comprehensive survey (2022)