OOD Link Prediction Generalization Capabilities of Message-Passing GNNs in Larger Test Graphs

Yangze Zhou
Department of Statistics
Purdue University
West Lafayette, IN 47903
zhou950@purdue.edu

Gitta Kutyniok
Department of Mathematics
Ludwig-Maximilians-Universitat München
Munich, Germany
kutyniok@math.lmu.de

Bruno Ribeiro
Department of Computer Science
Purdue University
West Lafayette, IN 47903
ribeiro@cs.purdue.edu

Abstract

This work provides the first theoretical study on the ability of graph Message Passing Neural Networks (gMPNNs) —such as Graph Neural Networks (GNNs)— to perform inductive out-of-distribution (OOD) link prediction tasks, where deployment (test) graph sizes are larger than training graphs. We first prove non-asymptotic bounds showing that link predictors based on permutation-equivariant (structural) node embeddings obtained by gMPNNs can converge to a random guess as test graphs get larger. We then propose a theoretically-sound gMPNN that outputs structural pairwise (2-node) embeddings and prove non-asymptotic bounds showing that, as test graphs grow, these embeddings converge to embeddings of a continuous function that retains its ability to predict links OOD. Empirical results on random graphs show agreement with our theoretical results.

1 Introduction

Link prediction is the task of predicting whether two nodes likely have a missing link [1, 12, 30, 37, 66]. Link prediction tasks arise in many settings, ranging from predicting edges on bipartite graphs between users and products or content in recommender systems [6, 11, 31, 32, 39, 62], to knowledge graph reconstruction [4, 14, 20, 54, 66, 67], to predicting protein-protein interactions [57].

In recent years, there has been growing interest in applying neural network models to inductive link prediction tasks. Inductive link prediction considers methods trained on a graph G^{tr} and deployed at test time on another graph G^{te}. It also encompasses the task of training the method on a smaller induced subgraph G^{tr} of a larger graph G^{te}, then deploying it on the entire graph. In particular, our work focuses on graph message-passing Neural Networks (gMPNNs) [21, 60] or, more precisely, the widely used Graph Neural Network (GNN) framework [8, 9, 13, 22, 24, 28, 61, 64, 69].

Our work asks the following questions: *Are link prediction methods able to cope with the task of inductive out-of-distribution (OOD) link prediction, where (unseen) test graphs are significantly larger than training graphs?* How can these OOD link prediction tasks be theoretically defined? Can we obtain non-asymptotic bounds on the generalization capabilities of these methods?

The majority of today's link prediction methods are based on a similar principle. Consider an attributed graph $G = (V, E)$, with node set $V = \{1, ..., N\}$, edge set $E \subseteq V \times V$, and node features

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

$\boldsymbol{F} \in \mathbb{R}^{N \times F_0}$, $F_0 \geq 1$. Then, given a pair of nodes $i, j \in V$, after $T \geq 1$ iterations over G, these methods produce associated node embeddings (representation vectors) $\Theta^{\bullet}_i, \Theta^{\bullet}_j \in \mathbb{R}^{F_T}$, $F_T \geq 1$, which are then used in a link function $\eta^{\bullet} : \mathbb{R}^{F_T} \times \mathbb{R}^{F_T} \to [0, 1]$ such that $\eta^{\bullet}(\Theta^{\bullet}_i, \Theta^{\bullet}_j)$ predicts the probability that i and j have a missing link in G. In our notation we will denote all node embeddings and associated functions with the superscript "$^{\bullet}$". Henceforth we denote gMPNNs that output structural node embeddings as gMPNNs$^{\bullet}$.

Node embeddings. The first part of our work considers a subset of these methods, where the output node embeddings are permutation equivariant (a.k.a. *structural node embeddings* [65]). Informally, a sequence of node embeddings $\Theta^{\bullet} \in \mathbb{R}^{N \times F_T}$ given by an embedding method is permutation-equivariant if for any arbitrary graph G and any permutation $\pi \in \mathbb{S}_N$ of the node indices, where $\mathbb{S}_N$ is the symmetric group, the resulting isomorphic graph $G' = (\pi \circ V, \pi \circ E, \pi \circ \boldsymbol{F})$ gets permuted node embeddings $\Theta^{\bullet\prime} = \pi \circ \Theta^{\bullet}$, where $\pi \circ M$ defines the action of π on M (we will provide a formal definition in Section 2). We leave the study of OOD link prediction with *positional node embeddings* (a.k.a. permutation-sensitive node embeddings [65]) to future work.

The application of GNNs to link prediction tasks is made difficult by the fact that, by construction, permutation-equivariant GNNs give the same embeddings $\Theta^{\bullet}_i, \Theta^{\bullet}_j$ to any isomorphic nodes i, j in G, as noted by You et al. [82] and Srinivasan and Ribeiro [65]. Isomorphic nodes are nodes that are structurally indistinguishable in G (even when considering node features) except by their (assumed arbitrary) node indices $i, j \in V$. That is, if a graph has isomorphic pairs, permutation-equivariant GNN link prediction can fail. The recent link prediction literature has significantly relied on isomorphic nodes for theoretical results (e.g., Zhang et al. [86, Theorem 2] uses isomorphic nodes to prove that, uniformly, graphs are likely to have many isomorphic nodes and hence are not amenable to accurate link prediction). However, isomorphic nodes are rare in both real-world graphs (see Figure 3 in the Appendix) and in large random graphs (Proposition 1).

An important open question is whether equivariant GNN would be able to predict links in asymmetric graphs. That is, the concerns of [65, 82] may not be of practical importance. Our work also answers this question: We see that for in-distribution link prediction tasks (where graph test sizes are the same as training sizes), permutation-equivariant GNNs are able to predict links by tapping into the graph asymmetries. However, we show theoretically and empirically that tapping into asymmetries can fail OOD even when it works in-distribution.

Pairwise embeddings. Taking a different route, Srinivasan and Ribeiro [65] provides an existence proof that the link prediction task between i and j can always be performed by a pairwise embedding $\Theta^{\bullet\bullet}_{ij}(G)$, i.e., for any pair of nodes i, j in a graph G, there exists a pairwise embedding $\Theta^{\bullet\bullet}_{ij}(G)$ and a link function $\eta^{\bullet\bullet} : \mathbb{R}^{F_T} \to [0, 1]$ such that $\eta^{\bullet\bullet}(\Theta^{\bullet\bullet}_{ij})$ approximates the probability that i and j have a hidden link. In our notation we will denote all pairwise (joint 2-node) embeddings and associated functions with the superscript "$^{\bullet\bullet}$". Unfortunately, as the test graph grows, we were unable to prove existing pairwise embedding methods [50, 72, 84, 86, 87] are able to perform OOD link prediction tasks. Hence, we propose a novel family of gMPNNs for pairwise embeddings, denoted gMPNNs$^{\bullet\bullet}$ henceforth. The second part of our work considers the OOD generalization capability of these gMPNNs$^{\bullet\bullet}$.

Contributions. In this work we study inductive OOD link prediction tasks for larger test graphs using permutation-equivariant node and pairwise embeddings, $\Theta^{\bullet}$ and $\Theta^{\bullet\bullet}$, respectively. Our work makes the following contributions:

1. We provide a theoretical framework defining OOD inductive link prediction tasks, where test graphs are significantly larger than training graphs.
2. We show that structural node embeddings from message-passing GNNs can fail in OOD link prediction tasks if the test graph (from the same graph family) is significantly larger than the training graph. Our work fills *an important gap in the literature*, where Bevilacqua et al. [7] studied the OOD capabilities of GNNs for *graph classification* using random graph models. Our work studies the OOD capabilities of GNNs for *inductive link prediction* in a similar setting.
3. We propose a new family of structural pairwise embeddings, denoted gMPNNs$^{\bullet\bullet}$, that can provably perform the above OOD task.
4. We provide non-asymptotic bounds on the convergence of pairwise gMPNNs embeddings. Extensive empirical experiments using stochastic block models (SBMs [63]) validate our theoretical

results. Our work focuses on providing a theoretical understanding of the challenges of OOD link prediction tasks rather than propose real-world link prediction tasks and compare baselines. However, we believe that our work lays the theoretical foundation (and challenges) for future application-focused works.

2 Preliminaries

Given an attributed graph $G = (V, E)$, with node set $V = \{1, ..., N\}$, edge set $E \subseteq V \times V$, adjacency matrix $\boldsymbol{A} \in \{0, 1\}^{N \times N}$, where $\boldsymbol{A}_{ij} = \mathbb{1}_{\{(i,j) \in E\}}$, and node features $\boldsymbol{F} \in \mathbb{R}^{N \times F_0}$, $F_0 > 0$. Let $\boldsymbol{P}_\pi \in \mathcal{B}_N$ be a permutation matrix associated with permutation $\pi \in \mathbb{S}_N$ (where $\mathbb{S}_N$ is the symmetric group), where $\mathcal{B}_N$ denotes the Birkhoff polytope of $N \times N$ doubly-stochastic matrices. Doubly-sctochastic matrices are non-negative square matrices whose rows and columns sum to one. The matrix $\boldsymbol{P}_\pi$ defines the action of permutation π on these matrices, e.g., $\pi \circ \boldsymbol{A} = \boldsymbol{P}_\pi \boldsymbol{A} \boldsymbol{P}_\pi^T$. We denote a pair of nodes $i, j \in V$ as isomorphic in G if exists $\pi \in \mathbb{S}_N$ such that $\pi_i = j$, $\boldsymbol{A} = \boldsymbol{P}_\pi \boldsymbol{A} \boldsymbol{P}_\pi^T$, and $\boldsymbol{F} = \boldsymbol{P}_\pi \boldsymbol{F}$. Node features can be defined by the *graph signal* $f : V \to \mathbb{R}^{F_0}$ as a function that maps a node to an F_0-dimensional feature in $\mathbb{R}^{F_0}$. Then the signal of the graph $\boldsymbol{F}$ can be represented by a matrix $\boldsymbol{F} = [\mathbf{f}_1, ..., \mathbf{f}_N]^T \in \mathbb{R}^{N \times F_0}$, where $\mathbf{f}_i \in \mathbb{R}^{F_0}$ are the features of node $i \in V$.

Random graph model for G**.** Denote the metric-measure space by $(\mathcal{X}, d, \mu)$, where $\mathcal{X}$ is a set, d is a metric, and μ is a probability Borel measure. A *graphon* is defined as a mapping $W : \mathcal{X} \times \mathcal{X} \to [0, 1]$ [15, 75]. In what follows we define how the graph G is sampled from the graphon models. The signal definition follows Maskey et al. [46, Definition 2.3] and the edge samples follow Airoldi et al. [3], Lawrence and Hyvärinen [34].

Definition 1 (Random graph model)**.** *We define* (W, f) *as a* random graph model *for* G *on* $(\mathcal{X}, d, \mu)$ *with the graphon* $W : \mathcal{X} \times \mathcal{X} \to [0, 1]$ *and the metric-space signal* $f : \mathcal{X} \to \mathbb{R}^{F_0}$*.* $f \in L^\infty(\mathcal{X})$ *is an essentially bounded measurable function with the essential supremum norm. We obtain* $(G, \boldsymbol{F})$ *by first sampling* N *i.i.d. random points* $X_1, ..., X_N$ *from* $\mathcal{X}$ *with probability density* μ*, as the nodes of G. Then the edge* (i, j) *between nodes i and j is sampled with probability* $W(X_i, X_j)$*, i.e., the adjacency matrix* $\boldsymbol{A} = (\boldsymbol{A}_{i,j})_{i,j}$ *of G is defined as* $\boldsymbol{A}_{i,j} = \mathbb{1}(Z_{i,j} < W(X_i, W_j))$ *for* $i, j = 1, ..., N$*, where* $\{Z_{i,j}\}_{i,j=1}^N$ *are sampled i.i.d. from Uniform*$(0, 1)$*. The graph signal* $\boldsymbol{F} = [\mathbf{f}_1, ..., \mathbf{f}_N]^T \in \mathbb{R}^{N \times F_0}$ *is defined as* $\mathbf{f}_i = f(X_i)$*. We say* $(G, \boldsymbol{F}) \sim (W, f)$*. Further, we restrict our attention to graphons* W *such that there exists a constant* d_{min} *satisfying the* graphon degree $d_W(x) := \int_{\mathcal{X}} W(x, y) d\mu(y) \geq d_{min} > 0, \forall x \in \mathcal{X}$.

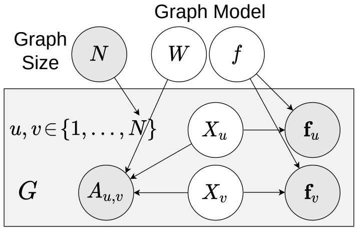

Figure 1: Templated causal DAG of G. Hidden and observed variables are shaded white and gray, respectively.

In an abuse of notation we identify node $i \in V$ with the sampled value $X_i \sim \mu, \forall i \in \{1, ..., N\}$, since generally μ is such that $P(X_i = X_j) = 0$ almost everywhere for $i \neq j$ (e.g., μ is uniform). The causal DAG of the data generation process of G is given in Figure 1. Our goal is to produce predictors that survive the distribution shift implied by a change in the distribution of graph sizes N during test time. Furthermore, we note that all proofs are relegated to the Appendix due to space constraints.

2.1 Inductive structural node representations with graph message-passing neural networks

Henceforth use the terms *node embeddings* and *node representations* interchangeably. Graph message-passing Neural Network (gMPNN$^\bullet$) is defined by realizing a message-passing Neural Network (MPNN) on a graph. We now restate the Maskey et al. [46, Definition 2.1] of MPNN.

Definition 2 (MPNN [46])**.** *Let* $T \in \mathbb{N}$ *denote the number of layers. For* $t = 1, ..., T$*, let* $\Phi^{(t)} : \mathbb{R}^{2F_{t-1}} \to \mathbb{R}^{H_{t-1}}$ *and* $\Psi^{(t)} : \mathbb{R}^{F_{t-1}+H_{t-1}} \to \mathbb{R}^{F_t}$ *be functions, where* $F_t \in \mathbb{N}$ *is called the feature dimension of layer t. The corresponding MPNN* Θ *is define by the sequence of message functions* $(\Phi^{(t)})_{t=1}^T$ *and update functions* $(\Psi^{(t)})_{t=1}^T$*, i.e.* $\Theta = ((\Phi^{(t)})_{t=1}^T, (\Psi^{(t)})_{t=1}^T)$*.*

We now introduce the gMPNN$^\bullet$ with T message-passing layers. For each node $i \in V$, $\mathbf{f}_i^{\bullet(t)}$ at layer $t \in \{1, \ldots, T\}$ is defined recursively using (a) its own representation at layer $t - 1$ ($\mathbf{f}_i^{\bullet(t-1)}$) and (b) an aggregated representation of its neighbors $m_i^{(t)}$. Unlike Maskey et al. [46, Definition 2.2] considering *mean aggregation, we consider here the (N-normalized)sum representation* as follows:

Definition 3 (gMPNN$^\bullet$)**.** *(Adaptation of [46, Definition 2.2] to N-normalized GNNs) Let $(G, \boldsymbol{F})$ be a graph with graph signals as in Definition 1 and Θ be a MPNN as in Definition 2. For layer $t = 1, ..., T$, define $\Theta_{\boldsymbol{A}}^{\bullet(t)}$ as maps from the input graph G and graph signals $\boldsymbol{F}^{(0)} = \boldsymbol{F} \in \mathbb{R}^{N \times F_0}$ to the features in the t-th neural layer by*

$$\Theta_{\boldsymbol{A}}^{\bullet(t)} : \mathbb{R}^{N \times F_0} \to \mathbb{R}^{N \times F_t}, \quad \boldsymbol{F} \mapsto \boldsymbol{F}^{(t)} = (\mathbf{f}_i^{\bullet(t)})_{i=1}^N$$

where $\boldsymbol{F}^{(t)}$ is defined by the (N-normalized) sum aggregation procedure, $\forall i \in V$, for $\Theta_{\boldsymbol{A}}^{\bullet(t)}$,

$$m_i^{(t)} := \frac{1}{N} \sum_{j=1}^{N} A_{i,j} \Phi^{(t)}(\mathbf{f}_i^{\bullet(t-1)}, \mathbf{f}_j^{\bullet(t-1)}),$$

and

$$\mathbf{f}_i^{\bullet(t)} := \Psi^{(t)}(\mathbf{f}_i^{\bullet(t-1)}, m_i^{(t)}).$$

Given a gMPNN$^\bullet$, $\Theta_{\boldsymbol{A}}^{\bullet(T)}$, with $T \geq 1$ layers as in Definition 3, their outputs are $\Theta_{\boldsymbol{A}}^{\bullet(T)}(\boldsymbol{F}) \in \mathbb{R}^{N \times F_T}$ for the (N-normalized) sum aggregation, and are henceforth denoted as node embedding outputs of the gMPNN$^\bullet$. We denote $\Theta_{\boldsymbol{A}}^{\bullet(T)}(\boldsymbol{F})_i$ as the node embedding for node $i \in V$.

2.2 Node embeddings with *continuous* message passing neural networks

Here we adapt the degree-normalized definition of Maskey et al. [46] on continuous message passing neural networks for structural node embeddings to our continuous integral aggregation (N-normalized GNNs).

Definition 4 (Continuous message-passing)**.** *Given a MPNN Θ as in Definition 2, the* node continuous message passing neural network (cMPNN$^\bullet$) *on graphons and metric-space signals $f : \mathcal{X} \to \mathbb{R}^{F_0}$ can be defined by replacing the graph node features and the aggregation scheme in Definition 3 by the following continuous counterparts. Using a message signal $U : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^H$, the continuous integral aggregation is defined as $M_W^\bullet(U)(x) = \int_{\mathcal{X}} W(x, y)U(x, y)d\mu(y)$, where W is a graphon.*

As defined in Maskey et al. [46, Definition 2.4], the same MPNN Θ can process metric-space signals instead of graph signals with the continuous aggregations. Instead of using continuous mean aggregation as [46, Definition 2.4], we are using continuous integral aggregation.

Definition 5 (cMPNN$^\bullet$)**.** *(Adaptation of [46, Definition 2.4] to N-normalized GNNs) Let (W, f) be a random graph model as in Definition 1 and Θ be a MPNN as in Definition 2. For $t = 1, ..., T$, define $\Theta_W^{\bullet(t)}$ as maps from input metric-space signal $f^{\bullet(0)} = f : \mathcal{X} \to \mathbb{R}^{F_0}$ to the features in the t-th layer by*

$$\Theta_W^{\bullet(t)} : L^2(\mathcal{X}) \to L^2(\mathcal{X}), \quad f^\bullet \mapsto f^{\bullet(t)},$$

where $\overline{f}^{\bullet(t)}$ are defined sequentially through the integral aggregation for

$$\begin{aligned} \Theta_W^{\bullet(t)} : g^{\bullet(t)}(x) : &= M_W^\bullet(\Phi^{(t)}(f^{\bullet(t-1)}, f^{\bullet(t-1)}))(x) \\ &= \int_{\mathcal{X}} W(x, y)\Phi^{(t)}(f^{\bullet(t-1)}(x), f^{\bullet(t-1)}(y))d\mu(y), \\ f^{\bullet(t)}(x) : &= \Psi^{(t)}(f^{\bullet(t-1)}(x), g^{\bullet(t)}(x)). \end{aligned}$$

3 Size-stability of node representation and its drawbacks

We now present our results about convergence of gMPNN$^\bullet$ to cMPNN$^\bullet$ for test graphs G^{te} sampled from the graphon random graph model (see Definition 1), and how it leads to size-stability of gMPNN$^\bullet$ for nodes that have the same representation under cMPNNs$^\bullet$. In what follows we will focus on the neighbor-average aggregation procedure (a) of Definition 3, since this is the more difficult case to prove. *Similar results for the (N-normalized) sum aggregation procedure (Definition 3(b)) are shown in the Appendix due to space constraints.* Moreover, common definitions (e.g., Lipschitz continuous functions) are also relegated to the Appendix to save space.

3.1 Convergence of gMPNNs towards cMPNNs as test graph size increase

We now prove that, with high probability, the maximum infinity difference between the gMPNN$^\bullet$ and cMPNN$^\bullet$ node representations decreases with N^{te}, the size of G^{te}. The proof of Theorem 1 closely follows the pointwise convergence proof in Maskey et al. [45], adapted to our OOD setting and can be found in the Appendix.

Theorem 1 (OOD convergence without in-distribution convergence)**.** *For a random graph model (W, f) satisfying Definition 1, let N^{tr} be a random variable defining the distribution of graph sizes in training. Define the test distribution $(G^{te}, \boldsymbol{F}^{te}) \sim (W, f)$ through the causal graph in Figure 1 as an interventional change to obtain larger test graph sizes where $\min(supp(N^{te})) \gg M_{tr} = \max(supp(N^{tr}))$ (which means any test graph is much larger than the largest possible training graph). Let $\Theta = ((\Phi^{(l)})_{l=1}^{T}, (\Psi^{(l)})_{l=1}^{T})$ be a MPNN as in Definition 2 with T layers such that $\Phi^{(l)} : \mathbb{R}^{2F_{l-1}} \to \mathbb{R}^{H_{l-1}}$ and $\Psi^{(l)} : \mathbb{R}^{F_{l-1}+H_{l-1}} \to \mathbb{R}^{F_l}$ are learned from the training distribution and are Lipschitz continuous with Lipschitz constants $L_{\Phi}^{(l)}(M_{tr})$ and $L_{\Psi}^{(l)}(M_{tr})$ that depend on M_{tr}. Let gMPNN$^\bullet$ $\Theta_A^{\bullet(T)}$ and cMPNN$^\bullet$ $\Theta_W^{\bullet(T)}$ be as in Definitions 3 and 5. Let $X_1^{te}, ..., X_{N^{te}}^{te}$ and $\boldsymbol{A}^{te}$ be as in Definition 1. Let $p \in (0, \frac{1}{\sum_{l=1}^{T} 2(H_l+1)})$. Then, if*

$$\frac{\sqrt{N^{te}}}{\sqrt{\log\left(2N^{te}/p\right)}} \geq \frac{4\sqrt{2}}{d_{min}}, \tag{1}$$

we have with probability at least $1 - \sum_{l=1}^{T} 2(H_l + 1)p$,

$$\delta^{\bullet}_{A\text{-}W} := \max_{i=1,\dots,N^{te}} \|\Theta^{\bullet(T)}_{\boldsymbol{A}^{te}}(\boldsymbol{F}^{te})_i - \Theta^{\bullet(T)}_{W}(f)(X_i^{te})\|_\infty \leq (C_1 + C_2\|f\|_\infty)\frac{\sqrt{\log(2N^{te}/p)}}{\sqrt{N^{te}}},$$

where the constants C_1 and C_2 are defined in the Appendix and depend on $\{L_{\Phi}^{(l)}(M_{tr}), L_{\Psi}^{(l)}(M_{tr})\}_{l=1}^{T}$ and the distribution of $(G^{tr}, \boldsymbol{F}^{tr})$.

Theorem 1 above shows that as the test graph size N^{te} grows, the node representations from the discrete gMPNNs$^\bullet$learned in the training data converge to the continuous cMPNNs$^\bullet$. *Theorem 1's OOD statement has profound consequences when it comes to predicting links using the node representations obtained by a gMPNN$^\bullet$.* Next, Corollary 1 shows that for any two nodes $i, j \in V^{\text{te}}$ that are indistinguishable in the cMPNN$^\bullet$ (defined as $\Theta_W^{\bullet(T)}(f)(X_i^{\text{te}}) = \Theta_W^{\bullet(T)}(f)(X_j^{\text{te}})$), they will get increasingly similar representations in the discrete gMPNN$^\bullet$ as N^{te} grows.

Corollary 1. *Let $\Theta = ((\Phi^{(l)})_{l=1}^{T}, (\Psi^{(l)})_{l=1}^{T}), \Theta_A^{\bullet(T)}, \Theta_W^{\bullet(T)}, p, (W, f), (G^{tr}, \boldsymbol{F}^{tr}), (G^{te}, \boldsymbol{F}^{te}), N^{tr}, N^{te}, A^{te}$, and $X_1^{te}, ..., X_{N^{te}}^{te}$ be as in Theorem 1. If there exists $i, j \in V^{te}, i \neq j$, s.t. $\Theta_W^{\bullet(T)}(X_i) = \Theta_W^{\bullet(T)}(X_j)$ and Equation (1) is satisfied, then, with C_1 and C_2 as in Theorem 1, we have that with probability at least $1 - \sum_{l=1}^{T} 2(H_l + 1)p$,*

$$\|\Theta^{\bullet(T)}_{\boldsymbol{A}^{te}}(\boldsymbol{F}^{te})_i - \Theta^{\bullet(T)}_{\boldsymbol{A}^{te}}(\boldsymbol{F}^{te})_j\|_\infty \leq (C_1 + C_2\|f\|_\infty)\frac{2\sqrt{\log(2N^{te}/p)}}{\sqrt{N^{te}}}.$$

Implications of Corollary 1 on Stochastic Block Models (SBMs). In what follows, we will discuss circumstances where two nodes $i, j \in V$ get the same cMPNN$^\bullet$ representations (i.e., both $\Theta_W^{\bullet(T)}(f)(X_i) = \Theta_W^{\bullet(T)}(f)(X_j)$. In what follows we restrict our results to an important family of graphon models: Stochastic Block Models (SBMs) [63], where we also model node attributes. SBMs were chosen because they can consistently model large graphs generated by any piecewise Lipschitz graphon model [3]. SBMs are also intuitive models, which makes them useful to illustrate our results.

Definition 6 (Stochastic Block Model (SBM))**.** *An SBM (W, f) is a random graph model (Definition 1) with cluster structures in W and f. Partition the node set into $r \geq 2$ disjoint subsets $S_1, S_2, ..., S_r \subseteq V$ (known as blocks or communities) with an associated $r \times r$ symmetric matrix $\boldsymbol{S}$, where the probability of an edge (i, j), $i \in S_a$ and $j \in S_b$ is $\boldsymbol{S}_{ab}$, for $a, b \in \{1, \dots, r\}$. Let $\mathcal{X} = [0, 1]$, and μ be the uniform distribution on $[0, 1]$. By dividing $\mathcal{X} = [0, 1]$ into disjoint convex sets $[t_0, t_1), [t_1, t_2), \dots, [t_{r-1}, t_r]$, where $t_0 = 0$ and $t_r = 1$, node i belongs to block S_a if $X_i \sim$ Uniform$(0, 1)$ satisfies $X_i \in [t_{a-1}, t_a)$. The graphon function W is defined as $W(X_i, X_j) = \sum_{a,b \in \{1,\dots,r\}} \boldsymbol{S}_{ab}\mathbb{1}(X_i \in [t_{a-1}, t_a))\mathbb{1}(X_j \in [t_{b-1}, t_b))$. We take the liberty to also define node signals in our SBM model, where for $\boldsymbol{B} = [B_1, ..., B_r]^T \in \mathbb{R}^{r \times F_0}$ the metric-space signal $f : \mathcal{X} \to \mathbb{R}^{F_0}$ is defined as $f(x) = \sum_{a \in \{1,\dots,r\}} \mathbb{1}(x \in [t_{a-1}, t_a))B_a$.*

We define the action of permutation π on $\boldsymbol{B}$ of Definition 6 as $\pi \circ \boldsymbol{B}$, where $(\pi \circ \boldsymbol{B})_{\pi_a} = \boldsymbol{B}_a$.

Definition 7 (Isomorphic SBM blocks)**.** *For the SBM model (W, f) in Definition 6, we say two blocks $a, b \in \{1, \dots, r\}$ are isomorphic if the SBM satisfies the following two conditions: (a) $t_a - t_{a-1} = t_b - t_{b-1}$, and (b) for $\pi \in \mathbb{S}_r$, such that $\pi_a = b$, $\pi_b = a$ and $\pi_c = c, \forall c \in \{1, ..., r\},\ c \neq a, b$, $\boldsymbol{S} = \pi \circ \boldsymbol{S}$, and $\boldsymbol{B} = \pi \circ \boldsymbol{B}$.*

A similar definition can be obtained for the general graphons in Definition 1 using the isomorphic graphon definition of Lovász and Szegedy [40].

Now that we have the definition for isomorphic blocks in SBM models, we can prove that all nodes in these isomorphic blocks will obtain the same representations under integral aggregation cMPNNs$^\bullet$.

Lemma 1. *Let $\Theta = ((\Phi^{(l)})_{l=1}^T, (\Psi^{(l)})_{l=1}^T)$ be a MPNN as in Definition 2, and $\Theta_W^{\bullet(T)}$ as in Definition 5. For the SBM model (W, f) in Definition 6 with N^{te} nodes $X_1, \dots, X_{N^{te}}$. If there exists $i, j \in V^{te}$ such that X_i^{te}, X_j^{te} are nodes that belong to isomorphic SBM blocks (Definition 7), then $\Theta_W^{\bullet(T)}(f)(X_i^{te}) = \Theta_W^{\bullet(T)}(f)(X_j^{te})$.*

Note that even though any two nodes in isomorphic SBM blocks get the same cMPNN$^\bullet$ representations per Lemma 1, these nodes are likely not isomorphic in G^{te} (as shown in Proposition 1 in Appendix) and, hence, they get different gMPNN$^\bullet$ representations. However, Corollary 1 shows that these representations become increasingly similar as the test graph size grows ($N^{\text{te}} \gg 1$). We use this observation to understand the ability of gMPNNs$^\bullet$ to perform link prediction tasks next.

3.2 The hardness of OOD inductive link prediction using structural node embeddings

The convergence of gMPNNs$^\bullet$ to cMPNNs$^\bullet$ as the test graph size N^{te} grows (Theorem 1) implies through Corollary 1 and Lemma 1 that node representations of distinct SBM blocks can become increasingly similar as the test graph size grows ($N^{\text{te}} \gg 1$), even though these nodes are not isomorphic in G^{te} with high probability (see Proposition 1 in the Appendix).

Definition 8 (Link prediction function from structural node embeddings)**.** *An inductive link prediction function $\eta^\bullet : \mathbb{R}^{F_T} \times \mathbb{R}^{F_T} \to [0, 1]$ takes the gMPNN$^\bullet$ node representations of two nodes $i, j \in V^{te}$ and predicts the edge probability $P(\boldsymbol{A}_{ij}^{te} = 1)$. We assume $\eta^\bullet$ is Lipschitz continuous with Lipschitz constant $L_{\eta^\bullet}(M_{tr})$ that depends on $\max(supp(N^{tr}))$. In the context of graphon random graph models (Definition 1), we aim to learn $\eta^\bullet(\Theta_{\boldsymbol{A}^{te}}^{\bullet(T)}(\boldsymbol{F}^{te})_i, \Theta_{\boldsymbol{A}^{te}}^{\bullet(T)}(\boldsymbol{F}^{te})_j) \approx W(i, j)$. We further assume we predict a link if $\eta^\bullet(\cdot, \cdot) > \tau$, while no link if $\eta^\bullet(\cdot, \cdot) < \tau$, for some (arbitrary) threshold $\tau \in [0, 1]$ chosen by the user of such system.*

The next corollary showcases the difficulty in OOD predicting links using structural node representations as N^{te} grows.

Corollary 2. *Let $\Theta = ((\Phi^{(l)})_{l=1}^T, (\Psi^{(l)})_{l=1}^T)$ be the MPNN with T layers and $\Theta_A^{\bullet(T)}, \Theta_W^{\bullet(T)}$ as in Theorem 1. Let $\eta^\bullet : \mathbb{R}^{F_T} \times \mathbb{R}^{F_T} \to [0, 1]$ be as in Definition 8. Consider the SBM (W, f) in Definition 6 with isomorphic blocks (Definition 7). Let $(G^{tr}, \boldsymbol{F}^{tr}) \sim (W, f)$ and $(G^{te}, \boldsymbol{F}^{te}) \sim (W, f)$ be the training and test graphs with N^{tr} and N^{te} nodes, respectively. Consider any two test nodes $i, j \in \{1, ..., N^{te}\}$, $i \neq j$, for which we can make a link prediction decision with $\eta^\bullet$ (i.e., $\eta^\bullet(\Theta_{\boldsymbol{A}^{te}}^{\bullet(T)}(\boldsymbol{F}^{te})_i, \Theta_{\boldsymbol{A}}^{\bullet(T)}(\boldsymbol{F}^{te})_j) \neq \tau$). Let G^{te} be large enough to satisfy both Equation* (1) *and*

$$\frac{\sqrt{N^{te}}}{\sqrt{\log(2N^{te}/p)}} > \frac{2(C_1 + C_2\|f\|_\infty)}{|\eta^\bullet(\Theta_{\boldsymbol{A}^{te}}^{\bullet(T)}(\boldsymbol{F}^{te})_i, \Theta_{\boldsymbol{A}^{te}}^{\bullet(T)}(\boldsymbol{F}^{te})_j) - \tau| / L_\eta^\bullet(M_{tr})},$$

where p, C_1, and C_2 are as given in Corollary 1. Then, if i and j belong to isomorphic blocks (i.e., $\Theta_W^{\bullet(T)}(f)(X_i^{te}) = \Theta_W^{\bullet(T)}(f)(X_j^{te})$), with probability at least $1 - \sum_{l=1}^T 2(H_l + 1)p$ the link prediction method in Definition 8 will make the same link prediction regardless of the SBM probability matrix $\boldsymbol{S}$ (Definition 6) and whether i and j are in the same block or distinct isomorphic blocks.

Corollary 2 proves that link prediction with structural node embeddings form gMPNNs$^\bullet$ is unreliable. That is, for any link prediction method satisfying Definition 8, as the test graph grows $N^{\text{te}} \gg 1$, the method will increasingly struggle to give different predictions within and across isomorphic SBM blocks, even when these probabilities are arbitrarily different in the underlying graph model. In what follows we show that pairwise embeddings can address this challenge.

4 Size-stability of structural *pairwise* embeddings and its advantages

We have discussed the limitation of gMPNNs$^\bullet$ on node representation for link prediction. Now we claim that a joint continuous message passing graph neural network is capable of link prediction in graphon random graph models (Definition 1). We define the joint continuous message passing graph neural network inspired by the cMPNNs$^\bullet$ for node representations (Definition 5). First, we need to define the *graphon fraction of common neighbors* for graphon nodes x and y, $c_W(x, y) := \int_{\mathcal{X}} W(x, z)W(y, z)d\mu(z)$. We only consider graphons W such that there exists d_{cmin} satisfying $c_W(x, y) \geq d_{\text{cmin}} > 0, \forall x, y \in \mathcal{X}$ in this section. Since we do not have edge feature as in Definition 1, we define the metric-space pair-wise signal as $f^{\bullet\bullet}(x, y) = 1, \forall x, y \in \mathcal{X}$.

Definition 9 (cMPNN$^{\bullet\bullet}$)**.** *Let (W, f) be a random graph model as in Definition 1 and Θ be a MPNN as in Definition 2. For $t = 1, ..., T$, define the continuous (pairwise) cMPNN$^{\bullet\bullet}$ $\Theta_W^{\bullet\bullet(t)}$ as the mapping that maps input pairwise metric-space signals $f^{\bullet\bullet(0)} = f^{\bullet\bullet}$ to the features in the t-th layer by*

$$\Theta_W^{\bullet\bullet(t)} : L^2(\mathcal{X}, \mathcal{X}) \to L^2(\mathcal{X}, \mathcal{X}), \quad f^{\bullet\bullet(0)} \mapsto f^{\bullet\bullet(t)},$$

where $f^{\bullet\bullet(t)}$ are defined recursively by

$$\begin{aligned} g^{\bullet\bullet(t)}(x, y) := M_W^{\bullet\bullet}(\Phi^{(t)}(f^{\bullet\bullet(t-1)}))(x, y) = \frac{1}{2}\int_{\mathcal{X}} (&\frac{W(y, z)}{c_W(x, y)}\Phi^{(t)}(f^{\bullet\bullet(t-1)}(x, y), f^{\bullet\bullet(t-1)}(x, z)) \\ &+ \frac{W(x, z)}{c_W(x, y)}\Phi^{(t)}(f^{\bullet\bullet(t-1)}(x, y), f^{\bullet\bullet(t-1)}(y, z)))d\mu(z), \end{aligned}$$

$$f^{\bullet\bullet(t)}(x, y) := \Psi^{(t)}(f^{\bullet\bullet(t-1)}(x, y), g^{\bullet\bullet(t)}(x, y)).$$

The intuition of the aggregation function is that two edges with one same node is considered neighbors in a higher-order graph [51], and to go from (x, y) to (x, z), we need to transition from y to z, which has probability $W(y, z)$. The same holds for going from (x, y) to (y, z).

Lemma 2. *If $\Phi(x, y) = y$ and $\Psi(x, y) = x/y$, then $f^{\bullet\bullet(t)}(x, y) = W(x, y)$, $\forall x, y \in \mathcal{X}$ is a stationary point in the cMPNN$^{\bullet\bullet}$, i.e. if $f^{\bullet\bullet(t-1)}(x, y) = W(x, y)$, then $f^{\bullet\bullet(t)}(x, y) = W(x, y)$, $\forall x, y \in \mathcal{X}$.*

We define the corresponding gMPNN$^{\bullet\bullet}$ as follows. First we define the *fraction of common neighbors* between nodes i and j as $c_{Ai,j} = \frac{1}{N}\sum_{z=1}^{N} A_{i,z} \cdot A_{j,z}$. If two nodes do not have common neighbors, then we set $c_{Ai,j} = \frac{1}{N}$ to avoid computation error. Further, we define $\mathbf{f}^{\bullet\bullet}{}_{i,j} = 1 \; \forall i, j \in V$ for any graph G, and $\boldsymbol{F}^{\bullet\bullet} = (\mathbf{f}^{\bullet\bullet}{}_{i,j})_{i,j\in V}$ as the pair-wise graph signals.

Definition 10 (gMPNN$^{\bullet\bullet}$)**.** *Let $(G, \boldsymbol{F})$ be a graph with graph signals as in Definition 1 and Θ be a MPNN as in Definition 2. For $t = 1, ..., T$ layers we define the gMPNN$^{\bullet\bullet}$ $\Theta_A^{\bullet\bullet(t)}$ as the mapping that maps input pairwise graph signals $\boldsymbol{F}^{\bullet\bullet(0)} = \boldsymbol{F}^{\bullet\bullet}$ to the features in the $t-th$ layer by*

$$\Theta_A^{\bullet\bullet(t)} : \mathbb{R}^{N^2\times F_0} \to \mathbb{R}^{N^2\times F_t}, \boldsymbol{F}^{\bullet\bullet(0)} \mapsto \boldsymbol{F}^{\bullet\bullet(t)} = (\mathbf{f}^{\bullet\bullet}{}^{(t)}_{i,j})_{i,j=1}^{N}$$

where $\mathbf{f}^{\bullet\bullet(t)}$ are defined recursively by the following function,

$$\begin{aligned} m^{\bullet\bullet}{}^{(t)}_{i,j} &:= \frac{1}{2N}\sum_{z=1}^{N} \frac{A_{j,z}}{c_{Ai,j}}\Phi^{(t)}(\mathbf{f}^{\bullet\bullet}{}^{(t-1)}_{i,j}, \mathbf{f}^{\bullet\bullet}{}^{(t-1)}_{i,z}) + \frac{A_{i,z}}{c_{Ai,j}}\Phi^{(t)}(\mathbf{f}^{\bullet\bullet}{}^{(t-1)}_{i,j}, \mathbf{f}^{\bullet\bullet}{}^{(t-1)}_{j,z}), \\ \mathbf{f}^{\bullet\bullet}{}^{(t)}_{i,j} &:= \Psi^{(t)}(\mathbf{f}^{\bullet\bullet}{}^{(t-1)}_{i,j}, m^{\bullet\bullet}{}^{(t)}_{i,j}). \end{aligned}$$

Next, Theorem 2 shows that these discrete joint representations gMPNN$^{\bullet\bullet}$ converge to the continuous pairwise representation cMPNN$^{\bullet\bullet}$ under the causal DAG of Figure 1.

Theorem 2 (OOD convergence without in-distribution convergence)**.** *For a random graph model (W, f) satisfying Definition 1, let N^{tr} be a random variable defining the distribution of graph sizes in training. Define the test distribution $(G^{te}, \boldsymbol{F}^{te}) \sim (W, f)$ through the causal graph in Figure 1 as an interventional change to obtain larger test graph sizes where $\min(supp(N^{te}))\gg M_{tr} = \max(supp(N^{tr}))$ (which means any test graph is much larger than the largest possible training* *graph). Let* $\Theta = ((\Phi^{(l)})_{l=1}^{T}, (\Psi^{(l)})_{l=1}^{T})$ *be a MPNN as in Definition 2 with* T *layers such that* $\Phi^{(l)}$ *and* $\Psi^{(l)}$ *that are learned from the training data and are Lipschitz continuous with Lipschitz constants* $L_{\Phi}^{(l)}(M_{tr})$ *and* $L_{\Psi}^{(l)}(M_{tr})$*. Let gMPNN*$^{\bullet\bullet}$ $\Theta_{W}^{\bullet\bullet(T)}$ *and cMPNN*$^{\bullet\bullet}$ $\Theta_{W}^{\bullet\bullet(T)}$ *be as in Definitions 9 and 10. For a random graph model* (W, f) *as in Definition 1 with* $\mathrm{d}_{cmin} > 0$*. Let* $X_1^{te}, ..., X_{N^{te}}^{te}$ *and* $\boldsymbol{A}^{te}$ *be as in Definition 1. Let* $p \in (0, \frac{1}{\sum_{l=1}^{T} 2(H_l+1)})$*. Then, if* $\frac{\sqrt{N^{te}}}{\sqrt{\log{(2(N^{te})^2/p)}}} \geq \frac{4\sqrt{2}}{\mathrm{d}_{cmin}}$*, we have with probability at least* $1 - \sum_{l=1}^{T} 2(H_l + 1)p$*,*

$$\delta_{\text{A-W}}^{\bullet\bullet} = \max_{i,j=1,\ldots,N^{te}} \|\Theta_{A}^{\bullet\bullet(T)}(\boldsymbol{F}^{\bullet\bullet})_{i,j} - \Theta_{W}^{\bullet\bullet(T)}(f^{\bullet\bullet})(X_i^{te}, X_j^{te})\|_{\infty} \leq (C_3 + C_4\|f^{\bullet\bullet}\|_{\infty}) \frac{\sqrt{\log(2(N^{te})^2/p)}}{\sqrt{N^{te}}},$$

where the constants C_3 *and* C_4 *are defined in the Appendix and depend on* $\{L_{\Phi}^{(l)}(M_{tr}), L_{\Psi}^{(l)}(M_{tr})\}_{l=1}^{T}$ *and the distribution of* $(G^{tr}, \boldsymbol{F}^{tr})$*.*

Hence, as the test graph size N^{te} gets larger w.r.t. N^{tr} (that is, as we intervene on the causal DAG of Figure 1 to change the support of the distribution of N in order to obtain larger test graphs), the link predictor learned in the training data using gMPNN$^{\bullet\bullet}$ will converge to a continuous method (cMPNN$^{\bullet\bullet}$) that can predict links in OOD tasks (i.e., $W(X_i^{\text{te}}, X_j^{\text{te}})$ is a stationary solution of cMPNN$^{\bullet\bullet}$ per Lemma 2). This convergence is also observed in our empirical results.

5 Further Related Work

In what follows we describe works related to learning transferability in GNNs. The concept of transferability of GNN is introduced by Levie et al. [35], Ruiz et al. [58], which state that if two graphs represent same phenomena (e.g., are sampled from the same distribution), then a transferable GNN has approximately the same predictive performance on both graphs. This is closely related to in-distribution generalization capabilities of GNNs to unseen test data, i.e., generalization error when train and test data come from the same distribution. Existing works [26, 44, 58, 59] prove the transferability for spectral-based GCNs under graphon models, and Maskey et al. [46] extends these results to more general message passing GNNs. The GNN smoothness conditions needed to prove uniform convergence of node-embedding equivariant GNNs in Maskey et al. [46] means their GNNs would be unable to perform *in-distribution* link prediction in some tasks (such as the graphs in Definition 7). However, in practice, we observe (Section 6) that GNNs are capable of performing these in-distribution link prediction tasks. Our results are also based on general message passing GNNs. Our goal (OOD link prediction) is, however, significantly different than these prior works, which focus on in-distribution graph and node classification. The link prediction challenge for node-embedding equivariant GNNs is either in symmetric graphs (Srinivasan and Ribeiro [65]) or OOD (this work). Theorem 1 says they vanish as the test graphs grow larger, but Theorem 2 says that our pairwise equivariant representation is capable of performing these OOD link prediction task. Related works relating to the representation power, higher order structural and positional link prediction methods (not already covered in our introduction) can be found in Appendix A due to space constraints.

6 Empirical Evaluation

In what follows we empirically validate our theoretical results in two parts. We implement all our models in Pytorch Geometric [19] and make it available[1]. Due to space constraints we relegate a detailed description of our experiments to the Appendix.

Convergence and stability. First we will empirically validate Theorems 1 and 2 and Corollary 1. Consider an SBM (Definition 6) with three blocks ($r = 3$) and $\boldsymbol{S}_{a,a} = 0.55$, $a = 1, 2, 3$, $\boldsymbol{S}_{1,2} = \boldsymbol{S}_{2,1} = 0.05$, $\boldsymbol{S}_{1,3} = \boldsymbol{S}_{3,1} = 0.02$. Note that one and three are isomorphic blocks (see Definition 7). We use a randomly initialized GraphSAGE [24] GNN model for node embedding, and test both the Φ and Ψ of Lemma 2, and a scenario where Ψ is a randomly-initialized MLP for pairwise embeddings.

Figures 2(a-c) show log-log plots of the convergence of gMPNNs to their continuous cMPNN counterparts as the test graph size N^{te} increases. The empirical approximation errors $\delta_{\text{A-W}}^{\bullet}$ (Theorem 1) (Figure 2(a)) and $\delta_{\text{A-W}}^{\bullet\bullet}$ (Theorem 2) are shown as a function of the test graph size $N^{\text{te}} = 2^n$, $n = 5, ..., 13$. The empirical results show agreement with the theory since $\delta_{\text{A-W}}^{\bullet}$ and $\delta_{\text{A-W}}^{\bullet\bullet}$ are

[1] `https://github.com/yangzez/OOD-Link-Prediction-Generalization-MPNN`

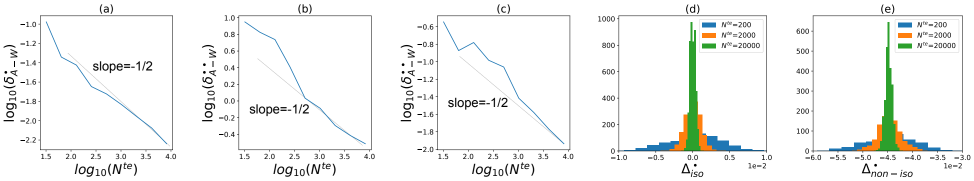

Figure 2: **Experimental agreement with theory:** (a) shows $\delta^{\bullet}_{\text{A-W}}$ (Theorem 1) of a GraphSAGE GNN as a function of N^{te}; (b) shows $\delta^{\bullet\bullet}_{\text{A-W}}$ (Theorem 2) with the gMPNN$^{\bullet\bullet}$ of Lemma 2 as a function of N^{te}; (c) replicates (b) with Ψ as a randomly-initialized neural network. Results shows close agreement with Theorems 1 and 2 that predicts slope $\approx -1/2$ in log-log scale for large N^{te}; (d) shows stable node representations between isomorphic SBM blocks, while (e) shows constant difference in node representations between non-isomorphic SBM blocks, which validate Corollary 1.

bounded above by $O(\sqrt{\log N^{\text{te}}}/\sqrt{N^{\text{te}}})$, which is approximated by the slope $-1/2$ in a log-log plot. Figures 2(d-e) show histograms of the difference between gMPNN$^{\bullet}$ embeddings of different nodes in G^{te}. Let $\Delta^{\bullet}_{i,j} := \overline{\Theta}^{\bullet(T)}_{A}(\boldsymbol{F})_i - \overline{\Theta}^{\bullet(T)}_{A}(\boldsymbol{F})_j$ for $i, j \in V^{\text{te}}$, $\Delta^{\bullet}_{i,j} \in \mathbb{R}^{F_T}$ and further define $\Delta^{\bullet}_{\text{iso (resp. non-iso)}_{i,j}} := (\Delta^{\bullet}_{i,j})_{\arg\max_k |(\Delta^{\bullet}_{i,j})_k|}$, where $k \in \{1, \ldots, F_T\}$ is the dimension of the embedding. We use subscript `iso` (resp. `non-iso`) when $i, j \in V^{\text{te}}$ are in isomorphic (resp. non-isomorphic) SBM blocks (Definition 7). As N^{te} increases, Figure 2(d) shows that embeddings between isomorphic blocks converge, validating Corollary 1, while Figure 2(e) shows that non-isomorphic blocks do not.

Link prediction performance evaluation with SBMs (in-distribution and OOD). In what follows we introduce empirical results using a SBM similar in the previous setting. Details can be found in Appendix B.3. We start by sampling the training graph $(G^{\text{tr}}, \boldsymbol{F}^{\text{tr}})$ with $N^{\text{tr}} = 10^3$ nodes. We randomly hide 10% of E^{tr} from the original graph G^{tr} for link prediction purpose since the goal of link prediction is to predict possible missing links that is not observed in the original graph. We call these edges $E^{\text{hid-tr}}$. Then we split $E^{\text{hid-tr}}$ into positive train (80%) and validation (10%) edges (we reserve 10% of $E^{\text{hid-tr}}$ for the transductive test scenario), and uniformly sample the same number of across-block non-edges as negative train and validation edges. The embedding method gMPNN$^{\bullet}$ (resp. gMPNN$^{\bullet\bullet}$) along with link predictor $\eta^{\bullet}$ (resp. $\eta^{\bullet\bullet}$) are trained in an end-to-end manner for predicting positive and negative edges in training using cross-entropy loss. Our experiments consider three scenarios (in all scenarios we use the same number of negative test edges as positive test edges, sampled from non-edges in G^{te} with endpoints in different isomorphic blocks): (i) (In-distribution) transductive scenario where $G^{\text{te}} = G^{\text{tr}}$, where positive test edges are the 10% reserved in $E^{\text{hid-tr}}$ not used in training or validation; (ii) In-distribution inductive scenario where G^{te} is sampled from the same SBM with $N^{\text{te}} = N^{\text{tr}}$, where we also hide 10% of the edges and sample $0.1|E^{\text{hid-tr}}|$ positive test edges from $E^{\text{hid-te}}$ (for fair comparison across all scenarios); (c) OOD inductive scenario where G^{te} is sampled from the same SBM with $N^{\text{te}} = 10 \times N^{\text{tr}}$, where we also hide 10% of the edges and sample $0.1|E^{\text{hid-tr}}|$ positive test edges from $E^{\text{hid-te}}$(for fair comparison across all scenarios).

For *structural node embeddings* we consider GraphSAGE [24], GCN [28] (without positional features), GAT [70] and GIN [78] as the representatives of gMPNN$^{\bullet}$ models. The link predictor $\eta^{\bullet}$ is as feedfoward network (with 3 hidden layers and 10 neurons each) that receives the two node embeddings as input, and has link prediction threshold $\tau = 0.5$ (see Definition 8 for details).

For *structural pairwise embeddings* we choose our proposed gMPNN$^{\bullet\bullet}$ method of Definition 10, since we can prove that our approach is theoretically sound in Lemma 2. We test gMPNN$^{\bullet\bullet}$ in two versions: The Φ and Ψ functions in Lemma 2 (denoted *fixed* Ψ) and a feedforward neural network for Ψ with 2 hidden layers and 5 neurons each (denoted *learn* Ψ). The link predictor $\eta^{\bullet\bullet}$ is the same as $\eta^{\bullet}$ except it just takes one pairwise embedding as input, rather than two node embeddings.

Table 1 presents our empirical results. The oracle predictor knows the graphon values $W(X^{\text{te}}_i, X^{\text{te}}_j)$. Our evaluation metrics include the Matthews correlation coefficient (mcc) [47], balanced accuracy, and Hits@K for $K = 10, 50, 100$ that counts the ratio of positive edges ranked at the k-th place or above against all negative edges. Note that gMPNN$^{\bullet}$ structural node representations can very accurately predict links in the transductive tasks, and still performs reasonably well in inductive in-distribution tasks. However, as expected from Corollary 2, this performance suffers significantly as N^{te} becomes 10× larger than N^{tr}. Now all gMPNN$^{\bullet}$ methods produce predictors that are no better than a random guess over all metrics (e.g., see OOD mcc and accuracy (in red)). In contrast, the gMPNN$^{\bullet\bullet}$ is able to consistently offer good performance on both in-distribution and OOD tasks.

Table 1: Test performance over 50 runs of node and pairwise gMPNNs for in-distribution and OOD link prediction over SBM graphs. Methods marked with $*$ indicate best result out of distinct configurations detailed in the Appendix.

			Training graph size $\boldsymbol{N^{\text{tr}} = 10^3}$				
Tasks		Model	Hit@10(%)	Hit@50(%)	Hit@100(%)	mcc.(%)	balanced acc.(%)
In-distribution link prediction	Transductive	GraphSAGE*	95.55(0.52)	95.93(0.73)	96.14(0.74)	**95.42**(0.37)	**97.66**(0.19)
		GCN*	93.15(14.57)	93.99(13.08)	94.35(12.72)	92.41(14.72)	95.97(8.24)
		GAT*	93.77(13.03)	94.01(13.02)	94.14(13.03)	90.94(16.09)	95.26(8.38)
		GIN*	95.77(0.59)	96.09(0.58)	96.28(0.59)	95.48(0.41)	97.69(0.22)
		gMPNN$^{\bullet\bullet}$ (fixed Ψ)	93.76(0.55)	94.17(0.51)	94.51(0.49)	93.64(0.53)	96.72(0.28)
		gMPNN$^{\bullet\bullet}$ (learn Ψ)	**96.71(0.32)**	**96.88(0.31)**	**97.00(0.30)**	94.23(0.55)	97.03(0.29)
		Oracle	96.92(0.36)	96.92(0.36)	96.92(0.36)	93.74(0.42)	96.77(0.22)
	Inductive $N^{\text{te}} = N^{\text{tr}}$	GraphSAGE*	47.38(39.08)	52.13(38.87)	54.94(37.83)	19.34(43.19)	61.46(20.17)
		GCN*	66.29(37.67)	68.52(35.87)	69.92(35.12)	31.76(35.12)	67.21(22.75)
		GAT*	40.05(39.05)	41.34(39.39)	41.96(39.54)	19.44(35.22)	59.52(16.94)
		GIN*	39.33(34.62)	42.93(33.86)	43.90(33.72)	18.59(39.43)	59.79(18.24)
		gMPNN$^{\bullet\bullet}$ (fixed Ψ)	93.85(0.49)	94.23(0.51)	94.55(0.49)	93.74(0.48)	96.77(0.25)
		gMPNN$^{\bullet\bullet}$ (learn Ψ)	**96.71**(0.30)	**96.91**(0.28)	**97.02**(0.27)	**94.23**(0.59)	**97.03**(0.31)
		Oracle	97.01(0.31)	97.01(0.31)	97.01(0.31)	93.87(0.39)	96.84(0.20)
OOD link prediction	**Inductive $N^{\text{te}} = 10^4$**	GraphSAGE*	9.97(19.47)	11.73(21.80)	12.98(23.70)	-6.56(5.12)	49.32(0.60)
		GCN*	39.29(31.33)	42.15(30.81)	44.19(30.97)	-4.88(14.84)	50.33(6.72)
		GAT*	27.31(26.93)	28.13(26.78)	28.72(26.93)	-2.00(8.96)	50.20(3.37)
		GIN*	0.00(0.00)	0.00(0.00)	0.00(0.00)	-3.93(5.12)	49.59(0.57)
		gMPNN$^{\bullet\bullet}$ (fixed Ψ)	**96.74**(0.07)	**96.93**(0.04)	**97.01**(0.04)	**93.76**(0.05)	**96.78**(0.03)
		gMPNN$^{\bullet\bullet}$ (learn Ψ)	**96.97**(0.04)	**97.02**(0.04)	**97.08**(0.04)	**93.94**(0.67)	**96.88**(0.35)
		Oracle	96.96(0.03)	96.96(0.03)	96.96(0.03)	93.77(0.04)	96.79(0.02)

Link prediction performance evaluation with ogbl-ddi (in-distribution and OOD). In what follows we introduce empirical results using the ogbl-ddi dataset, which represents a drug-drug interaction network. For the purpose of performing OOD tasks, we start by sampling 10% of the nodes (427 nodes) and its induced subgraph to be the training graph. Further experimental details can be found in Appendix B.4. The in-distribution inductive scenario has G^{te} constructed as an induced subgraph with $N^{\text{te}} = N^{\text{tr}}$ nodes from the remaining ogbl-ddi graph. Our OOD inductive scenario has G^{te} as the induced subgraph without the training nodes ($N^{\text{te}} = 3840$ nodes). The test edges are obtained by applying the original edge split on the newly induced test subgraph, where we further down-sample to the same amount of test edges as in our in-distribution scenarios for fair comparison across all scenarios. Table 3 in the Appendix presents our empirical results on the ogbl-ddi link prediction task. All gMPNN$^{\bullet}$ methods performs worse in inductive settings than transductive settings, and suffer much worse performance in OOD transductive setting except GCNs. In contrast, the gMPNN$^{\bullet\bullet}$ is able to consistently offer good performance on both in-distribution and OOD tasks, showing that the theoretical results are not limited to SBM models.

7 Conclusions

This work studied and provided the first theoretical framework for the task of out-of-distribution (OOD) link prediction, where test graphs are larger than training graphs. Using non-asymptotic bounds, this work showed that OOD link prediction methods using structural node embeddings given by message-passing GNNs converge to link predictors that may perform no better than random guesses. The work also proposed a theoretically-sound structural pairwise embedding with a message-passing algorithm which is able to perform our OOD link prediction task by being approximately invariant to interventions on test graph sizes, as the discrete joint embedding converges to the continuous one. This means that as graph sizes grow in test (OOD), it is still possible to find neural networks parameters that allows our joint representation to converge to the true link probability. We show that the same is not guaranteed for node-embedding equivariant message-passing GNNs. Extensive empirical evaluation showed agreement with these theoretical results. We do not foresee adverse social impacts for this theoretical work, but it does raise awareness of the shortcomings of node-embedding equivariant massage-passing GNNs for link prediction tasks in applications such as recommender systems.

References

[1] Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social networks*, 25(3): 211–230, 2003.

[2] Amr Ahmed, Nino Shervashidze, Shravan Narayanamurthy, Vanja Josifovski, and Alexander J Smola. Distributed large-scale natural graph factorization. In *Proceedings of the 22nd international conference on World Wide Web*, pages 37–48, 2013.

[3] Edo M Airoldi, Thiago B Costa, and Stanley H Chan. Stochastic blockmodel approximation of a graphon: Theory and consistent estimation. *Advances in Neural Information Processing Systems*, 26, 2013.

[4] Gabor Angeli and Christopher D Manning. Philosophers are mortal: Inferring the truth of unseen facts. In *Proceedings of the seventeenth conference on computational natural language learning*, pages 133–142, 2013.

[5] Fabian Ball and Andreas Geyer-Schulz. How symmetric are real-world graphs? a large-scale study. *Symmetry*, 10(1):29, 2018.

[6] Robert M Bell and Yehuda Koren. Lessons from the netflix prize challenge. *Acm Sigkdd Explorations Newsletter*, 9(2):75–79, 2007.

[7] Beatrice Bevilacqua, Yangze Zhou, and Bruno Ribeiro. Size-invariant graph representations for graph classification extrapolations. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 837–851. PMLR, 2021.

[8] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34 (4):18–42, 2017.

[9] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations*, 2014.

[10] Yongqiang Chen, Yonggang Zhang, Han Yang, Kaili Ma, Binghui Xie, Tongliang Liu, Bo Han, and James Cheng. Invariance principle meets out-of-distribution generalization on graphs. *CoRR*, abs/2202.05441, 2022.

[11] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, pages 271–280, 2007.

[12] Luc De Raedt. *Logical and relational learning*. Springer Science & Business Media, 2008.

[13] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

[14] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[15] Persi Diaconis and Svante Janson. Graph limits and exchangeable random graphs. *arXiv preprint arXiv:0712.2749*, 2007.

[16] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.

[17] Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. In *International Conference on Learning Representations*, 2022.

[18] Paul Erdos and Alfréd Rényi. Asymmetric graphs. *Acta Math. Acad. Sci. Hungar*, 14(295-315): 3, 1963.

[19] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[20] Lise Getoor and Christopher P Diehl. Link mining: a survey. *Acm Sigkdd Explorations Newsletter*, 7(2):3–12, 2005.

[21] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1263–1272. PMLR, 2017.

[22] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE international joint conference on neural networks*, volume 2, pages 729–734, 2005.

[23] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proc. of KDD*, pages 855–864. ACM, 2016.

[24] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[25] 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. *Advances in neural information processing systems*, 33:22118–22133, 2020.

[26] Nicolas Keriven, Alberto Bietti, and Samuel Vaiter. Convergence and stability of graph convolutional networks on large random graphs. *Advances in Neural Information Processing Systems*, 33:21512–21523, 2020.

[27] Jeong Han Kim, Benny Sudakov, and Van H Vu. On the asymmetry of random regular graphs and random graphs. *Random Structures & Algorithms*, 21(3-4):216–224, 2002.

[28] Thomas Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[29] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *NIPS Workshop on Bayesian Deep Learning*, 2016.

[30] Daphne Koller, Nir Friedman, Sašo Džeroski, Charles Sutton, Andrew McCallum, Avi Pfeffer, Pieter Abbeel, Ming-Fai Wong, Chris Meek, Jennifer Neville, et al. *Introduction to statistical relational learning*. MIT press, 2007.

[31] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[32] Yehuda Koren, Steffen Rendle, and Robert Bell. Advances in collaborative filtering. *Recommender systems handbook*, pages 91–142, 2022.

[33] Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34, 2021.

[34] Neil Lawrence and Aapo Hyvärinen. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of machine learning research*, 6(11), 2005.

[35] Ron Levie, Wei Huang, Lorenzo Bucci, Michael Bronstein, and Gitta Kutyniok. Transferability of spectral graph convolutional neural networks. *Journal of Machine Learning Research*, 22 (272):1–59, 2021.

[36] Pan Li, Yanbang Wang, Hongwei Wang, and Jure Leskovec. Distance encoding: Design provably more powerful neural networks for graph representation learning. *Advances in Neural Information Processing Systems*, 33:4465–4478, 2020.

[37] 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.

[38] Derek Lim, Joshua Robinson, Lingxiao Zhao, Tess Smidt, Suvrit Sra, Haggai Maron, and Stefanie Jegelka. Sign and basis invariant networks for spectral graph representation learning. *arXiv preprint arXiv:2202.13013*, 2022.

[39] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003.

[40] László Lovász and Balázs Szegedy. The automorphism group of a graphon. *Journal of Algebra*, 421:136–166, 2015.

[41] Tomasz Luczak, Abram Magner, and Wojciech Szpankowski. Asymmetry and structural information in preferential attachment graphs. *Random Structures & Algorithms*, 55(3):696–718, 2019.

[42] Ben D MacArthur, Rubén J Sánchez-García, and James W Anderson. Symmetry in complex networks. *Discrete Applied Mathematics*, 156(18):3525–3531, 2008.

[43] Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. In *Advances in Neural Information Processing Systems*, pages 2156–2167, 2019.

[44] Sohir Maskey, Ron Levie, and Gitta Kutyniok. Transferability of graph neural networks: an extended graphon approach. *arXiv preprint arXiv:2109.10096*, 2021.

[45] Sohir Maskey, Yunseok Lee, Ron Levie, and Gitta Kutyniok. Convergence and transferability of message passing graph neural networks. *(under submission) currently available as arXiv:2202.00645v3*, 2022.

[46] Sohir Maskey, Ron Levie, Yunseok Lee, and Gitta Kutyniok. Generalization analysis of message passing neural networks on large random graphs. *Advances in neural information processing systems*, 2022.

[47] Brian W Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975.

[48] Brendan McKay. Practical graph isomorphism. *Dagstuhl Reports, Vol. 5, Issue 12 ISSN 2192-5283*, page 11, 2016.

[49] Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20, 2007.

[50] Federico Monti, Oleksandr Shchur, Aleksandar Bojchevski, Or Litany, Stephan Günnemann, and Michael M Bronstein. Dual-primal graph convolutional networks. *arXiv preprint arXiv:1806.00770*, 2018.

[51] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4602–4609, 2019.

[52] Christopher Morris, Yaron Lipman, Haggai Maron, Bastian Rieck, Nils M Kriege, Martin Grohe, Matthias Fey, and Karsten Borgwardt. Weisfeiler and leman go machine learning: The story so far. *arXiv preprint arXiv:2112.09992*, 2021.

[53] Ryan Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Relational pooling for graph representations. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[54] 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, 2015.

[55] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[56] 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*, pages 701–710, 2014.

[57] 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, 2006.

[58] Luana Ruiz, Luiz Chamon, and Alejandro Ribeiro. Graphon neural networks and the transferability of graph neural networks. *Advances in Neural Information Processing Systems*, 33: 1702–1712, 2020.

[59] Luana Ruiz, Fernando Gama, and Alejandro Ribeiro. Graph neural networks: architectures, stability, and transferability. *Proceedings of the IEEE*, 109(5):660–682, 2021.

[60] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. *Advances in neural information processing systems*, 30, 2017.

[61] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[62] Brent Smith and Greg Linden. Two decades of recommender systems at amazon. com. *Ieee internet computing*, 21(3):12–18, 2017.

[63] Tom AB Snijders and Krzysztof Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of classification*, 14(1):75–100, 1997.

[64] Alessandro Sperduti and Antonina Starita. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–735, 1997.

[65] Balasubramaniam Srinivasan and Bruno Ribeiro. On the equivalence between positional node embeddings and structural graph representations. *ICLR*, 2020.

[66] Ben Taskar, Ming-Fai Wong, Pieter Abbeel, and Daphne Koller. Link prediction in relational data. *Advances in neural information processing systems*, 16, 2003.

[67] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR, 2016.

[68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[69] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[70] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *ICLR*, 2018.

[71] Haorui Wang, Haoteng Yin, Muhan Zhang, and Pan Li. Equivariant and stable positional encoding for more powerful graph neural networks. In *International Conference on Learning Representations*, 2022.

[72] Yanbang Wang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. Inductive representation learning in temporal networks via causal anonymous walks. In *International Conference on Learning Representations*, 2021.

[73] Boris Weisfeiler and AA Lehman. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia*, 2(9):12–16, 1968.

[74] David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, et al. Drugbank 5.0: a major update to the drugbank database for 2018. *Nucleic acids research*, 46(D1):D1074–D1082, 2018.

[75] Patrick J Wolfe and Sofia C Olhede. Nonparametric graphon estimation. *arXiv preprint arXiv:1309.5936*, 2013.

[76] Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf. Handling distribution shifts on graphs: An invariance perspective. In *International Conference on Learning Representations*, 2022.

[77] Yingxin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. Discovering invariant rationales for graph neural networks. In *International Conference on Learning Representations*, 2022.

[78] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

[79] Keyulu Xu, Mozhi Zhang, Jingling Li, Simon Shaolei Du, Ken-Ichi Kawarabayashi, and Stefanie Jegelka. How neural networks extrapolate: From feedforward to graph neural networks. In *International Conference on Learning Representations*, 2021.

[80] Keyulu Xu, Mozhi Zhang, Jingling Li, Simon Shaolei Du, Ken-Ichi Kawarabayashi, and Stefanie Jegelka. How neural networks extrapolate: From feedforward to graph neural networks. In *International Conference on Learning Representations*, 2021.

[81] Gilad Yehudai, Ethan Fetaya, Eli Meirom, Gal Chechik, and Haggai Maron. From local structures to size generalization in graph neural networks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11975–11986. PMLR, 18–24 Jul 2021.

[82] Jiaxuan You, Rex Ying, and Jure Leskovec. Position-aware graph neural networks. In *International Conference on Machine Learning*, pages 7134–7143. PMLR, 2019.

[83] Jiaxuan You, Jonathan Gomes-Selman, Rex Ying, and Jure Leskovec. Identity-aware graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[84] Muhan Zhang and Yixin Chen. Weisfeiler-lehman neural machine for link prediction. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 575–583, 2017.

[85] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31, 2018.

[86] Muhan Zhang, Pan Li, Yinglong Xia, Kai Wang, and Long Jin. Labeling trick: A theory of using graph neural networks for multi-node representation learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[87] Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Advances in Neural Information Processing Systems*, 34, 2021.

Appendix of "*OOD Link Prediction Generalization Capabilities of Message-Passing GNNs in Larger Test Graphs*"

In Appendix A, we introduce more related work that has not been discussed in the main paper. In Appendix B, we provide more details in experiments set up and model training. In Appendix C, we introduce notations and definitions that we will use throughout the rest of the appendix. In Appendix D, we show large random and real world graphs have few isomorphic nodes. In Appendix E, we prove the convergence results (Theorem 1) for gMPNN$^{\bullet}$ when different aggregation functions are used. In Appendix F, we prove the results for hardness of link prediction for gMPNN$^{\bullet}$. Finally, we prove the convergence results for gMPNN$^{\bullet\bullet}$ and cMPNN$^{\bullet\bullet}$ (Theorem 2) in Appendix G.

A Further Related Work

Representation power of GNNs. The representation power of GNNs is widely studied in recent years. [51, 78] first show that gMPNN is no more powerful than 1-WL test [73]. Many works have been proposed [43, 51–53] to increase the representation power of GNNs for graph representation, but little has studied on representation power for node and link prediction.

Structural link prediction. Existing link prediction methods assume that, with powerful enough node representations, combining them can guarantee powerful link representations [23, 29]. However, Hu et al. [25] empirically shows that these approaches perform worse than simple heuristic approaches such as Common Neighbor and Adamic-Adar [1, 37]. Theoretically, Srinivasan and Ribeiro [65] was the first work to formally analyze the difference between structural node and link representations, and show that even most-expressive structural node representations are not able to perform link prediction tasks in graphs with high degree of symmetry. In order to remedy this, the state-of-the-art link prediction methods like SEAL [85] use GNNs but transform the task into a graph classification task (the link is an attribute of an induced subgraph around the two target end nodes), where each node in the subgraph are labeled according to their distances to the pair of target end nodes. Zhang et al. [86] unifies such approaches [36, 83, 85] through a method they call "labeling trick", which they show is able to learn structural link representations with a node-most expressive GNN.

Ability of GNNs to emulate graph algorithms as graph sizes increase. Recently, Xu et al. [79] shows that GNNs can extrapolate in algorithmic-related tasks as the graph size grows, if the GNN uses max as an aggregator (rather than mean and sum we considered in this paper). Unfortunately, our Definition 3 of gMPNN$^{\bullet}$ does not allow max aggregators, in part because it is unclear how one could reach stability using the max aggregator. Fortunately, while we could not obtain theoretical results using the max aggregator, we can test it empirically. Table 2 reproduces all our empirical results using the max aggregator (on GraphSAGE and GAT, since these are the only GNNs designed for the max aggregator). Our experiments show that the max aggregator, just like the mean and sum aggregators, shows poor OOD performance as test graph sizes increase. Other works Bevilacqua et al. [7], Yehudai et al. [81] also talk about graph extrapolation as size grows but focus on graph classification. Chen et al. [10], Wu et al. [76, 77] also explore environment-invarian GNN representations for graph classification or node classification tasks. These works differ in that they focus on node classification and graph classification. As Srinivasan and Ribeiro [65] shows, link prediction tasks are significantly different from graph and node classification. Moreover, whether or not one can prove that the max aggregator is or is not able to perform our OOD task is left as future work.

Positional node embeddings for link prediction. Another way to perform link prediction tasks is to use positional node embeddings (PE), which preserves relative positions of the nodes in a graph. The original link predictor in Kipf and Welling [28] uses positional embedding as node attributes for this type of task. However, such approaches can lose the desired permutation equivariance property in graph models. Traditional PE methods include DeepWalk [56] and matrix factorization [2, 49]. You et al. [82] proposes position-aware GNN that only aggregates message from randomly selected anchor nodes, which has poor generalization ability on inductive tasks. Srinivasan and Ribeiro [65] proves that using set representations of PE embeddings over all permutations of the graph input and all random decisions made by the embedding algorithm (e.g., the set of all eigenvectors of randomly permuted graph Laplacian matrices and random eigenvectors due to geometric multiplicity of eigenvalues) can achieve the desired permutation equivariance for link prediction. Dwivedi and

Bresson [16], Kreuzer et al. [33] propose PE that randomly flips of the sign of eigenvectors to alleviate sign ambiguity and pass it to a transformers architecture [68]. Lim et al. [38] proposes a representation that is invariant to the elements of the set described by Srinivasan and Ribeiro [65] in order to achieve equivariant representations for spectral node embeddings. Wang et al. [71] proposes a provable solution for using PEs to learn equivariant and stable representation using separate channels in GNN layers. Dwivedi et al. [17] turns to the idea of learning PE that can be combined with structural representations, and design architecture to decouple structural and positional representations in order to improve both representations.

B Further Experiment details

In this section we present the details of the experimental section, discussing implementation details. Training was performed on NVIDIA Telsa P100, GeForce RTX 2080 Ti, and TITAN V GPUs.

B.1 Model implementation

All neural network approaches, including the models proposed in this paper, are implemented in PyTorch [55] and Pytorch Geometric [19] (available respectively under the BSD and MIT license).

Our GIN [78], GCN [28], GAT [70] and GraphSAGE [24] implementations are based on their Pytorch Geometric implementations. We also consider max aggregation as proposed by Xu et al. [80] for extrapolations although it does not fit our theoretical framework.

We use the Adam optimizer to optimize all the neural network models. We use the neural network weights that achieve best validation-set performance for prediction.

B.2 Empirical validation for convergence and Stability

Consider an SBM (Definition 6) with three blocks ($r = 3$) and $\boldsymbol{S}_{a,a} = 0.55$, $a = 1, 2, 3$, $\boldsymbol{S}_{1,2} = \boldsymbol{S}_{2,1} = 0.05$, $\boldsymbol{S}_{1,3} = \boldsymbol{S}_{3,1} = 0.02$. The probability a node belongs to block one or three is 0.45, while for block two it is 0.1. Note that one and three are isomorphic blocks (see Definition 7). Since our results are valid for any gMPNN functions Θ, for our first experiment with node embeddings we use a randomly initialized GraphSAGE [24] GNN model, where following standard GNN procedures we initialize node features as size-normalized degrees (where $d_i = \frac{1}{N}\sum_{j=1,\dots,N} A_{i,j}$). For the experiment with pairwise embeddings, we test both the Φ and Ψ of Lemma 2, and a scenario where Ψ is a randomly-initialized feedforward neural network. Later in this section we show how to efficiently compute the exact cMPNN$^\bullet$ and cMPNN$^{\bullet\bullet}$ embeddings of our GraphSAGE and gMPNN$^{\bullet\bullet}$ models.

The validation procedure follows Maskey et al. [45]. We use SBM graphs as examples. Consider an SBM (Definition 6) with three blocks ($r = 3$) and $\boldsymbol{S} = \begin{bmatrix} 0.55 & 0.05 & 0.02 \\ 0.05 & 0.55 & 0.05 \\ 0.02 & 0.05 & 0.55 \end{bmatrix}$. The probability a node belongs to block one or three is 0.45, while for block two it is 0.1. The in-block edge probability is 0.55, and across-isomorphic block probability is 0.02 and across-non-isomorphic block probability is 0.05. Note that blocks one and three are isomorphic blocks (see Definition 7).

Since our results is valid for any gMPNN functions, we use a randomly initialized GraphSAGE [24] GNN model for our first experiments with node embeddings. Following our Definition 6, the initial node embeddings within the same block should be the same, however, following standard GNN procedures we initialize node features as size-normalized degrees. Note that in theory, node within each block has the same expected graphon degree, but this setting is more realistic and shows a stronger results than proposed in our theorems when initial node embeddings also have variance.

To efficiently calculate exact cMPNN$^\bullet$ embeddings, we need to make use of the property of SBMs, i.e. the graphon values within a block is constant. The graphon degree d_W for nodes in block 1, 2 and 3 is $0.2615, 0.1$ and 0.2615. Then we can write the integral $\int_{\mathcal{X}} \frac{W(x,y)}{d_W(x)} \Phi^{(t)}(\overline{f}^{\bullet(t-1)}(x), \overline{f}^{\bullet(t-1)}(y)) d\mu(y)$ as $\frac{1}{0.2615}(0.45 \times \boldsymbol{S}_{1,1}\Phi^{(t)}(\boldsymbol{B}_1^{(t-1)}, \boldsymbol{B}_1^{(t-1)}) + 0.1 \times \boldsymbol{S}_{1,2}\Phi^{(t)}(\boldsymbol{B}_1^{(t-1)}, \boldsymbol{B}_2^{(t-1)}) + 0.45 \times \boldsymbol{S}_{1,3}\Phi^{(t)}(\boldsymbol{B}_1^{(t-1)}, \boldsymbol{B}_3^{(t-1)}))$. This can be calculated exactly by extracting the neural network weights from the GNN model for Φ and Ψ.

Table 2: Test performance over 50 runs of node and pairwise gMPNNs for in-distribution and OOD link prediction over SBM graphs. Methods marked with $*$ indicate best result out of distinct configurations detailed in the Appendix.

			Training graph size $\boldsymbol{N}^{\text{tr}} = \mathbf{10^3}$				
Tasks		Model	Hit@10(%)	Hit@50(%)	Hit@100(%)	mcc.(%)	balanced acc.(%)
In-distribution link prediction	Transductive	GraphSAGE*	95.55(0.52)	95.93(0.73)	96.14(0.74)	**95.42**(0.37)	**97.66**(0.19)
		GraphSAGE(max)*	95.43(0.38)	96.13(0.57)	96.54(0.60)	**95.38**(0.36)	**97.64**(0.19)
		GCN*	93.15(14.57)	93.99(13.08)	94.35(12.72)	92.41(14.72)	95.97(8.24)
		GAT*	93.77(13.03)	94.01(13.02)	94.14(13.03)	90.94(16.09)	95.26(8.38)
		GAT(max)*	92.91(12.27)	93.88(9.12)	94.08(8.82)	87.36(20.41)	93.34(10.95)
		GIN*	95.77(0.59)	96.09(0.58)	96.28(0.59)	95.48(0.41)	97.69(0.22)
		gMPNN$^{\bullet\bullet}$ (fixed Ψ)	93.76(0.55)	94.17(0.51)	94.51(0.49)	93.64(0.53)	96.72(0.28)
		gMPNN$^{\bullet\bullet}$ (learn Ψ)	**96.71(0.32)**	**96.88(0.31)**	**97.00(0.30)**	94.23(0.55)	97.03(0.29)
		Oracle	96.92(0.36)	96.92(0.36)	96.92(0.36)	93.74(0.42)	96.77(0.22)
	Inductive $N^{\text{te}} = N^{\text{tr}}$	GraphSAGE*	47.38(39.08)	52.13(38.87)	54.94(37.83)	19.34(43.19)	61.46(20.17)
		GraphSAGE(max)*	17.72(22.89)	25.91(27.75)	31.43(30.18)	18.24(30.43)	58.65(14.53)
		GCN*	66.29(37.67)	68.52(35.87)	69.92(35.12)	31.76(35.12)	67.21(22.75)
		GAT*	40.05(39.05)	41.34(39.39)	41.96(39.54)	19.44(35.22)	59.52(16.94)
		GAT(max)*	41.98(39.23)	43.34(38.71)	43.54(38.69)	22.66(38.99)	61.46(19.02)
		GIN*	39.33(34.62)	42.93(33.86)	43.90(33.72)	18.59(39.43)	59.79(18.24)
		gMPNN$^{\bullet\bullet}$ (fixed Ψ)	93.85(0.49)	94.23(0.51)	94.55(0.49)	93.74(0.48)	96.77(0.25)
		gMPNN$^{\bullet\bullet}$ (learn Ψ)	**96.71**(0.30)	**96.91**(0.28)	**97.02**(0.27)	**94.23**(0.59)	**97.03**(0.31)
		Oracle	97.01(0.31)	97.01(0.31)	97.01(0.31)	93.87(0.39)	96.84(0.20)
OOD link prediction	**Inductive $N^{\text{te}} = 10^4$**	GraphSAGE*	9.97(19.47)	11.73(21.80)	12.98(23.70)	-6.56(5.12)	49.32(0.60)
		GraphSAGE(max)*	1.44(2.35)	2.60(4.76)	3.58(6.53)	-2.52(4.44)	49.83(0.57)
		GCN*	39.29(31.33)	42.15(30.81)	44.19(30.97)	-4.88(14.84)	50.33(6.72)
		GAT*	27.31(26.93)	28.13(26.78)	28.72(26.93)	-2.00(8.96)	50.20(3.37)
		GAT(max)*	32.56(26.94)	33.01(27.16)	33.24(27.27)	-2.85(9.76)	49.82(3.43)
		GIN*	0.00(0.00)	0.00(0.00)	0.00(0.00)	-3.93(5.12)	49.59(0.57)
		gMPNN$^{\bullet\bullet}$ (fixed Ψ)	**96.74**(0.07)	**96.93**(0.04)	**97.01**(0.04)	**93.76**(0.05)	**96.78**(0.03)
		gMPNN$^{\bullet\bullet}$ (learn Ψ)	**96.97**(0.04)	**97.02**(0.04)	**97.08**(0.04)	**93.94**(0.67)	**96.88**(0.35)
		Oracle	96.96(0.03)	96.96(0.03)	96.96(0.03)	93.77(0.04)	96.79(0.02)

Then we compare the difference between gMPNN$^{\bullet}$ and cMPNN$^{\bullet}$ for increasing number of nodes. We first plot log-log plots, where a $O(\frac{1}{\sqrt{N}})$ decay rate will have slope $-\frac{1}{2}$ in the log-log plot. Our theory bounds the decay rate by $O(\frac{\log N}{\sqrt{N}})$, which can be approximated by the $-\frac{1}{2}$ slope and is validated in Figure 2.

Pairwise embeddings For the experiment with pairwise embeddings, we test both the Φ and Ψ of Lemma 2, and a scenario where Ψ is a randomly initialized two layer feed-forward neural network. To compute the cMPNN$^{\bullet\bullet}$ embeddings, without choosing the adjacency matrix as input to the model, we can input the graphon value matrix $\boldsymbol{W}$ where $\boldsymbol{W}_{i,j} = W(X_i, X_j)$. In our experiment, we choose graph with 20 nodes, 9 in block 1, 2 in block 2, and 9 in block 3. The result of cMPNN$^{\bullet\bullet}$ is stable for graphs with different sizes. Then we plot the same log-log plot as above.

B.3 Link prediction performance evaluation with SBMs

First, we use a slightly modified SBM with $\boldsymbol{S} = \begin{bmatrix} 0.6 & 0.05 & 0.02 \\ 0.05 & 0.6 & 0.05 \\ 0.02 & 0.05 & 0.6 \end{bmatrix}$ with other things the same as in the above subsection. Here we increase the in-block edge probability to 0.6 since we are going to hide edges for link prediction purpose.

We start by sampling the training graph $(G^{\text{tr}}, \boldsymbol{F}^{\text{tr}})$ with $N^{\text{tr}} = 10^3$ nodes. We randomly hide 10% of E^{tr} from the original graph G^{tr} for link prediction purpose since the goal of link prediction is to predict possible missing links that is not observed in the original graph. We call these edges $E^{\text{hid-tr}}$.

Then we split $E^{\text{hid-tr}}$ into positive train (80%) and validation (10%) edges (we reserve 10% of $E^{\text{hid-tr}}$ for the transductive test scenario), and uniformly sample the same number of across-block non-edges

Table 3: Test performance over 50 runs of node and pairwise gMPNNs for in-distribution and OOD link prediction over the ogbl-ddi graph. Methods marked with $*$ indicate best result out of distinct configurations detailed in the Appendix.

			Training graph size $\boldsymbol{N^{\text{tr}} = 427}$				
Tasks		Model	Hit@10(%)	Hit@50(%)	Hit@100(%)	mcc.(%)	balanced acc.(%)
In-distribution link prediction	Transductive	GraphSAGE*	30.23(2.03)	47.70(1.75)	60.36(1.79)	71.47(0.70)	**85.72(0.36)**
		GCN*	17.91(0.52)	33.69(0.60)	44.34(0.85)	59.45(0.50)	78.85(0.36)
		GAT*	1.46(0.52)	8.20(1.34)	16.37(1.95)	52.64(1.62)	74.75(0.61)
		GIN*	17.21(4.74)	28.76(5.79)	37.46(6.60)	54.27(1.59)	76.84(1.19)
		gMPNN$^{\bullet\bullet}$ (fixed Ψ)	14.09(0.06)	50.32(0.01)	65.41(0.01)	**73.23(0.10)**	**86.60(0.04)**
		gMPNN$^{\bullet\bullet}$ (learn Ψ)	**38.60(1.68)**	**59.04(0.22)**	**68.63(0.06)**	71.96(0.06)	**85.74(0.03)**
		Random	0.48(2.58)	1.16(4.58)	2.01(6.54)	0.05(0.39)	50.00(0.01)
	Inductive $N^{\text{te}} = N^{\text{tr}}$	GraphSAGE*	10.52(1.33)	23.85(1.29)	36.60(1.37)	47.58(2.98)	71.59(2.46)
		GCN*	10.76(0.90)	24.79(0.73)	34.99(0.70)	50.82(0.19)	74.73(0.21)
		GAT*	0.07(0.02)	0.22(0.10)	0.51(0.07)	-0.93(0.77)	50.00(0.01)
		GIN*	10.95(4.19)	24.42(5.75)	33.71(6.70)	40.67(2.36)	66.24(1.75)
		gMPNN$^{\bullet\bullet}$ (fixed Ψ)	34.24(0.07)	66.87(0.03)	73.91(0.02)	**67.89(0.34)**	**83.76(0.20)**
		gMPNN$^{\bullet\bullet}$ (learn Ψ)	**56.45(0.08)**	**68.42(0.03)**	**74.93(0.02)**	65.55(0.15)	82.62(0.09)
		Random	0.41(1.64)	2.20(4.88)	4.97(8.77)	-0.03(0.22)	50.00(0.00)
OOD link prediction	**Inductive $N^{\text{te}} = 3840$**	GraphSAGE*	1.79(1.21)	13.70(6.71)	25.31(8.77)	16.65(3.31)	52.79(1.01)
		GCN*	12.38(1.23)	27.28(1.27)	37.45(1.43)	55.03(0.76)	77.38(0.36)
		GAT*	2.76(1.27)	7.55(3.28)	12.78(4.50)	23.83(16.31)	59.54(6.96)
		GIN*	0.00(0.00)	0.00(0.00)	0.00(0.00)	45.87(3.55)	68.92(2.78)
		gMPNN$^{\bullet\bullet}$ (fixed Ψ)	9.31(5.23)	67.42(0.02)	78.44(0.01)	**75.42(0.17)**	**87.37(0.11)**
		gMPNN$^{\bullet\bullet}$ (learn Ψ)	**57.97(0.02)**	**74.75(0.07)**	**80.00(0.11)**	72.04(0.20)	84.57(0.14)
		Random	1.21(3.50)	3.39(7.72)	5.71(11.13)	0.00(0.00)	50.00(0.00)

Table 4: Test performance over 50 runs of node and pairwise gMPNNs for in-distribution (large) and OOD (small) link prediction over SBM graphs. Methods marked with $*$ indicate best result out of distinct configurations detailed in the Appendix.

			Training graph size $\boldsymbol{N^{\text{tr}} = 10^4}$				
Tasks		Model	Hit@10(%)	Hit@50(%)	Hit@100(%)	mcc.(%)	balanced acc.(%)
In-distribution link prediction	Transductive	GraphSAGE*	75.35(38.50)	75.41(38.53)	75.46(38.55)	70.81(43.47)	85.93(20.62)
		GCN*	86.23(27.88)	86.48(27.85)	86.56(27.85)	82.73(32.32)	91.36(15.84)
		GAT*	59.21(43.07)	59.62(43.09)	59.79(43.12)	50.19(42.84)	75.51(21.35)
		GIN*	80.89(33.65)	81.12(33.71)	81.20(33.72)	82.46(30.01)	90.49(16.59)
		gMPNN$^{\bullet\bullet}$ (fixed Ψ)	95.74(0.12)	96.15(0.06)	96.33(0.04)	93.77(0.04)	96.79(0.02)
		gMPNN$^{\bullet\bullet}$ (learn Ψ)	**96.95(0.03)**	**96.95(0.03)**	**96.95(0.03)**	**93.76(0.06)**	**96.79(0.03)**
		Oracle	96.96(0.03)	96.96(0.03)	96.96(0.03)	93.77(0.04)	96.79(0.02)
	Inductive $N^{\text{te}} = N^{\text{tr}}$	GraphSAGE*	64.77(40.22)	65.88(39.91)	66.60(39.87)	33.19(50.16)	68.30(23.45)
		GCN*	79.67(34.82)	79.90(34.55)	80.07(34.31)	51.16(49.53)	76.23(23.72)
		GAT*	46.73(37.62)	47.12(37.64)	47.31(37.65)	19.14(39.03)	60.02(18.77)
		GIN*	59.68(41.62)	61.15(41.47)	61.69(41.37)	44.80(46.15)	71.90(22.57)
		gMPNN$^{\bullet\bullet}$ (fixed Ψ)	95.67(0.11)	96.15(0.06)	96.33(0.04)	93.77(0.05)	96.79(0.03)
		gMPNN$^{\bullet\bullet}$ (learn Ψ)	**96.94(0.04)**	**96.94(0.04)**	**96.94(0.04)**	**93.76(0.06)**	**96.78(0.03)**
		Oracle	96.95(0.04)	96.95(0.04)	96.95(0.04)	93.77(0.05)	96.79(0.03)
OOD link prediction	**Inductive $N^{\text{te}} = 10^3$**	GraphSAGE*	33.52(44.93)	33.70(44.87)	33.97(44.77)	32.72(47.00)	66.97(22.73)
		GCN*	72.28(40.06)	73.95(38.58)	74.17(38.56)	68.93(40.98)	84.54(19.69)
		GAT*	23.31(39.07)	23.32(39.07)	23.34(39.07)	24.07(39.18)	61.74(19.39)
		GIN*	1.31(1.62)	1.39(1.62)	1.42(1.62)	-0.64(5.63)	49.93(0.63)
		gMPNN$^{\bullet\bullet}$ (fixed Ψ)	93.68(0.40)	93.72(0.41)	93.74(0.41)	93.40(0.42)	96.59(0.22)
		gMPNN$^{\bullet\bullet}$ (learn Ψ)	**96.12(0.28)**	**96.44(0.34)**	**96.57(0.36)**	**94.43(0.31)**	**97.14(0.16)**
		Oracle	96.94(0.30)	96.94(0.30)	96.94(0.30)	93.82(0.40)	96.81(0.21)

as negative train and validation edges. The embedding method gMPNN$^{\bullet}$ (resp. gMPNN$^{\bullet\bullet}$) along with link predictor $\eta^{\bullet}$ (resp. $\eta^{\bullet\bullet}$) are trained in an end-to-end manner for predicting positive and negative edges in training using cross-entropy loss. Our experiments consider three scenarios (in all scenarios we use the same number of negative test edges as positive test edges, sampled from non-edges in G^{te} with endpoints in different isomorphic blocks): (i) (In-distribution) transductive scenario where $G^{\text{te}} = G^{\text{tr}}$, where positive test edges are the 10% reserved in $E^{\text{hid-tr}}$ not used in training or validation; (ii) In-distribution inductive scenario where G^{te} is sampled from the same SBM with $N^{\text{te}} = N^{\text{tr}}$, where we also hide 10% of the edges and sample $0.1|E^{\text{hid-tr}}|$ positive test edges from $E^{\text{hid-te}}$ (for fair comparison across all scenarios); (c) OOD inductive scenario where G^{te} is sampled from the same SBM with $N^{\text{te}} = 10 \times N^{\text{tr}}$, where we also hide 10% of the edges and sample $0.1|E^{\text{hid-tr}}|$ positive test edges from $E^{\text{hid-te}}$(for fair comparison across all scenarios).

For *structural node embeddings* we consider GraphSAGE [24], GCN [28] (without positional features), GAT [70] and GIN [78] as the representatives of gMPNN$^{\bullet}$ models. Here we also add $\max$ aggregation for GAT and GraphSAGE model as proposed by Xu et al. [80] for extrapolation. The link predictor $\eta^{\bullet}$ is as feedfoward network that receives the two node embeddings as input, and has link prediction threshold $\tau = 0.5$ (see Definition 8 for details). We initialize the node features as the size-normalized degrees.

For *structural pairwise embeddings* we choose our proposed gMPNN$^{\bullet\bullet}$ method of Definition 10, since we can prove that our approach is theoretically sound in Lemma 2. We test gMPNN$^{\bullet\bullet}$ in two versions: The Φ and Ψ functions in Lemma 2 (denoted *fixed* Ψ) and a feedforward neural network for Ψ (denoted *learn* Ψ). The link predictor $\eta^{\bullet\bullet}$ is the same as $\eta^{\bullet}$ except it just takes one pairwise embedding as input, rather than two node embeddings. We initialize the pairwise features as all 1's to contain no additional information about connectivity between the pair of the nodes.

Many existing link prediction methods rely on positional node embeddings and our work focuses on permutation-equivariant MPNN GNNs. These positional node embedding link prediction methods are not equivariant (they are positional node representations) based on matrix and tensor factorization methods. Developing a theory for the effect of positional node representations in OOD link prediction is far from trivial and an entirely new paper that requires a new theory. At this point we do not even know how positional representations could be approximately counterfactually-invariant.

For all models including gMPNN$^{\bullet}$, gMPNN$^{\bullet\bullet}$, $\eta^{\bullet}$ and $\eta^{\bullet\bullet}$. The number of hidden layers was chosen between $\{2, 3\}$, and the number of hidden neurons was chosen between $\{5, 10\}$ due to the simple experimental set up. For GAT, we have 2 attention heads. Specifically. We optimized all models using Adam with learning rate chosen from $\{1\times10^{-3}, 5\times10^{-4}, 1\times10^{-4}\}$. We also choose $\eta^{\bullet}$ as taking the inner product between pair of nodes as input (as Hu et al. [25]) and the concatenated node embeddings as input. The hyperparameter search is performed by training all models with 10 different initialization seeds and selecting the configuration that achieved the highest mean accuracy on the validation data, and we mark the methods with $*$ in Tables 1 and 2 indicating the optimal configuration is being used. The training time is around 10 minutes for $1,000$ epochs.

Table 2 presents our empirical results in the new setting over 50 independent runs. The oracle predictor knows the graphon values $W(X_i^{\text{te}}, X_j^{\text{te}})$. The reason why it can not achieve 100% accuracy is because there exists rarely sampled positive edges between blocks. Our evaluation metrics include the Matthews correlation coefficient (mcc) [47], balanced accuracy, and Hits@K for $K = 10, 50, 100$ that counts the ratio of positive edges ranked at the k-th place or above against all negative edges. The results from the new table conveys the same message as Table 1 and has been discussed in Section 6.

We also include a new setting for training on larger graphs (10^4 nodes) and extrapolating to smaller graphs (10^3 nodes) in Table 4. We are able to see the structure node representations gMPNN$^{\bullet}$ are still able to perform relatively well on in-distribution inductive tasks, although the graphs are large, while still suffer from OOD performance to smaller graphs except GCN, although it is not related to the theoretical discussions of this paper. In contrast, the gMPNN$^{\bullet\bullet}$ is able to consistently offer good performance on both in-distribution and OOD tasks.

As discussed in Appendix A, Xu et al. [79] shows that GNNs can extrapolate in algorithmic-related tasks as the graph size grows, if the GNN uses max as an aggregator (rather than mean and sum we considered in this paper). Unfortunately, our Definition 3 of gMPNN$^{\bullet}$ does not allow max aggregators, in part because it is unclear how one could reach stability using the max aggregator. Fortunately, while we could not obtain theoretical results using the max aggregator, we can test it empirically. Table 2 reproduces all our empirical results using the max aggregator (on GraphSAGE and GAT, since these are the only GNNs designed for the max aggregator). Our experiments show that the max aggregator, just like the mean and sum aggregators, shows poor OOD performance as test graph sizes increase. Whether there is theoretical proof that the max aggregator is not able to perform this OOD task is left as future work.

B.4 Link prediction performance evaluation with ogbl-ddi

In what follows we introduce empirical results using the ogbl-ddi dataset, which represents a drug-drug interaction network. For the purpose of performing OOD tasks, we start by sampling 10% of the nodes (427 nodes) and its induced subgraph to be the training graph, where node features are constructed as size-normalized degrees in the training graph. Validation positive and negative edges are obtained by applying the original edge split on the induced training subgraph. Our experiments consider three scenarios: (i) (In-distribution) transductive scenario where $G^{\text{te}} = G^{\text{tr}}$, where test positive and negative edges are obtained by applying the original edge split on the induced training subgraph; (ii) In-distribution inductive scenario where G^{te} is constructed as sampling $N^{\text{te}} = N^{\text{tr}}$ nodes from the remaining ogbl-ddi graph and its induced subgraph, where the test edges are obtained by applying the original edge split on the newly induced test subgraph; (iii) OOD inductive scenario where G^{te} is the induced subgraph without the training nodes with $N^{\text{te}} = 3840$, the test edges are obtained by applying the original edge split on the newly induced test subgraph, where we further down-sample to the same amount of test edges as in (ii) for fair comparison across all scenarios.

We used the same benchmarking methods as in the SBM experiments, and add a random guesser where it is constructed as randomly-initialized GraphSAGE model with a randomly-initialized link predictor. We initialize the pairwise features as all 1's to contain no additional information about connectivity between the pair of the nodes for gMPNN$^{\bullet\bullet}$.

For *structural node embeddings* we consider GraphSAGE [24], GCN [28] (without positional features), GAT [70] and GIN [78] as the representatives of gMPNN$^{\bullet}$ models. The link predictor $\eta^{\bullet}$ is as feedfoward network that receives the two node embeddings as input, and has link prediction threshold $\tau = 0.5$ (see Definition 8 for details). We initialize the node features as the size-normalized degrees.

For *structural pairwise embeddings* we choose our proposed gMPNN$^{\bullet\bullet}$ method of Definition 10, since we can prove that our approach is theoretically sound in Lemma 2. We test gMPNN$^{\bullet\bullet}$ in two versions: The Φ and Ψ functions in Lemma 2 (denoted *fixed* Ψ) and a feedforward neural network for Ψ (denoted *learn* Ψ). The link predictor $\eta^{\bullet\bullet}$ is the same as $\eta^{\bullet}$ except it just takes one pairwise embedding as input, rather than two node embeddings. We initialize the pairwise features as all 1's to contain no additional information about connectivity between the pair of the nodes.

For all models including gMPNN$^{\bullet}$, gMPNN$^{\bullet\bullet}$, $\eta^{\bullet}$ and $\eta^{\bullet\bullet}$. The number of hidden layers was chosen between $\{2, 3\}$, and the number of hidden neurons was chosen between $\{16, 32\}$ due to the simple experimental set up. For GAT, we have 2 attention heads. Specifically. We optimized all models using Adam with learning rate chosen from $\{1\times10^{-3}, 5\times10^{-4}, 1\times10^{-4}\}$. We also choose $\eta^{\bullet}$ as taking the inner product between pair of nodes as input (as Hu et al. [25]) and the concatenated node embeddings as input. We train all the models with 200 epochs. The hyperparameter search is performed by training all models with 10 different initialization seeds and selecting the configuration that achieved the highest mean accuracy on the validation data, and we mark the methods with $*$ in Table 3 indicating the optimal configuration is being used.

Table 3 presents our empirical results on the ogbl-ddi link prediction task. All gMPNN$^{\bullet}$ methods performs worse in inductive settings than transductive settings, and suffer much worse performance in OOD transductive setting except GCNs. In contrast, the gMPNN$^{\bullet\bullet}$ is able to consistently offer good performance on both in-distribution and OOD tasks, showing that the theoretical results are not limited to SBM models.

C Defintion and notations

In this section, we follow the definitions and notations from Maskey et al. [46, Appendix A]. As in Maskey et al. [46, Appendix A], we call the metric space (χ, d), where the metric in the space χ is defined as $d : \chi \times \chi \to [0, \infty)$. The nodes of the graph are considered as sampled point from χ, the node i is identified with X_i for the graph G with nodes $X = (X_1, \ldots, X_N)$. We also represent $\boldsymbol{F}(X_i) := \mathbf{f}_i$ for $i = 1, \ldots, N$.

Next, we define various notions of degree for the pairwise node embedding.

Definition 11. *Let W defined in Definition 1, and G as the sampled graph with nodes $X = (X_1, ..., X_N)$.*

- *We define the graphon fraction of common neighbors at $x, y \in \mathcal{X}$ by*

$$c_W(x,y) = \int_{\mathcal{X}} W(x,z)W(y,z)d\mu(z), \tag{2}$$

- *Given two points x, y that need not be in X, we define the graph-graphon fraction of common neighbors of X at x, y by*

$$c_X(x,y) = \frac{1}{N}\sum_{i=1}^{N} W(x,X_i)W(y,X_i), \tag{3}$$

- *Given two points x, y that need not be in X, we define the sampled-graph fraction of common neighbors of X at x, y by*

$$c_A(x,y) = \frac{1}{N}\sum_{i=1}^{N} A(x,X_i)A(y,X_i), \tag{4}$$

  where we define $A(x, X_i) \sim \mathit{Ber}(W(x, X_i))$ and $A(y, X_i) \sim \mathit{Ber}(W(y, X_i))$ as independent random variables.

where $c_X(x, y)$ and $c_A(x, y)$ are interpreted as the graph fraction of common of neighbors of the node pair (x, y) in the graph $(x, y, X_1, ..., X_n)$.

Adapting Maskey et al. [46, Definition A.3] to the continuous integral aggregation,

Definition 12. *Let W be defined in Definition 1, for a metric-space message signal $U : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^F$, the continuous integral aggregation is defined by*

$$M_W^{\bullet} U = \int_{\mathcal{X}} W(\cdot, y)U(\cdot, y)d\mu(y).$$

Adapting Maskey et al. [46, Definition A.4] to the N-normalized sum aggregation,

Definition 13. *Let W be defined in Definition 1, $X = X_1, ..., X_n$ sample points. For a metric-space message signal $U : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^F$, we define the graph-graphon (N-normalized) sum aggregation by*

$$M_X^{\bullet} U = \frac{1}{N}\sum_{i} W(\cdot, X_i)U(\cdot, X_i),$$

and the sampled-graph (N-normalized) sum aggregation by

$$M_A^{\bullet} U = \frac{1}{N}\sum_{i} A(\cdot, X_i)U(\cdot, X_i),$$

where we define $A(x, X_i) \sim \mathit{Ber}(W(x, X_i))$ as a random variable.

Definition 14. *Let W be defined in Definition 1, $X = X_1, ..., X_n$ sample points. For a metric-space message signal $U : (\mathcal{X} \times \mathcal{X}) \times (\mathcal{X} \times \mathcal{X}) \to \mathbb{R}^F$, we define the graphon pairwise aggregation by*

$$M_W^{\bullet\bullet} U = \frac{1}{2}\int_{\mathcal{X}} (\frac{W(y,z)}{c_W(\cdot,\cdot)} U(\cdot, (x,z)) + \frac{W(x,z)}{c_W(\cdot,\cdot)} U(\cdot, (y,z)))d\mu(z),$$

and the graph-graphon pairwise aggregation by

$$M_X^{\bullet\bullet} U = \frac{1}{2N}\sum_{i=1}^{N} (\frac{W(y,X_i)}{c_X(\cdot,\cdot)} U(\cdot, (x,X_i)) + \frac{W(x,X_i)}{c_X(\cdot,\cdot)} U(\cdot, (y,X_i))),$$

and the sample-graph pairwise aggregation by

$$M_A^{\bullet\bullet} U = \frac{1}{2N}\sum_{i=1}^{N} (\frac{A(y,X_i)}{c_A(\cdot,\cdot)} U(\cdot, (x,X_i)) + \frac{A(x,X_i)}{c_A(\cdot,\cdot)} U(\cdot, (y,X_i))),$$

where we define $A(x, X_i) \sim \mathit{Ber}(W(x, X_i))$ as a random variable.

Maskey et al. [46, Definition A.7] has defined for a vector $\mathbf{z} = (z_1, \ldots, z_F) \in \mathbb{R}^F$, we define as usual

$$\|\mathbf{z}\|_\infty = \max_{1 \leq k \leq F} |z_k|.$$

For every $x, x' \in \mathcal{X}$, we say a function $f : \chi \to \mathbb{R}^F$ is Lipschitz continuous if there exists a $L_f > 0$ such that for every $x, x' \in \chi$, we have

$$\|f(x) - f(x')\|_\infty \leq L_f d(x, x').$$

Here if $\chi = \mathbb{R}^F$, $d(x, x') = \|x - x'\|_\infty$. For our theoretical results, we make the following assumptions:

Assumption 1. *(extension of Maskey et al. [46, Definition A.10]) Let (χ, d) be a metric space and $W : \chi \times \chi \to [0, \infty)$. Let Θ be a MPNN with message and update functions $\Phi^{(l)} : \mathbb{R}^{2F_l} \to \mathbb{R}^{H_l}$ and $\Psi^{(l)} : \mathbb{R}^{F_l + H_l} \to \mathbb{R}^{F_{l+1}}$, $l = 1, \ldots, T-1$.*

1. *By Definition 1 of the graphon , the graphon satisfies $\|W\|_\infty \leq 1$.*
2. *[46, Definition A.10, item 6]: There exists a constant $\mathrm{d}_{\min} > 0$ such that for every $x \in \chi$, we have $d_W(x) \geq \mathrm{d}_{\min}$.*
3. *There exists a constant d_{cmin} such that for every $x, y \in \mathcal{X}$, we have $c_W(x, y) \geq d_{cmin}$.*
4. *$M_{tr} = \max(supp(N^{tr}))$ is the largest graph in training, where N^{tr} is the distribution of graph sizes in the training data.*
5. *[46, Similar to Definition A.10, item 7 adding dependence on M_{tr}]: For every $l = 1, \ldots, T$, the message function $\Phi^{(l)}$ and update function $\Psi^{(l)}$ are Lipschitz continuous with Lipschitz constants $L_\Phi^{(l)}(M_{tr})$ and $L_\Psi^{(l)}(M_{tr})$ respectively.*

D Large real-world and random graphs have relatively few isomorphic nodes

In what follows we show that isomorphic nodes are rare both in many real-world networks and SBMs. We start with real-world graphs. MacArthur et al. [42] has computed the fraction of *non-isomorphic* nodes (denoted as the *network redundancy* $r_\mathcal{G}$ by [42]) of different types of small (< 23,000 nodes) real-world graphs. MacArthur et al. [42] shows that the majority of biological graphs are composed of mostly non-isomorphic nodes. Small technological networks (e.g., road network) tend to have significantly more isomorphic nodes.

In order to see whether these results also hold for larger graphs, we performed a similar experiment on the following datasets.

- The ogbl-ppa dataset is an undirected, unweighted graph. Nodes represent proteins from 58 different species, and edges indicate biologically meaningful associations between proteins [74], e.g., physical interactions, co-expression, homology or genomic neighborhood.
- The ogbl-ddi dataset is a homogeneous, unweighted, undirected graph, representing the drug-drug interaction network. Each node represents an FDA-approved or experimental drug [74]. Edges represent interactions between drugs and can be interpreted as a phenomenon where the joint effect of taking the two drugs together is considerably different from the expected effect in which drugs act independently of each other.
- The Slashdot graph contains friend/foe links between the users of Slashdot (where we ignore edge types).
- HepPh is a co-authorship network where if an author i co-authored a paper with author j, the graph contains a undirected edge from i to j. If the paper is co-authored by k authors this generates a completely connected (sub)graph on k nodes.
- The Github graph shows GitHub developers (nodes) who have starred at least 10 repositories and edges are mutual follower relationships between them (we make the graph undirected for our analysis).

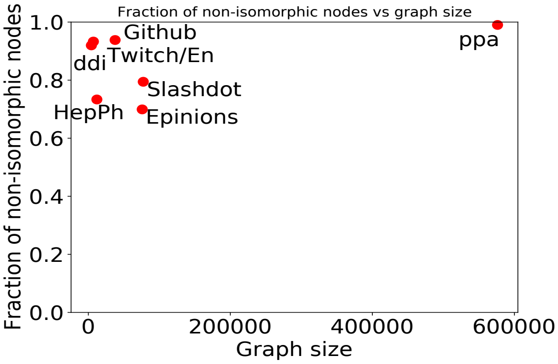

Figure 3: **Fraction of isomorphic nodes in real-world graphs:** The fraction of non-isomorphic nodes (also denoted as the *network redundancy* r_G by [42]) in real-world graphs tends to be close to 90%, except in the HepPh collaboration network and Slashdot, which contain many small disconnected components. We assume all graphs are undirected and unattributed for this analysis.

- The Twitch/En graph shows Twitch users (who stream in English) as nodes and links are mutual friendships between them.
- The Epinions graph shows users of the consumer review site Epinions.com as nodes and edges as trust relationships between users.

Figure 3 shows the fraction of *non-isomorphic* node shown against the size of the graph. This analysis considers a few datasets widely used in the neural network literature to benchmark link prediction methods such as OGB[2] ppa and ddi, where ppa is the largest dataset we were able to run the *nauty*[3] isomorphism checking algorithm without crashing. Nauty [48] is one of the most efficient graph isomorphism algorithms available, which we use to calculate a lower bound on the size of the automorphism group of our graphs. Using social networks in the SNAP[4] repository we again observe a large fraction of the nodes are non-isomorphic. Visual inspection shows that most isomorphic nodes are low-degree siblings (e.g., the most common are nodes with degree one that have the same parent). Note that our results do not contradict Ball and Geyer-Schulz [5], which shows that many real-world graphs have isomorphic nodes. Having isomorphic nodes is different than containing a large fraction of isomorphic nodes.

The results in Figure 3 show that most nodes in real-world graphs tend to be non-isomorphic for reasonably large graphs (in particular ppa). In what follows we show that random graph models generally *do not* contain isomorphic nodes with high probability.

Theoretical results on random graphs. Regarding isomorphic nodes on random graphs, we can prove the following result:

Corollary 3. *Consider a random graph $G = (V, E)$ with N given nodes so that all possible $2^{\binom{N}{2}}$ graphs should have the same probability to be chosen. Then, as $N \to \infty$ all nodes in G are non-isomorphic, regardless whether we take the nodes in G to be attributed or unattributed.*

Proof. The proof for unattributed G is a direct consequence of Erdos and Rényi [18, Theorem 2]. Adding node attributes cannot make two non-isomorphic nodes be isomorphic, which concludes our proof. □

Erdos and Rényi [18, Theorem 3] (see also Kim et al. [27, Theorem 3.1]) shows that the statement in Corollary 3 is also true for $G(N, p)$ graphs with p satisfying $(\ln N)/N \leq p \leq 1 - (\ln N)/N$. Kim et al. [27, Theorem 3.1] shows a similar result for random d-regular graph on N vertices with $3 \leq d \leq n - 4$. Luczak et al. [41] has shown similar results for preferential-attachment graphs, where in each step a new node with $m \geq 3$ edges is added. In what follows we show a similar result for SBMs.

[2] https://ogb.stanford.edu/docs/graphprop/
[3] https://pallini.di.uniroma1.it/
[4] https://snap.stanford.edu/data/index.html

Proposition 1. *Consider a random graph $G = (V, E)$ with N given nodes, generated by the SBM in Definition 6, where within-block and inter-block probabilities in $\boldsymbol{S}$ lie in the interval $(p, 1-p)$, with $(\ln N)/N \leq p \leq 1 - (\ln N)/N$. Then, as $N \to \infty$ all nodes in G are non-isomorphic, regardless whether we take the nodes in G to be attributed or unattributed.*

Proof. Let $\boldsymbol{S}$ have $r > 0$ blocks. Consider generating the G first by sampling the within-block edges. Let G_a be the induced subgraph of all nodes that belong to a single block $a \in \{1, \ldots, r\}$. By the results in Erdos and Rényi [18, Theorem 3], as $N \to \infty$, G_a has no isomorphic nodes. The above is true for all within-block edges. Now consider sampling the between-block edges of two $i, j \in V$ nodes in G. The event of i and j being isomorphic (if considering just their edges to G_a) is the same as they connecting to the same nodes in G_a (since each node on a block is non-isomorphic, if they connect to different nodes they would no longer be isomorphic). The probability of this event is at most $(1-\epsilon)^{\alpha N}$, for $\epsilon = \min(p, 1-p)$, where $\alpha > 0$ is the fraction of nodes in G_a (which is not a function of N). As $N \to \infty$, by the union bound, the probability that this will happen with any pair of edges is at most $\binom{N}{2}(1-\epsilon)^{\alpha N}$, which goes to zero.

W.l.o.g. now assume a is the block with the least number of nodes (which is also diverging as $N \to \infty$). The only alternative for i and j to be isomorphic is to do so by connecting to nodes in distinct blocks. For instance, we could imagine r copies of G_a: Even though i and j did not connect to the same nodes in the same graphs, they connected to their isomorphic equivalent nodes in different copies. But since there are only r blocks, and r does not depend on N, this event must have probability at most $(1-\epsilon)^{\alpha N/r}$. As $N \to \infty$, by the union bound, the probability $\binom{N}{2}(1-\epsilon)^{\alpha N/r}$ goes to zero. Replacing the copies of G_a with the actual sampled blocks only makes this probability smaller, since the subgraphs of the other blocks are larger and may contain different topologies than G_a (making their nodes distinct from the nodes in G_a). Finally, adding node attributes cannot make two non-isomorphic nodes be isomorphic, which concludes our proof.

□

E Proof results for Theorem 1

In what follows we provide the elements to prove Theorem 1.

First we prove the following lemma of the difference between a graph-graphon (N-normalized) sum aggregation and a sampled-graph (N-normalized) sum aggregation in Definition 13. Using the same assumptions as Maskey et al. [45, Lemma B.3]:

Lemma 3. *Let (χ, d, μ) be a metric-measure space and W be a graphon s.t. Assumptions 1.1-2. are satisfied. Let $\Phi : \mathbb{R}^{2F} \to \mathbb{R}^H$ be Lipschitz continuous with Lipschitz constant $L_\Phi(M_{tr})$, with M_{tr} as in Assumption 1 item 4. Consider a metric-space signal $f : \chi \to \mathbb{R}^F$ with $\|f\|_\infty < \infty$. Suppose that $X_1, \ldots, X_N$ are drawn i.i.d. from μ on χ, and let $p \in (0, 1/H)$. Let $x \in \chi$, and define the random variable*

$$T_x = \frac{1}{N}\sum_{i=1}^{N} A(x, X_i)\Phi\big(f(x), f(X_i)\big) - \frac{1}{N}\sum_{i=1}^{N} W(x, X_i)\Phi\big(f(x), f(X_i)\big)$$

on the sample space $\mathcal{X}^N \times [0,1]^N$. Then, with probability at least $1 - Hp$, we have

$$\|T_x\|_\infty \leq \sqrt{2}\frac{(L_\Phi(M_{tr})\|f\|_\infty + \|\Phi(0,0)\|_\infty)\sqrt{\log 2/p}}{\sqrt{N}}. \tag{5}$$

Proof. The proof of the bound is the same as the proof in Maskey et al. [45, Lemma B.3], even though T_x is a different quantity than the quantity used in Maskey et al. [45, Lemma B.3]. □

Combining Maskey et al. [45, Lemma B.3] and Lemma 3, we can use the triangle inequality to prove the following lemma about concentration of error between a sampled-graph (N-normalized) sum aggregation in Definition 13 and the continuous integral aggregation in Definition 12, which is used in Definitions 3 and 5. Using the same assumption as Maskey et al. [45, Lemma B.4]:

Lemma 4. *Let (χ, d, μ) be a metric-measure space and W be a graphon s.t. Assumptions 1.1-2. are satisfied. Let $\Phi : \mathbb{R}^{2F} \to \mathbb{R}^H$ be Lipschitz continuous with Lipschitz constant $L_\Phi(M_{tr})$. Consider a metric-space signal $f : \chi \to \mathbb{R}^F$ with $\|f\|_\infty < \infty$. Suppose that $X_1, \dots, X_N$ are drawn i.i.d. from μ on χ, and let $p \in (0, 1/(2H))$. Let $x \in \chi$, and define the random variable*

$$R_x = \frac{1}{N} \sum_{i=1}^{N} A(x, X_i) \Phi\big(f(x), f(X_i)\big) - \int_\chi W(x, y) \Phi\big(f(x), f(y)\big) d\mu(y)$$

on the sample space $\chi^N \times [0,1]^N$. Then, with probability at least $1 - 2Hp$, we have

$$\|R_x\|_\infty \leq 2\sqrt{2} \frac{(L_\Phi(M_{tr})\|f\|_\infty + \|\Phi(0,0)\|_\infty)\sqrt{\log 2/p}}{\sqrt{N}}. \tag{6}$$

Proof. Use the triangle inequality, the results from Maskey et al. [45, Lemma B.3] and Lemma 3. Define $Y_x = \frac{1}{N} \sum_{i=1}^{N} W(x, X_i) \Phi\big(f(x), f(X_i)\big) - \int_\chi W(x, y) \Phi\big(f(x), f(y)\big) d\mu(y)$.

$$\|R_x\|_\infty = \|T_x + Y_x\|_\infty \leq \|T_x\|_\infty + \|Y_x\|_\infty.$$

From Maskey et al. [45, Lemma B.3] and Lemma 3, $\|T_x\|_\infty \leq \sqrt{2} \frac{(L_\Phi(M_{\mathrm{tr}})\|f\|_\infty + \|\Phi(0,0)\|_\infty)\sqrt{\log 2/p}}{\sqrt{N}}$ w.p. $1 - Hp$ and $\|Y_x\|_\infty \leq \sqrt{2} \frac{(L_\Phi(M_{\mathrm{tr}})\|f\|_\infty + \|\Phi(0,0)\|_\infty)\sqrt{\log 2/p}}{\sqrt{N}}$ w.p. $1 - Hp$. We have with probability at least $1 - 2Hp$ using the union bound of the two events,

$$\|R_x\|_\infty \leq 2\sqrt{2} \frac{(L_\Phi(M_{\mathrm{tr}})\|f\|_\infty + \|\Phi(0,0)\|_\infty)\sqrt{\log 2/p}}{\sqrt{N}}.$$

□

Based on Lemma 4, we can prove the following corollary about the maximum concentration error between sampled-graph (N-normalized) sum aggregation ($M_A^\bullet$) and continuous integral aggregation ($M_W^\bullet$) for all the nodes in the sampled graph G. Using the same overall framing as [45, Lemma B.3]:

Corollary 4. *Consider (χ, d, μ) a metric-measure space and graphon W satisfying items 1 and 2 of Assumption 1. Let $\Phi : \mathbb{R}^{2F} \to \mathbb{R}^H$ be Lipschitz continuous with Lipschitz constant $L_\Phi(M_{tr})$, and a metric-space signal $f : \chi \to \mathbb{R}^F$ with $\|f\|_\infty < \infty$. Define $X_1, \dots, X_N$ as drawn i.i.d. from μ on χ, and then edges $A_{i,j} \sim Ber(W(X_i, X_j))$ i.i.d sampled. Let $p \in (0, 1/2H)$, and define the random variable*

$$R_{X_i} = \frac{1}{N} \sum_{j=1}^{N} A(X_i, X_j) \Phi\big(f(X_i), f(X_j)\big) - \int_\chi W(X_i, y) \Phi\big(f(X_i), f(y)\big) d\mu(y)$$

on the sample space $\chi^N \times [0,1]^N$. Then, with probability at least $1 - 2Hp$, we have

$$\begin{aligned}\max_{i=1,\dots,N} \|(M_A^\bullet - M_W^\bullet)\big(\Phi(f,f)\big)(X_i)\|_\infty &= \max_{i=1,\dots,N} \|R_{X_i}\|_\infty \\ &\leq 2\sqrt{2} \frac{(L_\Phi(M_{tr})\|f\|_\infty + \|\Phi(0,0)\|_\infty)\sqrt{\log(2N/p)}}{\sqrt{N}}.\end{aligned} \tag{7}$$

Proof. Using the result from Lemma 3 we have with probability $1 - \frac{Hp}{N}$,

$$\begin{aligned}&\|\frac{1}{N} \sum_{i=1}^{N} A(x, X_i) \Phi\big(f(x), f(X_i)\big) - \frac{1}{N} \sum_{i=1}^{N} W(x, X_i) \Phi\big(f(x), f(X_i)\big)\|_\infty \\ &\leq \sqrt{2} \frac{(L_\Phi(M_{\mathrm{tr}})\|f\|_\infty + \|\Phi(0,0)\|_\infty)\sqrt{\log(2N/p)}}{\sqrt{N}}.\end{aligned}$$

Using the union bound of the N events that the above equations happens for $x = X_1, ..., X_N$, with probability at least $1 - Hp$, we have

$$\max_{i=1,...,N} \|\frac{1}{N}\sum_{j=1}^{N} A(X_i, X_j)\Phi\big(f(X_i), f(X_j)\big) - \frac{1}{N}\sum_{j=1}^{N} W(X_i, X_j)\Phi\big(f(X_i), f(X_j)\big)\|_\infty$$
$$\leq \sqrt{2}\frac{(L_\Phi(M_{\text{tr}})\|f\|_\infty + \|\Phi(0,0)\|_\infty)\sqrt{\log(2N/p)}}{\sqrt{N}}.$$

The same logic can be applied to $Y_{X_i}, \forall i \in \{1, ..., N\}$. Thus, using the triangle inequality, and the union bound of the two events, we have with probability at least $1 - 2Hp$,

$$\max_{i=1,...,N} \|R_{X_i}\|_\infty \leq 2\sqrt{2}\frac{(L_\Phi(M_{\text{tr}})\|f\|_\infty + \|\Phi(0,0)\|_\infty)\sqrt{\log(2N/p)}}{\sqrt{N}}.$$

□

Now the layer-wise error between a cMPNN$^\bullet$ and gMPNN$^\bullet$ can be bounded as follows:

Corollary 5. *Consider (χ, d, μ) a metric-measure space and graphon W consistent with items 1 and 2 of Assumption 1. Let $\Phi : \mathbb{R}^{2F} \to \mathbb{R}^H$ and $\Psi : \mathbb{R}^{F+H} \to \mathbb{R}^{F'}$ be Lipschitz continuous with Lipschitz constants $L_\Phi(M_{tr})$ and $L_\Psi(M_{tr})$. Consider a metric-space signal $f : \chi \to \mathbb{R}^F$ with $\|f\|_\infty < \infty$. Let $p \in (0, \frac{1}{2(H+1)})$. Suppose that $X_1, \ldots, X_N$ are drawn i.i.d. from μ in χ, and then edges $A_{i,j} \sim$ Ber$(W(X_i, X_j))$ i.i.d sampled. Then with probability at least $1 - 2Hp$,*

$$\max_{i=1,...,N} \|\Psi\Big(f(\cdot), M_A^\bullet\big(\Phi(f,f)\big)(X_i)\Big) - \Psi\Big(f(\cdot), M_W^\bullet\big(\Phi(f,f)\big)(X_i)\Big)\|_\infty$$
$$\leq L_\Psi(M_{tr})\Big(2\sqrt{2}\frac{(L_\Phi(M_{tr})\|f\|_\infty + \|\Phi(0,0)\|_\infty)\sqrt{\log(2N/p)}}{\sqrt{N}}\Big). \tag{8}$$

Proof. The proof is the same as Maskey et al. [45, Lemma B.6]. The different result comes from the different bound between Corollary 4 and Maskey et al. [45, Lemma B.5]. □

E.1 Proof of Theorem 1

Following [45, Appendix B.2], they first bound the layer-wise error as Corollary 5, and derive the final bound through a recurrence relation. The only difference is on the layer-wise bound Corollary 6 and Maskey et al. [45, Corollary B.6]. We will omit the middle parts. Hence, finally, we can prove Theorem 1 by slightly adpating the proof in Maskey et al. [45, Theorem B.14] to our setting.

Theorem 1 (OOD convergence without in-distribution convergence)**.** *For a random graph model (W, f) satisfying Definition 1, let N^{tr} be a random variable defining the distribution of graph sizes in training. Define the test distribution $(G^{te}, \boldsymbol{F}^{te}) \sim (W, f)$ through the causal graph in Figure 1 as an interventional change to obtain larger test graph sizes where $\min(supp(N^{te})) \gg M_{tr} = \max(supp(N^{tr}))$ (which means any test graph is much larger than the largest possible training graph). Let $\Theta = ((\Phi^{(l)})_{l=1}^T, (\Psi^{(l)})_{l=1}^T)$ be a MPNN as in Definition 2 with T layers such that $\Phi^{(l)} : \mathbb{R}^{2F_{l-1}} \to \mathbb{R}^{H_{l-1}}$ and $\Psi^{(l)} : \mathbb{R}^{F_{l-1}+H_{l-1}} \to \mathbb{R}^{F_l}$ are learned from the training distribution and are Lipschitz continuous with Lipschitz constants $L_\Phi^{(l)}(M_{tr})$ and $L_\Psi^{(l)}(M_{tr})$ that depend on M_{tr}. Let gMPNN$^\bullet$ $\Theta_A^{\bullet(T)}$ and cMPNN$^\bullet$ $\Theta_W^{\bullet(T)}$ be as in Definitions 3 and 5. Let $X_1^{te}, ..., X_{N^{te}}^{te}$ and $\boldsymbol{A}^{te}$ be as in Definition 1. Let $p \in (0, \frac{1}{\sum_{l=1}^T 2(H_l+1)})$. Then, if*

$$\frac{\sqrt{N^{te}}}{\sqrt{\log(2N^{te}/p)}} \geq \frac{4\sqrt{2}}{d_{min}}, \tag{1}$$

we have with probability at least $1 - \sum_{l=1}^T 2(H_l + 1)p$,

$$\delta_{A\text{-}W}^\bullet := \max_{i=1,...,N^{te}} \|\Theta_{\boldsymbol{A}^{te}}^{\bullet(T)}(\boldsymbol{F}^{te})_i - \Theta_W^{\bullet(T)}(f)(X_i^{te})\|_\infty \leq (C_1 + C_2\|f\|_\infty)\frac{\sqrt{\log(2N^{te}/p)}}{\sqrt{N^{te}}},$$

where the constants C_1 and C_2 are defined in the Appendix and depend on $\{L_\Phi^{(l)}(M_{tr}), L_\Psi^{(l)}(M_{tr})\}_{l=1}^T$ and the distribution of $(G^{tr}, \boldsymbol{F}^{tr})$.

Proof. In this case, $\|\Phi^{(l)}(0,0)\|_\infty$, $\|\Psi^{(l)}(0,0)\|_\infty$ can be determined by $(G^{\text{tr}}, \boldsymbol{F}^{\text{tr}})$, N^{tr} if the MPNN Θ has been trained on the training graph $(G^{\text{tr}}, \boldsymbol{F}^{\text{tr}})$.

Following the procedure of Maskey et al. [45, Appendix B.2] with Corollary 5, we can derive similarly, with probability at least $1 - \sum_{l=1}^{T}(2H_l + 1)p$,

$$\begin{aligned}\delta^{\bullet}_{\text{A-W}} \leq &\sum_{l=1}^{T} L_{\Psi}^{(l)}(M_{\text{tr}})\Big(2\sqrt{2}\frac{(L_{\Phi}(M_{\text{tr}})^{(l)}\|f^{\bullet(l)}\|_\infty + \|\Phi^{(l)}(0,0)\|_\infty)\sqrt{\log(2N^{\text{te}}/p)}}{\sqrt{N^{\text{te}}}}\Big) \\ &\prod_{l'=l+1}^{T}((L_{\Psi}^{(l')}(M_{\text{tr}}))^2 + 2(L_{\Phi}^{(l')}(M_{\text{tr}}))^2(L_{\Psi}^{(l')}(M_{\text{tr}}))^2),\end{aligned} \tag{9}$$

Using the same proof in Maskey et al. [45, Lemma B.9], we can derive

$$||f^{\bullet(l)}||_\infty \leq {B_1}^{(l)} + {B_2}^{(l)}||f||_\infty,$$

where ${B_1}^{(l)}$, ${B_2}^{(l)}$ are independent of f, and

$${B_1}^{(l)} = \sum_{k=1}^{l}\big(L_{\Psi}^{(k)}(M_{\text{tr}})\|\Phi^{(k)}(0,0)\|_\infty + \|\Psi^{(k)}(0,0)\|_\infty\big)\prod_{l'=k+1}^{l} L_{\Psi}^{(l')}(M_{\text{tr}})\big(1 + L_{\Phi}^{(l')}(M_{\text{tr}})\big) \tag{10}$$

and

$${B_2}^{(l)} = \prod_{k=1}^{l} L_{\Psi}^{(k)}(M_{\text{tr}})\left(1 + L_{\Phi}^{(k)}(M_{\text{tr}})\right). \tag{11}$$

Now we can decompose the summation in Equation (9). First, we defince C_1 as

$$\begin{aligned}C_1 = &\sum_{l=1}^{T} L_{\Psi}^{(l)}(M_{\text{tr}})\Big(2\sqrt{2}(L_{\Phi}^{(l)}(M_{\text{tr}})\ {B_1}^{(l)} + \|\Phi^{(l+1)}(0,0)\|_\infty)\Big) \\ &\times \prod_{l'=l+1}^{T}((L_{\Psi}^{(l')}(M_{\text{tr}}))^2 + 2(L_{\Phi}^{(l')}(M_{\text{tr}}))^2(L_{\Psi}^{(l')}(M_{\text{tr}}))^2),\end{aligned} \tag{12}$$

Then we can define C_2 as

$$C_2 = \sum_{l=1}^{T} L_{\Psi}^{(l)}(M_{\text{tr}})\Big(2\sqrt{2}L_{\Phi}^{(l)}(M_{\text{tr}}){B_2}^{(l)}\Big)\prod_{l'=l+1}^{T}((L_{\Psi}^{(l')}(M_{\text{tr}}))^2 + 2(L_{\Phi}^{(l')}(M_{\text{tr}}))^2(L_{\Psi}^{(l')}(M_{\text{tr}}))^2), \tag{13}$$

It is clear to see we can rewrite Equation (9) as

$$\delta^{\bullet}_{\text{A-W}} \leq (C_1 + C_2\|f\|_\infty)\frac{\sqrt{\log(2N^{\text{te}}/p)}}{\sqrt{N^{\text{te}}}}. \tag{14}$$

Thus C_1 and C_2 depends on $\{L_{\Phi}^{(l)}(M_{\text{tr}})\}_{l=1}^{T}$ and $\{L_{\Psi}^{(l)}(M_{\text{tr}})\}_{l=1}^{T}$. □

F Proof of theoretical results for hardness of link prediction

In this section, we prove the results for $\Theta_A^{\bullet(T)}$ and $\Theta_W^{\bullet(T)}$ based on Theorem 1. Now we can prove Corollary 1.

Corollary 1. *Let* $\Theta = ((\Phi^{(l)})_{l=1}^{T}, (\Psi^{(l)})_{l=1}^{T}), \Theta_A^{\bullet(T)}, \Theta_W^{\bullet(T)}, p, (W, f), (G^{tr}, \boldsymbol{F}^{tr}), (G^{te}, \boldsymbol{F}^{te}), N^{tr}$, N^{te}, A^{te}, *and* $X_1^{te}, ..., X_{N^{te}}^{te}$ *be as in Theorem 1. If there exists* $i, j \in V^{te}, i \neq j$, *s.t.* $\Theta_W^{\bullet(T)}(X_i) = \Theta_W^{\bullet(T)}(X_j)$ *and Equation* (1) *is satisfied, then, with* C_1 *and* C_2 *as in Theorem 1, we have that with probability at least* $1 - \sum_{l=1}^{T} 2(H_l + 1)p$,

$$\|\Theta_{\boldsymbol{A}^{te}}^{\bullet(T)}(\boldsymbol{F}^{te})_i - \Theta_{\boldsymbol{A}^{te}}^{\bullet(T)}(\boldsymbol{F}^{te})_j\|_\infty \leq (C_1 + C_2\|f\|_\infty)\frac{2\sqrt{\log(2N^{te}/p)}}{\sqrt{N^{te}}}.$$

Proof. The proof follows Theorem 1 by using the triangle inequality.

From Theorem 1, we know with probability at least $1 - 2\sum_{l=1}^{T}(H_l + 1)p$, $\|\Theta_{A^{\text{te}}}^{\bullet(T)}(\boldsymbol{F}^{\text{te}})_i - \Theta_W^{\bullet(T)}(f)(X_i^{\text{te}})\|_\infty \leq (C_1 + C_2\|f\|_\infty)\frac{\sqrt{\log 2N^{\text{te}}/p}}{\sqrt{N^{\text{te}}}}$ and $\|\Theta_{A^{\text{te}}}^{\bullet(T)}(\boldsymbol{F}^{\text{te}})_j - \Theta_W^{\bullet(T)}(f)(X_j^{\text{te}})\|_\infty \leq (C_1 + C_2\|f\|_\infty)\frac{\sqrt{\log 2N^{\text{te}}/p}}{\sqrt{N^{\text{te}}}}$.

Then

$$\begin{aligned}
&\|\Theta_{A^{\text{te}}}^{\bullet(T)}(\boldsymbol{F}^{\text{te}})_i - \Theta_{A^{\text{te}}}^{\bullet(T)}(\boldsymbol{F}^{\text{te}})_j\|_\infty \\
&\leq \|\Theta_{A^{\text{te}}}^{\bullet(T)}(\boldsymbol{F}^{\text{te}})_i - \Theta_W^{\bullet(T)}(f)(X_i^{\text{te}})\|_\infty + \|\Theta_W^{\bullet(T)}(f)(X_i^{\text{te}}) - \Theta_{A^{\text{te}}}^{\bullet(T)}(\boldsymbol{F}^{\text{te}})_j\|_\infty \\
&= \|\Theta_{A^{\text{te}}}^{\bullet(T)}(\boldsymbol{F}^{\text{te}})_i - \Theta_W^{\bullet(T)}(f)(X_i^{\text{te}})\|_\infty + \|\Theta_W^{\bullet(T)}(f)(X_j^{\text{te}}) - \Theta_{A^{\text{te}}}^{\bullet(T)}(\boldsymbol{F}^{\text{te}})_j\|_\infty \\
&\leq (C_1 + C_2\|f\|_\infty)\frac{2\sqrt{\log(2N^{\text{te}}/p)}}{\sqrt{N^{\text{te}}}}.
\end{aligned}$$

The first inequality holds by traingle inequality, and the second equation holds since $\Theta_W^{\bullet(T)}(f)(X_i^{\text{te}}) = \Theta_W^{\bullet(T)}(f)(X_j^{\text{te}})$. □

Then we are able to prove Lemma 1 by induction. By our Definition 7, we can also claim $t_k - t_{k-1} = t_{\pi(k)} - t_{\pi(k)-1}, \forall k \in \{1, ..., r\}$.

Lemma 1. *Let $\Theta = ((\Phi^{(l)})_{l=1}^T, (\Psi^{(l)})_{l=1}^T)$ be a MPNN as in Definition 2, and $\Theta_W^{\bullet(T)}$ as in Definition 5. For the SBM model (W, f) in Definition 6 with N^{te} nodes $X_1, \ldots, X_{N^{te}}$. If there exists $i, j \in V^{te}$ such that X_i^{te}, X_j^{te} are nodes that belong to isomorphic SBM blocks (Definition 7), then $\Theta_W^{\bullet(T)}(f)(X_i^{te}) = \Theta_W^{\bullet(T)}(f)(X_j^{te})$.*

Proof. We prove the lemma by induction.

We assume in layer l, $f^{\bullet(l)}(X_i^{\text{te}}) = f^{\bullet(l)}(X_j^{\text{te}}), 1 \leq l \leq T-1$, $f^{\bullet(l)}$ outputs the same value within each block $\boldsymbol{B}^{(l)}$, and $\boldsymbol{B}^{(l)} = \pi \circ \boldsymbol{B}^{(l)}$. By Definitions 6 and 7, we know the assumption holds for $l = 1$. First,

$$f^{\bullet(l+1)}(X_i^{\text{te}}) = \Psi^{(l+1)}\Big(f^{\bullet(l)}(X_i^{\text{te}}), M_W^{\bullet}\big(\Phi^{(l+1)}(f^{\bullet(l)}, f^{\bullet(l)})\big)(X_i^{\text{te}})\Big).$$

Since $f^{\bullet(l)}(X_i^{\text{te}}) = f^{\bullet(l)}(X_j^{\text{te}})$, we only need to show $M_W^{\bullet}\big(\Phi^{(l+1)}(f^{\bullet(l)}, f^{\bullet(l)})\big)(X_i^{\text{te}}) = M_W^{\bullet}\big(\Phi^{(l+1)}(f^{\bullet(l)}, f^{\bullet(l)})\big)(X_j^{\text{te}})$.

$$\begin{aligned}
&M_W^{\bullet}\big(\Phi^{(l+1)}(f^{\bullet(l)}, f^{\bullet(l)})\big)(X_i^{\text{te}}) \\
&= \int_{[0,1]} W(X_i^{\text{te}}, y)\Phi^{(l+1)}(f^{\bullet(l)}(X_i^{\text{te}}), f^{\bullet(l)}(y))dy \\
&= \sum_{k=1}^{r} \int_{[t_{k-1}, t_k)} W(X_i^{\text{te}}, y)\Phi^{(l+1)}(f^{\bullet(l)}(X_i^{\text{te}}), f^{\bullet(l)}(y))dy \\
&= \sum_{k=1}^{r} \Phi^{(l+1)}(\boldsymbol{B}_a^{(l)}, \boldsymbol{B}_k^{(l)}) \int_{[t_{k-1}, t_k)} W(X_i^{\text{te}}, y)dy \qquad (15) \\
&= \sum_{k=1}^{r} \Phi^{(l+1)}(\boldsymbol{B}_a^{(l)}, \boldsymbol{B}_k^{(l)})(t_k - t_{k-1})\boldsymbol{S}_{i,k} \\
&= \sum_{k=1}^{r} \Phi^{(l+1)}(\boldsymbol{B}_a^{(l)}, \boldsymbol{B}_k^{(l)})(t_k - t_{k-1})\boldsymbol{S}_{\pi(i),\pi(k)}
\end{aligned}$$

$$\begin{aligned}
&= \sum_{k=1}^{r} \Phi^{(l+1)}(\boldsymbol{B}^{(l)}_{\pi(a)}, \boldsymbol{B}^{(l)}_{\pi(k)})(t_{\pi(k)} - t_{\pi(k)-1})\boldsymbol{S}_{\pi(i),\pi(k)} \\
&= \sum_{k=1}^{r} \Phi^{(l+1)}(\boldsymbol{B}^{(l)}_{b}, \boldsymbol{B}^{(l)}_{\pi(k)})(t_{\pi(k)} - t_{\pi(k)-1})\boldsymbol{S}_{j,\pi(k)} \\
&= \sum_{k=1}^{r} \Phi^{(l+1)}(\boldsymbol{B}^{(l)}_{b}, \boldsymbol{B}^{(l)}_{k})(t_{k} - t_{k-1})\boldsymbol{S}_{j,k} \\
&= \sum_{k=1}^{r} \Phi^{(l+1)}(\boldsymbol{B}^{(l)}_{b}, \boldsymbol{B}^{(l)}_{k}) \int_{[t_{k-1},t_k)} W(X^{\text{te}}_j, y)dy \\
&= \int_{[0,1]} W(X^{\text{te}}_j, y)\Phi^{(l+1)}(f^{\bullet(l)}(X^{\text{te}}_j), f^{\bullet(l)}(y))dy \\
&= M^{\bullet}_{W}\big(\Phi^{(l+1)}(f^{\bullet(l)}, f^{\bullet(l)})\big)(X^{\text{te}}_j)
\end{aligned} \tag{16}$$

Here we use the fact that $f^{\bullet(l)}$ $f^{\bullet(l)}$ outputs the same value within each block, and $\boldsymbol{B}^{(l)}_k = \boldsymbol{B}^{(l)}_{\pi(k)}, \forall k \in \{1, ..., r\}$.

We have shown $f^{\bullet(l+1)}(X^{\text{te}}_i) = f^{\bullet(l+1)}(X^{\text{te}}_j)$. And this proof applies for all $X^{\text{te}}_i \in [t_{a-1}, t_a)$ (in block a), and the same conclusion holds. So $f^{\bullet(l+1)}$ still outputs the same value within each block. Furthermore, $\boldsymbol{B}^{(l+1)}_a = \boldsymbol{B}^{(l+1)}_{\pi(a)}$ using the same proof technique. And this implies $\pi \circ \boldsymbol{B}^{(l+1)} = \boldsymbol{B}^{(l+1)}$.

Thus, $\Theta^{\bullet(T)}_W(X^{\text{te}}_i) = f^{\bullet(T)}(X^{\text{te}}_i) = f^{\bullet(T)}(X^{\text{te}}_j) = \Theta^{\bullet(T)}_W(X^{\text{te}}_j)$. □

Then we are ready to prove Corollary 2 by applying Corollary 1.

Corollary 2. *Let $\Theta = ((\Phi^{(l)})^T_{l=1}, (\Psi^{(l)})^T_{l=1})$ be the MPNN with T layers and $\Theta^{\bullet(T)}_A, \Theta^{\bullet(T)}_W$ as in Theorem 1. Let $\eta^\bullet : \mathbb{R}^{F_T} \times \mathbb{R}^{F_T} \to [0,1]$ be as in Definition 8. Consider the SBM (W, f) in Definition 6 with isomorphic blocks (Definition 7). Let $(G^{tr}, \boldsymbol{F}^{tr}) \sim (W, f)$ and $(G^{te}, \boldsymbol{F}^{te}) \sim (W, f)$ be the training and test graphs with N^{tr} and N^{te} nodes, respectively. Consider any two test nodes $i, j \in \{1, ..., N^{te}\}$, $i \neq j$, for which we can make a link prediction decision with $\eta^\bullet$ (i.e., $\eta^\bullet(\Theta^{\bullet(T)}_{\boldsymbol{A}^{te}}(\boldsymbol{F}^{te})_i, \Theta^{\bullet(T)}_{\boldsymbol{A}}(\boldsymbol{F}^{te})_j) \neq \tau$). Let G^{te} be large enough to satisfy both Equation* (1) *and*

$$\frac{\sqrt{N^{te}}}{\sqrt{\log(2N^{te}/p)}} > \frac{2(C_1 + C_2\|f\|_\infty)}{|\eta^\bullet(\Theta^{\bullet(T)}_{\boldsymbol{A}^{te}}(\boldsymbol{F}^{te})_i, \Theta^{\bullet(T)}_{\boldsymbol{A}^{te}}(\boldsymbol{F}^{te})_j) - \tau|/L^\bullet_\eta(M_{tr})},$$

where p, C_1, and C_2 are as given in Corollary 1. Then, if i and j belong to isomorphic blocks (i.e., $\Theta^{\bullet(T)}_W(f)(X^{te}_i) = \Theta^{\bullet(T)}_W(f)(X^{te}_j)$), with probability at least $1 - \sum_{l=1}^T 2(H_l + 1)p$ the link prediction method in Definition 8 will make the same link prediction regardless of the SBM probability matrix $\boldsymbol{S}$ (Definition 6) and whether i and j are in the same block or distinct isomorphic blocks.

Proof. To prove the corollary, we assume we have two nodes j and j', such that i and j are in the same block while i and j' are in distinct isomorphic blocks. In the proof, we will show that the link prediction between i and j and the prediction between i and j' will be the same.

First, from Corollary 1, since nodes j and j' are in distinct isomorphic SBM blocks, when Equation (1) is satisfied, we have with probability at least $1 - 2\sum_{l=1}^T (H_l + 1)p$

$$\|\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_j - \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_{j'}\|_\infty \leq (C_1 + C_2\|f\|_\infty)\frac{2\sqrt{\log 2N^{\text{te}}/p}}{\sqrt{N^{\text{te}}}}.$$

Then when the requirement for N^{te} is satisfied,

$$\begin{aligned}
&\|\eta^{\bullet}(\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_i, \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_j) - \eta^{\bullet}(\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_i, \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_{j'})\|_\infty \\
&\leq L^{\bullet}_{\eta}(M_{\text{tr}})\|\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_j - \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_{j'}\|_\infty \leq L^{\bullet}_{\eta}(M_{\text{tr}})(C_1 + C_2\|f\|_\infty)\frac{2\sqrt{\log 2N^{\text{te}}/p}}{\sqrt{N^{\text{te}}}} \\
&< \|\eta^{\bullet}(\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_i, \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_j) - \tau\|_\infty
\end{aligned}$$

If $\eta^{\bullet}(\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_i, \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_j) > \tau$, then

$$\begin{aligned}
&\eta^{\bullet}(\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_i, \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_{j'}) \\
&\geq \eta^{\bullet}(\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_i, \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_j) - |\eta^{\bullet}(\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_i, \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_j) - \eta^{\bullet}(\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_i, \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_{j'})| \\
&> \eta^{\bullet}(\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_i, \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_j) - |\eta^{\bullet}(\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_i, \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_j) - \tau| \\
&= \eta^{\bullet}(\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_i, \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_j) - (\eta^{\bullet}(\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_i, \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_j) - \tau) = \tau
\end{aligned}$$

If $\eta^{\bullet}(\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_i, \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_j) < \tau$, then

$$\begin{aligned}
&\eta^{\bullet}(\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_i, \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_{j'}) \\
&\leq \eta^{\bullet}(\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_i, \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_j) + |\eta^{\bullet}(\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_i, \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_j) - \eta^{\bullet}(\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_i, \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_{j'})| \\
&< \eta^{\bullet}(\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_i, \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_j) + |\eta^{\bullet}(\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_i, \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_j) - \tau| \\
&= \eta^{\bullet}(\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_i, \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_j) - (\eta^{\bullet}(\Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_i, \Theta^{\bullet(T)}_{A^{\text{te}}}(\boldsymbol{F}^{\text{te}})_j) - \tau) = \tau
\end{aligned}$$

So whether i and j are in the same block, or in distinct isomorphic SBM blocks, their prediction will be the same (both links have predictions larger than τ or less). □

G Proof for pairwise gMPNN$^{\bullet\bullet}$ and cMPNN$^{\bullet\bullet}$

First we prove Lemma 2 showing $W(x, y)$ is a stationary point in cMPNN$^{\bullet\bullet}$.

Lemma 2. *If $\Phi(x, y) = y$ and $\Psi(x, y) = x/y$, then $f^{\bullet\bullet(t)}(x, y) = W(x, y)$, $\forall x, y \in \mathcal{X}$ is a stationary point in the cMPNN$^{\bullet\bullet}$, i.e. if $f^{\bullet\bullet(t-1)}(x, y) = W(x, y)$, then $f^{\bullet\bullet(t)}(x, y) = W(x, y)$, $\forall x, y \in \mathcal{X}$.*

Proof. If $f^{\bullet\bullet(t-1)}(x, y) = W(x, y)$, then

$$\begin{aligned}
M^{\bullet\bullet}_W(\Phi^{(t)}(f^{(t-1)}))(x, y) &= \frac{1}{2}\int_{\mathcal{X}}(\frac{W(y, z)}{c_W(x, y)}\Phi^{(t)}(f^{\bullet\bullet(t-1)}(x, y), f^{\bullet\bullet(t-1)}(x, z)) \\
&+ \frac{W(x, z)}{c_W(x, y)}\Phi^{(t)}(f^{\bullet\bullet(t-1)}(x, y), f^{\bullet\bullet(t-1)}(y, z)))d\mu(z) \\
&= \frac{1}{2}\int_{\mathcal{X}}(\frac{W(y, z)}{c_W(x, y)}W(x, z) + \frac{W(x, z)}{c_W(x, y)}W(y, z))d\mu(z) \\
&= \frac{1}{c_W(x, y)}\int_{\mathcal{X}}W(x, z)W(y, z)d\mu(z) \\
&= \frac{c_W(x, y)}{c_W(x, y)} = 1
\end{aligned}$$

Thus $f^{\bullet\bullet(t)}(x, y) = \Psi^{(t)}(f^{\bullet\bullet(t-1)}(x, y), M^{\bullet\bullet}_W(\Phi^{(t)}(f^{\bullet\bullet(t-1)}, f^{\bullet\bullet(t-1)}))(x, y)) = \frac{W(x,y)}{1} = W(x, y)$.

We finish proving $W(x, y)$ is a stanionary point in cMPNN$^{\bullet\bullet}$. There are infinity choices of Φ and Ψ such that $W(x, y)$ is a stanionary point. □

Then we aim to prove Theorem 2, and the prove procedure should be very similar as Theorem 1.

G.1 Preparation

Following Maskey et al. [45, Lemma B.3], we propose the following lemma for cMPNN$^{\bullet\bullet}$. Using the same overall framing as Maskey et al. [45, Lemma B.3],

Lemma 5. *Let (χ, d, μ) be a metric-measure space and W be a graphon s.t. Assumptions 1.1-3. are satisfied. Let $\Phi : \mathbb{R}^{2F} \to \mathbb{R}^H$ be Lipschitz continuous with Lipschitz constant $L_\Phi(M_{tr})$. Consider a metric-space signal $f^{\bullet\bullet} : \chi \times \chi \to \mathbb{R}^F$ with $\|f^{\bullet\bullet}\|_\infty < \infty$. Suppose that $X_1, \dots, X_N$ are drawn i.i.d. from μ on χ, and let $p \in (0, 1/H)$. Let $x, y \in \chi$, and define the random variable*

$$\begin{aligned} Y^{\bullet\bullet}_{x,y} = \frac{1}{2N}\sum_{i=1}^{N}\Big(W(y, X_i)\Phi(f^{\bullet\bullet}(x, y), f^{\bullet\bullet}(x, X_i)) + W(x, X_i)\Phi(f^{\bullet\bullet}(x, y), f^{\bullet\bullet}(y, X_i))\Big) \\ - \frac{1}{2}\int_{\mathcal{X}}\Big(W(y, z)\Phi(f^{\bullet\bullet}(x, y), f^{\bullet\bullet}(x, z))) + W(x, z)\Phi(f^{\bullet\bullet}(x, y), f^{\bullet\bullet}(y, z))\Big)d\mu(z) \end{aligned}$$

on the sample space χ^N. Then, with probability at least $1 - Hp$, we have

$$\|Y^{\bullet\bullet}_{x,y}\|_\infty \le \sqrt{2}\frac{(L_\Phi(M_{tr})\|f^{\bullet\bullet}\|_\infty + \|\Phi(0, 0)\|_\infty)\sqrt{\log 2/p}}{\sqrt{N}}. \tag{17}$$

Proof. The proof of the bound is the same as the proof in Maskey et al. [45, Lemma B.3], even though $Y_{x,y}$ is a different quantity than the quantity used in Maskey et al. [45, Lemma B.3]. □

Lemma 6. *Let (χ, d, μ) be a metric-measure space and W be a graphon s.t. Assumptions 1.1-3. are satisfied. Let $\Phi : \mathbb{R}^{2F} \to \mathbb{R}^H$ be Lipschitz continuous with Lipschitz constant $L_\Phi(M_{tr})$. Consider a metric-space signal $f^{\bullet\bullet} : \chi \times \chi \to \mathbb{R}^F$ with $\|f^{\bullet\bullet}\|_\infty < \infty$. Suppose that $X_1, \dots, X_N$ are drawn i.i.d. from μ on χ, and let $p \in (0, 1/H)$. Let $x, y \in \chi$, and define the random variable*

$$\begin{aligned} T^{\bullet\bullet}_{x,y} = \frac{1}{2N}\sum_{i=1}^{N}\Big(A(y, X_i)\Phi(f^{\bullet\bullet}(x, y), f^{\bullet\bullet}(x, X_i)) + A(x, X_i)\Phi(f^{\bullet\bullet}(x, y), f^{\bullet\bullet}(y, X_i))\Big) \\ - \frac{1}{2N}\sum_{i=1}^{N}\Big(W(y, X_i)\Phi(f^{\bullet\bullet}(x, y), f^{\bullet\bullet}(x, X_i)) + W(x, X_i)\Phi(f^{\bullet\bullet}(x, y), f^{\bullet\bullet}(y, X_i))\Big) \end{aligned}$$

on the sample space $\mathcal{X}^N \times [0, 1]^{2N}$. Then, with probability at least $1 - Hp$, we have

$$\|T^{\bullet\bullet}_{x,y}\|_\infty \le \sqrt{2}\frac{(L_\Phi(M_{tr})\|f^{\bullet\bullet}\|_\infty + \|\Phi(0, 0)\|_\infty)\sqrt{\log 2/p}}{\sqrt{N}}. \tag{18}$$

Proof. The proof procedure is the same as Maskey et al. [45, Lemma B.3] where we use $\mathbb{E}[A(y, X_i)] = W(y, X_i)$ and $\mathbb{E}[A(x, X_i)] = W(x, X_i)$. □

Lemma 7. *Let (χ, d, μ) be a metric-measure space and W be a graphon s.t. Assumptions 1.1-3. are satisfied. Let $\Phi : \mathbb{R}^{2F} \to \mathbb{R}^H$ be Lipschitz continuous with Lipschitz constant $L_\Phi(M_{tr})$. Consider a metric-space signal $f^{\bullet\bullet} : \chi \times \chi \to \mathbb{R}^F$ with $\|f^{\bullet\bullet}\|_\infty < \infty$. Suppose that $X_1, \dots, X_N$ are drawn i.i.d. from μ on χ, and let $p \in (0, 1/(2H))$. Let $x, y \in \chi$, and define the random variable*

$$\begin{aligned} R^{\bullet\bullet}_{x,y} = \frac{1}{2N}\sum_{i=1}^{N}\Big(A(y, X_i)\Phi(f^{\bullet\bullet}(x, y), f^{\bullet\bullet}(x, X_i)) + A(x, X_i)\Phi(f^{\bullet\bullet}(x, y), f^{\bullet\bullet}(y, X_i))\Big) \\ - \frac{1}{2}\int_{\mathcal{X}}\Big(W(y, z)\Phi(f^{\bullet\bullet}(x, y), f^{\bullet\bullet}(x, z))) + W(x, z)\Phi(f^{\bullet\bullet}(x, y), f^{\bullet\bullet}(y, z))\Big)d\mu(z) \end{aligned}$$

on the sample space $\chi^N \times [0, 1]^{2N}$. Then, with probability at least $1 - 2Hp$, we have

$$\|R^{\bullet\bullet}_{x,y}\|_\infty \le 2\sqrt{2}\frac{(L_\Phi(M_{tr})\|f^{\bullet\bullet}\|_\infty + \|\Phi(0, 0)\|_\infty)\sqrt{\log 2/p}}{\sqrt{N}}. \tag{19}$$

Proof. Use the triangle inequality and the results from Lemmas 5 and 6.

$$\|R^{\bullet\bullet}_{x,y}\|_\infty = \|T^{\bullet\bullet}_{x,y} + Y^{\bullet\bullet}_{x,y}\|_\infty \leq \|T^{\bullet\bullet}_{x,y}\|_\infty + \|Y^{\bullet\bullet}_{x,y}\|_\infty.$$

From Lemmas 5 and 6, $\|T^{\bullet\bullet}_{x,y}\|_\infty \leq \sqrt{2}\frac{(L_\Phi(M_{\mathrm{tr}})\|f^{\bullet\bullet}\|_\infty + \|\Phi(0,0)\|_\infty)\sqrt{\log 2/p}}{\sqrt{N}}$ w.p. $1 - Hp$ and $\|Y^{\bullet\bullet}_{x,y}\|_\infty \leq \sqrt{2}\frac{(L_\Phi(M_{\mathrm{tr}})\|f^{\bullet\bullet}\|_\infty + \|\Phi(0,0)\|_\infty)\sqrt{\log 2/p}}{\sqrt{N}}$ w.p. $1 - Hp$. With probability at least $1 - 2Hp$, by intersecting the two events, we have

$$\|R^{\bullet\bullet}_{x,y}\|_\infty \leq 2\sqrt{2}\frac{(L_\Phi(M_{\mathrm{tr}})\|f^{\bullet\bullet}\|_\infty + \|\Phi(0,0)\|_\infty)\sqrt{\log 2/p}}{\sqrt{N}}.$$

□

Based on Lemma 7, we can prove the following corollary about the maximum concentration error for all pairs of nodes in the sampled graph G. Using the same overall framing as Maskey et al. [45, Lemma B.3],

Corollary 6. *Let (χ, d, μ) be a metric-measure space and W be a graphon s.t. Assumptions 1.1-3. are satisfied. Let $\Phi : \mathbb{R}^{2F} \to \mathbb{R}^H$ be Lipschitz continuous with Lipschitz constant $L_\Phi(M_{tr})$. Consider a metric-space signal $f^{\bullet\bullet} : \chi \times \chi \to \mathbb{R}^F$ with $\|f^{\bullet\bullet}\|_\infty < \infty$. Suppose that $X_1, \ldots, X_N$ are drawn i.i.d. from μ on χ, and then edges $A_{i,j} \sim Ber(W(X_i, X_j))$ i.i.d sampled. Let $p \in (0, 1/2H)$, and define the random variable*

$$\begin{aligned} R^{\bullet\bullet}_{X_i,X_j} = &\frac{1}{2N}\sum_{z=1}^{N}\Big(A(X_j, X_z)\Phi(f^{\bullet\bullet}(X_i, X_j), f^{\bullet\bullet}(X_i, X_z)) \\ &+ A(X_i, X_z)\Phi(f^{\bullet\bullet}(X_i, X_j), f^{\bullet\bullet}(X_j, X_z))\Big) \\ &- \frac{1}{2}\int_{\mathcal{X}}\Big(W(X_j, z)\Phi(f^{\bullet\bullet}(X_i, X_j), f^{\bullet\bullet}(X_i, z))) \\ &+ W(X_i, z)\Phi(f^{\bullet\bullet}(X_i, X_j), f^{\bullet\bullet}(X_j, z))\Big)d\mu(z) \end{aligned}$$

on the sample space $\chi^N \times [0,1]^{2N}$. Then, with probability at least $1 - 2Hp$, we have

$$\max_{i,j=1,\ldots,N}\|R^{\bullet\bullet}_{X_i,X_j}\|_\infty \leq 2\sqrt{2}\frac{(L_\Phi(M_{tr})\|f^{\bullet\bullet}\|_\infty + \|\Phi(0,0)\|_\infty)\sqrt{\log(2N^2/p)}}{\sqrt{N}}. \tag{20}$$

Proof. Using the result from Lemma 6, we have with probability $1 - \frac{Hp}{N^2}$,

$$\|T^{\bullet\bullet}_{X_i,X_j}\|_\infty \leq \sqrt{2}\frac{(L_\Phi(M_{\mathrm{tr}})\|f^{\bullet\bullet}\|_\infty + \|\Phi(0,0)\|_\infty)\sqrt{\log(2N^2/p)}}{\sqrt{N}}.$$

Using the union bound of the N^2 events that the above equations happens for $x = X_1, ..., X_N$ and $y = X_1, ..., X_N$, with probability at least $1 - Hp$, we have

$$\max_{i,j=1,\ldots,N}\|T^{\bullet\bullet}_{X_i,X_j}\|_\infty \leq \sqrt{2}\frac{(L_\Phi(M_{\mathrm{tr}})\|f^{\bullet\bullet}\|_\infty + \|\Phi(0,0)\|_\infty)\sqrt{\log(2N^2/p)}}{\sqrt{N}}.$$

The same logic can be applied to $Y^{\bullet\bullet}_{X_i,X_j}, \forall i \in \{1, ..., N\}$. Thus, using the triangle inequality, and the union bound of the two events, we have with probability at least $1 - 2Hp$,

$$\max_{i,j=1,\ldots,N}\|R^{\bullet\bullet}_{X_i,X_j}\|_\infty \leq 2\sqrt{2}\frac{(L_\Phi(M_{\mathrm{tr}})\|f^{\bullet\bullet}\|_\infty + \|\Phi(0,0)\|_\infty)\sqrt{\log(2N^2/p)}}{\sqrt{N}}.$$

□

Following Maskey et al. [45, Lemma B.2], we also bound the maximum sampled-graph fraction of common neighbors $c_A(\cdot, \cdot)$ under a condition of the sample size N. Using a same assumption as Maskey et al. [45, Lemma B.2],

Lemma 8. *Let (χ, d, μ) be a metric-measure space and W be a graphon s.t. Assumptions 1.1-3. are satisfied. Suppose that $X_1, \ldots, X_N$ are drawn i.i.d. from μ on χ, and then edges $A_{i,j} \sim$ Ber$(W(X_i, X_j))$ i.i.d sampled. And let $p \in (0, 1)$. If $N \in \mathbb{N}$ satisfy*

$$\sqrt{N} \geq 4\sqrt{2}\frac{\sqrt{\log\left(2N^2/p\right)}}{d_{cmin}}. \tag{21}$$

Then, with probability at least $1 - 2p$, we have

$$\max_{i,j=1,\ldots,N} \|c_A(X_i, X_j) - c_W(X_i, X_j)\|_\infty \leq 2\sqrt{2}\frac{\sqrt{\log\left(2N^2/p\right)}}{\sqrt{N}},$$

and

$$\min_{i,j=1,\ldots,N} c_A(X_i, X_j) \geq \frac{d_{cmin}}{2}. \tag{22}$$

Proof. For given $x, y \in \mathcal{X}$, define the random variable

$$c_A(x, y) - c_X(x, y) = \frac{1}{N}\sum_{i=1}^{N} A(x, X_i)A(y, X_i) - \frac{1}{N}\sum_{i=1}^{N} W(x, X_i)W(y, X_i)$$

on the sample space $\chi^N \times [0, 1]^{2N}$. Using the same proof technique in Lemmas 5 to 7, we can prove with probability at least $1 - 2p$, we have

$$\max_{i,j=1,\ldots,N} \|c_A(X_i, X_j) - c_W(X_i, X_j)\|_\infty \leq 2\sqrt{2}\frac{\sqrt{\log\left(2N^2/p\right)}}{\sqrt{N}},$$

Since $c_W(X_i, X_j) \geq d_{\text{cmin}}$, then with probability at least $1 - 2p$, when Equation (21) is satisfied, we have

$$\min_{i,j=1,\ldots,N} c_A(X_i, X_j) \geq \frac{d_{\text{cmin}}}{2}$$

□

Based on Lemma 8, we can prove a modified version of Maskey et al. [45, Lemma B.5]. Using a same overall framing as Maskey et al. [45, Lemma B.5],

Lemma 9. *Let (χ, d, μ) be a metric-measure space and W be a graphon s.t. Assumptions 1.1-3. are satisfied. Let $\Phi : \mathbb{R}^{2F} \to \mathbb{R}^H$ be Lipschitz continuous with Lipschitz constant $L_\Phi(M_{tr})$. Consider a metric-space signal $f^{\bullet\bullet} : \chi \times \chi \to \mathbb{R}^F$ with $\|f^{\bullet\bullet}\|_\infty < \infty$. Let $p \in (0, \frac{1}{2(H+1)})$, and let $N \in \mathbb{N}$ satisfy (21). Suppose that $X_1, \ldots, X_N$ are drawn i.i.d. from μ in χ, and then edges $A_{i,j} \sim$ Ber$(W(X_i, X_j))$ i.i.d sampled. Then, condition (22) together with (23) below are satisfied in probability at least $1 - 2(H + 1)p$:*

$$\begin{aligned}
&\max_{i,j=1,\ldots,N} \|(M_A^{\bullet\bullet} - M_W^{\bullet\bullet})\big(\Phi(f^{\bullet\bullet}, f^{\bullet\bullet})\big)(X_i, X_j)\|_\infty \\
&\leq 4\frac{\sqrt{2}\sqrt{\log(2N^2/P)}}{\sqrt{N}d_{cmin}^2}(L_\Phi(M_{tr})\|f^{\bullet\bullet}\|_\infty + \|\Phi(0, 0)\|_\infty) \\
&\quad + \frac{2\sqrt{2}(L_\Phi(M_{tr})\|f^{\bullet\bullet}\|_\infty + \|\Phi(0, 0)\|_\infty)\sqrt{\log(2N^2/P)}}{d_{cmin}\sqrt{N}},
\end{aligned} \tag{23}$$

Proof. The proof is the same as Maskey et al. [45, Lemma B.5]. The only difference is on the difference between Lemma 8 and Maskey et al. [45, Lemma B.2], and the difference between Corollary 6 and Maskey et al. [45, Lemma B.4]. □

Same as Maskey et al. [45, Lemma B.6], the layer-wise error for cMPNN$^{\bullet\bullet}$ and a gMPNN$^{\bullet\bullet}$ can be bounded. Using the same overall framing as Maskey et al. [45, Lemma B.6],

Corollary 7. *Let (χ, d, μ) be a metric-measure space and W be a graphon s.t. Assumptions 1.1-3. are satisfied. Let $\Phi : \mathbb{R}^{2F} \to \mathbb{R}^{H}$ and $\Psi : \mathbb{R}^{F+H} \to \mathbb{R}^{F'}$ be Lipschitz continuous with Lipschitz constants $L_\Phi(M_{tr})$ and $L_\Psi(M_{tr})$. Consider a metric-space signal $f^{\bullet\bullet} : \chi \times \chi \to \mathbb{R}^F$ with $\|f^{\bullet\bullet}\|_\infty < \infty$. Let $p \in (0, \frac{1}{2(H+1)})$, and let $N \in \mathbb{N}$ satisfy (21). Suppose that $X_1, \dots, X_N$ are drawn i.i.d. from μ in χ, and then edges $A_{i,j} \sim Ber(W(X_i, X_j))$ i.i.d sampled. Then, condition (22) together with (24) below are satisfied in probability at least $1 - 2(H+1)p$,*

$$\max_{i,j=1,\dots,N} \|\Psi\Big(f^{\bullet\bullet}(\cdot,\cdot), M_A^{\bullet\bullet}\big(\Phi(f^{\bullet\bullet}, f^{\bullet\bullet})\big)(X_i, X_j)\Big) - \Psi\Big(f^{\bullet\bullet}(\cdot,\cdot), M_W^{\bullet\bullet}\big(\Phi(f^{\bullet\bullet}, f^{\bullet\bullet})\big)(X_i, X_j)\Big)\|_\infty$$
$$\leq L_\Psi(M_{tr})\Big(4\frac{\sqrt{2}\sqrt{\log(2N^2/p)}}{\sqrt{N}d_{cmin}^2}(L_\Phi(M_{tr})\|f^{\bullet\bullet}\|_\infty + \|\Phi(0,0)\|_\infty)$$
$$+ \frac{2\sqrt{2}(L_\Phi(M_{tr})\|f^{\bullet\bullet}\|_\infty + \|\Phi(0,0)\|_\infty)\sqrt{\log(2N^2/p)}}{d_{cmin}\sqrt{N}}\Big), \tag{24}$$

Proof. The proof is the same as Maskey et al. [45, Lemma B.6]. The difference comes from the different bound in our Lemma 9 and the bound used by Maskey et al. [45, Lemma B.5]. □

G.2 Proof for Theorem 2

Finally, we can prove Theorem 2. The proof closely follows that of Maskey et al. [45, Theorem B.14], adapted to our setting. Using the same overall framing as Maskey et al. [45, Theorem B.14].

Theorem 2 (OOD convergence without in-distribution convergence)**.** *For a random graph model (W, f) satisfying Definition 1, let N^{tr} be a random variable defining the distribution of graph sizes in training. Define the test distribution $(G^{te}, \boldsymbol{F}^{te}) \sim (W, f)$ through the causal graph in Figure 1 as an interventional change to obtain larger test graph sizes where $\min(supp(N^{te})) \gg M_{tr} = \max(supp(N^{tr}))$ (which means any test graph is much larger than the largest possible training graph). Let $\Theta = ((\Phi^{(l)})_{l=1}^T, (\Psi^{(l)})_{l=1}^T)$ be a MPNN as in Definition 2 with T layers such that $\Phi^{(l)}$ and $\Psi^{(l)}$ that are learned from the training data and are Lipschitz continuous with Lipschitz constants $L_\Phi^{(l)}(M_{tr})$ and $L_\Psi^{(l)}(M_{tr})$. Let gMPNN$^{\bullet\bullet}$ $\Theta_W^{\bullet\bullet(T)}$ and cMPNN$^{\bullet\bullet}$ $\Theta_W^{\bullet\bullet(T)}$ be as in Definitions 9 and 10. For a random graph model (W, f) as in Definition 1 with $d_{cmin} > 0$. Let $X_1^{te}, ..., X_{N^{te}}^{te}$ and $\boldsymbol{A}^{te}$ be as in Definition 1. Let $p \in (0, \frac{1}{\sum_{l=1}^T 2(H_l+1)})$. Then, if $\frac{\sqrt{N^{te}}}{\sqrt{\log(2(N^{te})^2/p)}} \geq \frac{4\sqrt{2}}{d_{cmin}}$, we have with probability at least $1 - \sum_{l=1}^T 2(H_l + 1)p$,*

$$\delta_{\text{A-W}}^{\bullet\bullet} = \max_{i,j=1,\dots,N^{te}} \|\Theta_A^{\bullet\bullet(T)}(\boldsymbol{F}^{\bullet\bullet})_{i,j} - \Theta_W^{\bullet\bullet(T)}(f^{\bullet\bullet})(X_i^{te}, X_j^{te})\|_\infty \leq (C_3 + C_4\|f^{\bullet\bullet}\|_\infty)\frac{\sqrt{\log(2(N^{te})^2/p)}}{\sqrt{N^{te}}},$$

where the constants C_3 and C_4 are defined in the Appendix and depend on $\{L_\Phi^{(l)}(M_{tr}), L_\Psi^{(l)}(M_{tr})\}_{l=1}^T$ and the distribution of $(G^{tr}, \boldsymbol{F}^{tr})$.

Proof. In this case, $\|\Phi^{(l)}(0,0)\|_\infty$, $\|\Psi^{(l)}(0,0)\|_\infty$ can be determined by $(G^{\text{tr}}, \boldsymbol{F}^{\text{tr}})$, N^{tr} if the MPNN Θ has been trained on the training graph $(G^{\text{tr}}, \boldsymbol{F}^{\text{tr}})$.

Following the procedure of Maskey et al. [45, Appendix B.2] with Corollary 7, we can derive similarly, with probability at least $1 - \sum_{l=1}^T (2H_l + 1)p$,

$$\delta_{\text{A-W}}^{\bullet\bullet} \leq \sum_{l=1}^{T} L_\Psi^{(l)}(M_{\text{tr}})\Big(4\frac{\sqrt{2}\sqrt{\log(2(N^{\text{te}})^2/p)}}{\sqrt{N^{\text{te}}}d_{\text{cmin}}^2}(L_\Phi^{(l)}(M_{\text{tr}})\|f^{\bullet\bullet(l)}\|_\infty + \|\Phi^{(l)}(0,0)\|_\infty)$$
$$+ \frac{2\sqrt{2}(L_\Phi^{(l)(M_{\text{tr}})}\|f^{\bullet\bullet(l)}\|_\infty + \|\Phi^{(l)}(0,0)\|_\infty)\sqrt{\log(2(N^{\text{te}})^2/p)}}{d_{\text{cmin}}\sqrt{N^{\text{te}}}}\Big) \tag{25}$$
$$\prod_{l'=l+1}^{T}((L_\Psi^{(l')}(M_{\text{tr}}))^2 + \frac{8}{d_{\text{cmin}}^2}(L_\Phi^{(l')}(M_{\text{tr}}))^2(L_\Psi^{(l')}(M_{\text{tr}}))^2),$$

Using the same proof in Maskey et al. [45, Lemma B.9], we can derive

$$||f^{\bullet\bullet(l)}||_\infty \leq B_1^{\bullet\bullet(l)} + B_2^{\bullet\bullet(l)}||f||_\infty,$$

where ${B_1}^{\bullet\bullet(l)}$, ${B_2}^{\bullet\bullet(l)}$ are independent of $f^{\bullet\bullet}$, and

$$\begin{aligned} B_1^{\bullet\bullet(l+1)} = \sum_{k=1}^{l+1} \big(L_\Psi^{(k)}(M_{\text{tr}}) \frac{1}{d_{\text{cmin}}} \|\Phi^{(k)}(0,0)\|_\infty + \|\Psi^{(k)}(0,0)\|_\infty\big) \\ \prod_{l'=k+1}^{l+1} L_\Psi^{(l')}(M_{\text{tr}})\big(1 + \frac{1}{d_{\text{cmin}}} L_\Phi^{(l')}(M_{\text{tr}})\big) \end{aligned} \tag{26}$$

and

$$B_2^{\bullet\bullet(l+1)} = \prod_{k=1}^{l+1} L_\Psi^{(k)}(M_{\text{tr}}) \left(1 + \frac{1}{d_{\text{cmin}}} L_\Phi^{(k)}(M_{\text{tr}})\right). \tag{27}$$

Now we can decompose the summation in Equation (25). First, we defince C_3 as

$$\begin{aligned} C_3 = \sum_{l=1}^{T} L_\Psi^{(l)}(M_{\text{tr}}) \Big(4\frac{\sqrt{2}}{d_{\text{cmin}}^2} (L_\Phi^{(l)}(M_{\text{tr}}) B_1^{\bullet\bullet(l)} + \|\Phi^{(l)}(0,0)\|_\infty) \\ + \frac{2\sqrt{2}(L_\Phi^{(l)}(M_{\text{tr}}) B_1^{\bullet\bullet(l)} + \|\Phi^{(l)}(0,0)\|_\infty)}{d_{\text{cmin}}}\Big) \\ \prod_{l'=l+1}^{T} ((L_\Psi^{(l')}(M_{\text{tr}}))^2 + \frac{8}{d_{\text{cmin}}^2} (L_\Phi^{(l')}(M_{\text{tr}}))^2 (L_\Psi^{(l')}(M_{\text{tr}}))^2), \end{aligned} \tag{28}$$

Then we can define C_4 as

$$\begin{aligned} C_4 = \sum_{l=1}^{T} L_\Psi^{(l)}(M_{\text{tr}}) \Big(4\frac{\sqrt{2}}{d_{\text{cmin}}^2} L_\Phi^{(l)}(M_{\text{tr}}) B_2^{\bullet\bullet(l)} + \frac{2\sqrt{2} L_\Phi^{(l)}(M_{\text{tr}}) B_2^{\bullet\bullet(l)}}{d_{\text{cmin}}}\Big) \\ \prod_{l'=l+1}^{T} ((L_\Psi^{(l')}(M_{\text{tr}}))^2 + \frac{8}{d_{\text{cmin}}^2} (L_\Phi^{(l')}(M_{\text{tr}}))^2 (L_\Psi^{(l')}(M_{\text{tr}}))^2), \end{aligned} \tag{29}$$

It is clear to see we can rewrite Equation (25) as

$$\delta_{\text{A-W}}^{\bullet\bullet} \leq (C_3 + C_4 \|f^{\bullet\bullet}\|_\infty) \frac{\sqrt{\log(2(N^{\text{te}})^2/p)}}{\sqrt{N^{\text{te}}}}. \tag{30}$$

Thus C_3 and C_4 depends on $\{L_\Phi^{(l)}(M_{\text{tr}})\}_{l=1}^T$ and $\{L_\Psi^{(l)}(M_{\text{tr}})\}_{l=1}^T$ and possibly on $(G^{\text{tr}}, \boldsymbol{F}^{\text{tr}})$ and N^{tr}. □