Size-Invariant Graph Representations for Graph Classification Extrapolations

Beatrice Bevilacqua [*] [1] **Yangze Zhou** [*] [2] **Bruno Ribeiro** [1]

Abstract

In general, graph representation learning methods assume that the train and test data come from the same distribution. In this work we consider an underexplored area of an otherwise rapidly developing field of graph representation learning: The task of out-of-distribution (OOD) graph classification, where train and test data have different distributions, with test data unavailable during training. Our work shows it is possible to use a causal model to learn approximately invariant representations that better extrapolate between train and test data. Finally, we conclude with synthetic and real-world dataset experiments showcasing the benefits of representations that are invariant to train/test distribution shifts.

1. Introduction

In general, graph representation learning methods assume that the train and test data come from the same distribution. Unfortunately, this assumption is not always valid in real-world deployments (Hu et al., 2020; Koh et al., 2020; D'Amour et al., 2020). When the test distribution is different from training, the test data is described as *out of distribution (OOD)*. Differences in train/test distribution may be due to environmental factors such as those related to the way the data is collected or processed.

Particularly, in graph classification tasks, where $\mathcal{G}$ is the graph and Y its label, we often see different graph sizes and/or distinct arrangements of vertex attributes associated with the same target label. *How should we learn a graph representation for out-of-distribution inductive tasks (extrapolations), where the graphs in training and test (deployment) have distinct characteristics (i.e.,* $\mathrm{P}^{tr}(\mathcal{G}) \neq \mathrm{P}^{te}(\mathcal{G})$*)?* Are inductive graph neural networks (GNNs) robust to distribution shifts between $\mathrm{P}^{\mathrm{tr}}(\mathcal{G})$ and $\mathrm{P}^{\mathrm{te}}(\mathcal{G})$? If not, is it possible to design a graph classifier that is robust to such OOD shifts without access to samples from $\mathrm{P}^{\mathrm{te}}(\mathcal{G})$?

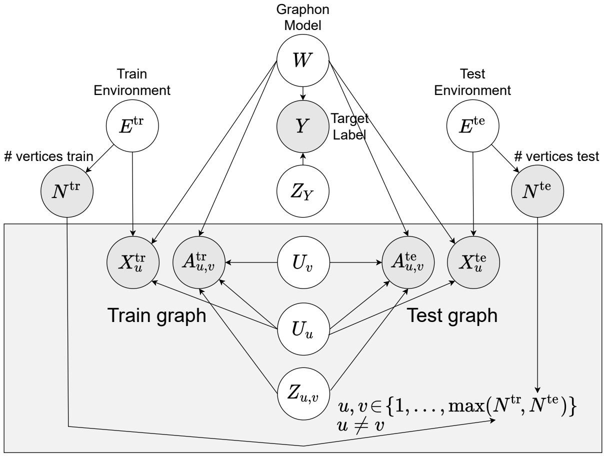

Figure 1. The twin network DAG (Balke & Pearl, 1994) of our structural causal model (SCM). Gray (resp. white) vertices represent observed (resp. hidden) random variables.

In this work we consider an OOD graph classification task with different train and test distributions based on graph sizes and vertex attributes. Our work focuses on simple (no self-loops) undirected graphs with discrete vertex attributes. We make the common assumption of independence between cause and mechanisms (Bengio et al., 2020; Besserve et al., 2018; Johansson et al., 2016; Louizos et al., 2017; Raj et al., 2020; Schölkopf, 2019; Arjovsky et al., 2019), which states that $\mathrm{P}(Y|\mathcal{G})$ remains the same between train and test. We also assume we do not have access to samples from $\mathrm{P}^{\mathrm{te}}(\mathcal{G})$, hence covariate shift adaptation methods (such as Yehudai et al. (2021)) are unfit for our scenario. In our setting we need to learn to extrapolate from a causal model.

Contributions. Our contributions are as follows:

1. We provide a causal model that formally describes a class of graph classification tasks where the training ($\mathrm{P}^{\mathrm{tr}}(\mathcal{G})$) and test ($\mathrm{P}^{\mathrm{te}}(\mathcal{G})$) graphs have different size and vertex attribute distributions.

2. Assuming Independence between Cause and Mechanism (ICM) (Louizos et al., 2017; Shajarisales et al., 2015), we introduce a graph representation method based on the work of Lovász & Szegedy (2006) and Graph Neural

[*]Equal contribution [1]Department of Computer Science, and [2]Department of Statistics, Purdue University, West Lafayette, Indiana, USA. Correspondence to: Beatrice Bevilacqua <bbevilac@purdue.edu>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

Networks (GNNs) (Kipf & Welling, 2017; Hamilton et al., 2017; You et al., 2019) that is invariant to the train/test distribution shifts of our causal model. Unlike existing invariant representations, this representation can perform extrapolations from single training environment (e.g., all training graphs have the same size).

3. Our empirical results show that, in most experiments, neither Invariant Risk Minimization (IRM) (Arjovsky et al., 2019) nor the GNN extrapolation modifications proposed by Xu et al. (2021) are able to perform well in graph classification tasks over the OOD test data.

2. Graph Classification: A Causal Model Based on Random Graphs

Out-of-distribution (OOD) shift. For any joint distribution $\mathrm{P}(Y, \mathcal{G})$ of graphs $\mathcal{G}$ and labels Y, there are infinitely many causal models that give the same joint distribution (Pearl, 2009). This phenomenon is known as model underspecification. Hence, if the training data distribution $\mathrm{P}^{\text{tr}}(Y, \mathcal{G})$ does not have the same support as the test distribution $\mathrm{P}^{\text{te}}(Y, \mathcal{G})$, a model trained with samples drawn from $\mathrm{P}^{\text{tr}}(Y, \mathcal{G})$ needs to be able to extrapolate in order to correctly predict $\mathrm{P}^{\text{te}}(Y|\mathcal{G})$. In this work, we assume Independence between Cause and Mechanism (ICM): $\mathrm{P}^{\text{tr}}(Y|\mathcal{G}) = \mathrm{P}^{\text{te}}(Y|\mathcal{G})$, which is a common assumption in the causal deep learning literature (Bengio et al., 2020; Besserve et al., 2018; Johansson et al., 2016; Louizos et al., 2017; Raj et al., 2020; Schölkopf, 2019; Arjovsky et al., 2019).

In inductive graph classification tasks, ICM implies that the shift between train and test distributions $\mathrm{P}^{\text{tr}}(Y, \mathcal{G}) \neq \mathrm{P}^{\text{te}}(Y, \mathcal{G})$ comes from $\mathrm{P}^{\text{tr}}(\mathcal{G}) \neq \mathrm{P}^{\text{te}}(\mathcal{G})$, since $\mathrm{P}^{\text{tr}}(Y|\mathcal{G}) = \mathrm{P}^{\text{te}}(Y|\mathcal{G})$. And because our task is inductive, i.e., no data from $\mathrm{P}^{\text{te}}(\mathcal{G})$ or a proxy variable, we must make assumptions about the causal mechanisms in order to extrapolate.

Causal model. A graph representation that is robust (invariant) to shifts in $\mathrm{P}^{\text{te}}(\mathcal{G})$ must know how the distribution shifts. Either we are given some examples from $\mathrm{P}^{\text{te}}(\mathcal{G})$ (a.k.a. covariate shift adaptation (Sugiyama et al., 2007)) or we are given a causal structure that describes how the test distribution can shift. Our paper focuses on the latter by giving a Structural Causal Model (SCM) for the data generation process in Definitions 1 and 2. The definition of the Structural Causal Model (SCM) is needed since the observational probability itself does not provide any causal information (see observational equivalence in Pearl (2009, Theorem 1.2.8)). Figure 1 depicts the Directed Acyclic Graph (DAG) of our causal model. It uses the twin network DAGs structure first proposed by Balke & Pearl (1994) (see Pearl (2009, Chapter 7.1.4)) in order to define how the test distribution can change.

In what follows we detail the SCM in Definitions 1 and 2. Our causal model is inspired by Stochastic Block Models (SBMs) (Diaconis & Freedman, 1981; Snijders & Nowicki, 1997) and their connection to graphon random graph models (Airoldi et al., 2013; Lovász & Szegedy, 2006):

Definition 1 (Training Graph $\mathcal{G}^{\text{tr}}_{N^{\text{tr}}}$)**.** *The training graph SCM is depicted at the left side of the twin network DAG in Figure 1.*

- *The training graph is characterized by a graphon $W \sim \mathrm{P}(W)$, where $W : [0,1]^2 \to [0,1]$ is a random symmetric measurable function (Lovász & Szegedy, 2006) sampled (according to some distribution) from $\mathbb{D}_W$, the set of all symmetric measurable functions on $[0,1]^2 \to [0,1]$. W defines both the graph's target label and some of its structural and attribute characteristics, but W is unknown.*
- *The* **training environment** *$E^{tr} \sim \mathrm{P}^{tr}(E)$ is a hidden environment variable that represents specific graph properties that change between the training and test. $E^{tr} \in \mathbb{E}$ for some properly defined environment space $\mathbb{E}$.*
- *The graph's size is determined by its environment $N^{tr} := \eta(E^{tr})$, where η is an unknown deterministic function.*
- *The graph's target label is given by $Y := h(W, Z_Y)$, $Y \in \mathbb{Y}$, with $\mathbb{Y}$ some properly defined discrete target space. Z_Y is an independent random noise variable and h is a deterministic function on the input space $\mathbb{D}_W \times \mathbb{R}$.*
- *The vertices are numbered $V^{tr} = \{1, \ldots, N^{tr}\}$. Each vertex $v \in V^{tr}$ has an associated hidden variable $U_v \sim$ Uniform$(0,1)$ sampled i.i.d.. The graph is undirected and its adjacency matrix $A^{tr} \in \{0,1\}^{N^{tr} \times N^{tr}}$ is defined by*
$$A^{tr}_{u,v} := \mathbb{1}(Z_{u,v} > W(U_u, U_v)), \forall u, v \in V^{tr}, u \neq v. \quad (1)$$
The diagonals are set to 0 because there is no self-loop. Here $\mathbb{1}$ is an indicator function, and $\{Z_{u,v} = Z_{v,u}\}_{u,v \in V^{tr}}$ are independent uniform noises on $[0,1]$.
- *The graph may contain discrete vertex attributes $X^{tr} \in \mathbb{X}^{N^{tr}}$ defined as*
$$X^{tr}_v := g_X(E^{tr}, W(U_v, U_v)), \ \forall v \in V^{tr},$$
where $X^{tr}_v \in \mathbb{X}$, and $\mathbb{X}$ is some properly defined attribute space. g_X is a deterministic function that determines a vertex attribute using $W(U_v, U_v) \in [0,1]$ via, say, inverse sampling (Tweedie, 1945) the vertex attribute distribution.
- *Then, the training graph is*
$$\mathcal{G}^{tr}_{N^{tr}} := (A^{tr}, X^{tr}).$$

The test data comes from the following (coupled) distribution, that is, the model uses some of the same random variables of the training graph model, effectively only replacing E^{tr} by E^{te}, as shown in the DAG of Figure 1.

Definition 2 (Test Graph $\mathcal{G}^{\text{te}}_{N^{\text{te}}}$). *The SCM of the test graph is given by the right side of the twin network DAG in Figure 1, changing the following variables from Definition 1:*

- *The* **test environment** $E^{te} \sim \mathrm{P}^{te}(E)$*, and* $E^{te} \in \mathbb{E}$ *belongs to the same space as* E^{tr}*. It represents specific properties of the graphs that change between the test and training data. Denote* $supp(\cdot) := \{x|\mathrm{P}(x) > 0\}$ *as the support of a random variable. The supports of* E^{te} *and* E^{tr} *may not overlap (i.e.,* $supp(E^{te}) \cap supp(E^{tr}) = \emptyset$*).*
- *The change in environment from* E^{tr} *to* E^{te} *may change the graph's size as* $N^{te} := \eta(E^{te})$*, where* η *is the same unknown deterministic function as in Definition 1.*
- *The vertices are numbered* $V^{te} = \{1, \ldots, N^{te}\}$*. The adjacency matrix* $A^{te} \in \{0,1\}^{N^{te} \times N^{te}}$ *is defined as in Equation (1).*
- *The graph may contain discrete vertex attributes* $X^{te} \in \mathbb{X}^{N^{te}}$ *defined as*

  $$X^{te}_v := g_X(E^{te}, W(U_v, U_v)),\ \forall v \in V^{te},$$

  with g_X *as given in Definition 1.*
- *Then the test graph is*

  $$\mathcal{G}^{te}_{N^{te}} := (A^{te}, X^{te}).$$

Our SCM has a direct connection with graphon random graph model (Lovász & Szegedy, 2006), and extends it by considering vertex attributes. Now we introduce examples of our graph classification tasks based on Definitions 1 and 2 using two classic random graph models.

Notation: $(\mathcal{G}^*_{N^*}, E^*, A^*, V^*, X^*)$ In what follows we use the superscript * as a wildcard to describe both train and test random variables. For instance, $\mathcal{G}^*_{N^*}$ is a variable that is a wildcard for referring to either $\mathcal{G}^{\text{tr}}_{N^{\text{tr}}}$ or $\mathcal{G}^{\text{te}}_{N^{\text{te}}}$. Also, from now on we define $\mathrm{P}^{\text{te}}(\mathcal{G}) = \mathrm{P}(\mathcal{G}^{\text{te}}_{N^{\text{te}}})$ and $\mathrm{P}^{\text{tr}}(\mathcal{G}) = \mathrm{P}(\mathcal{G}^{\text{tr}}_{N^{\text{tr}}})$.

Erdős-Rényi example. Consider a random training environment E^{tr} such that $N^{\text{tr}} = \eta(E^{\text{tr}})$ is the number of vertices for graphs in our training data. Let p be the probability that any two distinct vertices of the graph have an edge. Define W as a constant function that always outputs p. Sample independent uniform noises $Z_{u,v} \sim \text{Uniform}(0,1)$ (for each possible edge, $Z_{u,v} = Z_{v,u}$). An Erdős-Rényi graph can be defined as a graph whose adjacency matrix A^{tr} is $A^{\text{tr}}_{u,v} = \mathbb{1}(Z_{u,v} > W(U_u, U_v)) = \mathbb{1}(Z_{u,v} > p)$, $\forall u, v \in V^{\text{tr}}, u \neq v$. Here vertex attributes are not considered and we can define $X^{\text{tr}}_v = \emptyset, \forall v \in V^{\text{tr}}$ as the null attribute.

In the test data, we have a different environment E^{te} and graph size $N^{\text{te}} = \eta(E^{\text{te}})$, with $\text{supp}(N^{\text{te}}) \cap \text{supp}(N^{\text{tr}}) = \emptyset$. The variable $\{Z_{u,v}\}_{u,v\in\{1,\ldots,\max(\text{supp}(N^{\text{tr}})\cup\text{supp}(N^{\text{te}}))\}}$ can be thought as the seed of a random number generator to determine if two distinct vertices u and v are connected by an edge. The above defines our training and test data as a set of Erdős-Rényi random graphs of sizes N^{tr} and N^{te} with probability p. The targets of the Erdős-Rényi graphs can be, for instance, the value $Y = p$ in Definition 1, which is determined by W and invariant to graph sizes.

Stochastic Block Model (SBM) (Snijders & Nowicki, 1997). An SBM can be seen as a generalization of Erdős-Rényi graphs. SBMs partition the vertex set into disjoint subsets $S_1, S_2, ..., S_r$ (known as blocks or communities) with an associated $r \times r$ symmetric matrix $\boldsymbol{P}$, where the probability of an edge (u, v), $u \in S_i$ and $v \in S_j$ is $\boldsymbol{P}_{ij}$, for $i, j \in \{1, \ldots, r\}$. In the training and test data, we still have i.i.d sampled $Z_{u,v} = Z_{v,u}$ and different environments E^{tr}, E^{te}. Divide the interval $[0,1]$ into disjoint convex sets $[t_0, t_1), [t_1, t_2), \ldots, [t_{r-1}, t_r]$, where $t_0 = 0$ and $t_r = 1$, such that if $U_v \sim \text{Uniform}(0,1)$ satisfies $U_v \in [t_{i-1}, t_i)$, then vertex v belongs to block S_i. Thus $W(U_u, U_v) = \sum_{i,j\in\{1,\ldots,r\}} P_{ij}\mathbb{1}(U_u \in [t_{i-1}, t_i))\mathbb{1}(U_v \in [t_{j-1}, t_j))$. An SBM graph in training or test can be defined as a graph whose adjacency matrix A^* is $A^*_{u,v} = \mathbb{1}(Z_{u,v} > W(U_u, U_v))$, $\forall u, v \in V^*, u \neq v$. Now we have a set of SBM random graphs of sizes N^{tr} and N^{te} with $\boldsymbol{P}$. Consider if there are only two blocks, the target Y can be $\boldsymbol{P}_{1,2}$ which is the probability of an edge connecting vertices between the blocks, determined by W and invariant to graph sizes.

SBM with vertex attributes. For the SBM, assume the vertex attributes are tied to blocks, and are distinct for each block. The environment variable operates on changing the distributions of attributes assigned in each block. Consider the following SBM example with two blocks: Define $W(U_v, U_v) = \frac{U_v}{2t_1}\mathbb{1}(U_v \in [0, t_1)) + (\frac{1}{2} + \frac{U_v - t_1}{2(1-t_1)})\mathbb{1}(U_v \in [t_1, 1])$. So $W(U_v, U_v) < \frac{1}{2}$ if and only if v belongs to the first block. We only change the values of W for points on a zero-measure space. Let g_X be such that it defines constants as $0 < \alpha_{E^*,1} < \frac{1}{2} < \alpha_{E^*,2} < 1$, and vertex attributes as

$$X^*_v = g_X(E^*, W(U_v, U_v)) = \begin{bmatrix} \mathbb{1}(W(U_v, U_v) \in [0, \alpha_{E^*,1})) \\ \mathbb{1}(W(U_v, U_v) \in [\alpha_{E^*,1}, .5)) \\ \mathbb{1}(W(U_v, U_v) \in [.5, \alpha_{E^*,2})) \\ \mathbb{1}(W(U_v, U_v) \in [\alpha_{E^*,2}, 1]) \end{bmatrix},$$

where the attribute of vertex v, X^*_v, is one-hot encoded to represent 4 colors: red and blue (if v is in block 1) and green and yellow (if v is in block 2).

3. E-Invariant Graph Representations

In this section we discuss shortcomings of traditional graph representation methods for out-of-distribution (OOD) graph classification tasks. We will base our discussion on our Structural Causal Model (SCM) (described in Definitions 1 and 2 and Figure 1). We show that there is an approximately environment-invariant graph representation that is able to extrapolate to OOD test data.

The shortcomings of standard graph representation methods. Figure 1 shows that our target variable Y is a function only of the *graphon* variable W, rather than the training or test environments, E^{tr} and E^{te}, respectively. However, Y is not independent of E^{tr} given $\mathcal{G}^{\text{tr}}_{N^{\text{tr}}}$, since both E^{tr} and W affect A^{tr} and X^{tr} (which are colliders), and Y depends on W. Hence, traditional graph representation learning methods can pick up this easy spurious correlation in the training data (via shorcut learning (Geirhos et al., 2020)), which would prevent the model learning the correct OOD test predictor.

To address the challenge of correctly predicting Y in our OOD test data, regardless of spurious correlations between the variables, we need an estimator that can account for it. In what follows we focus on **environment-invariant (E-invariant)** graph representations. To show the ability of E-invariant representations to extrapolate to OOD test data, we introduce the definition and the effect on downstream OOD classification tasks in the following proposition.

Proposition 1. *[E-invariant Representation's Effect on OOD Classification] Consider a permutation-invariant graph representation* $\Gamma : \cup_{n=1}^{\infty}\{0,1\}^{n\times n} \times \mathbb{X}^n \to \mathbb{R}^d$, $d \geq 1$, *and a downstream function* $\rho : \mathbb{Y} \times \mathbb{R}^d \to [0,1]$ *(e.g., a feedforward neural network (MLP) with softmax outputs) such that, for some* $\epsilon, \delta > 0$*, the generalization error over the training distribution is:* $\forall y \in \mathbb{Y}$,

$$\mathrm{P}(\ |\mathrm{P}(Y = y|\mathcal{G}^{tr}_{N^{tr}}) - \rho(y, \Gamma(\mathcal{G}^{tr}_{N^{tr}}))|\ \leq \epsilon) \geq 1 - \delta,$$

Γ *is said to be* **environment-invariant (E-invariant)** *if* $\forall e \in supp(E^{tr}), \forall e^{\dagger} \in supp(E^{te})$,

$$\Gamma(\mathcal{G}^{tr}_{N^{tr}}|E^{tr} = e) = \Gamma(\mathcal{G}^{te}_{N^{te}}|E^{te} = e^{\dagger}).$$

If Γ *is E-invariant, then the OOD test error is the same as the generalization error over the training distribution, i.e.,* $\forall y \in \mathbb{Y}$,

$$\mathrm{P}(|\mathrm{P}(Y = y|\mathcal{G}^{te}_{N^{te}}) - \rho(y, \Gamma(\mathcal{G}^{te}_{N^{te}}))| \leq \epsilon) \geq 1 - \delta. \quad (2)$$

Proposition 1 shows that an E-invariant representation will perform no worse on the OOD test data (extrapolation samples from $(Y, \mathcal{G}^{\text{te}}_{N^{\text{te}}})$) than on a test dataset having the same environment distribution as the training data (samples from $(Y, \mathcal{G}^{\text{tr}}_{N^{\text{tr}}})$). *Our task now becomes finding an E-invariant graph representation* Γ *that can be used to predict* Y.

The shortcomings of Invariant Risk Minimization (IRM). Invariant Risk Minimization (IRM) (Arjovsky et al., 2019) aims to learn a representation that is invariant across all training environments, $\forall e \in \text{supp}(E^{\text{tr}})$, by adding a regularization penalty on the empirical risk. However, IRM will fail if: (i) $\text{supp}(E^{\text{te}}) \not\subseteq \text{supp}(E^{\text{tr}})$, since the penalty provides no guarantee that the representation will still be invariant w.r.t. $e^{\dagger} \in \text{supp}(E^{\text{te}})\backslash\text{supp}(E^{\text{tr}})$ if the representation is a nonlinear function of the input (Rosenfeld et al., 2020); and (ii) if the training data only contains a single environment, i.e., $\text{supp}(E^{\text{tr}}) = \{e\}$. For instance, the training data may contain only graphs of a single size. In this case, we are unable to apply IRM for size extrapolations. Our experiments show that the IRM procedure does not seem to work for graph representation learning.

In what follows we leverage the stability of subgraph densities (more precisely, induced homomorphism densities) in graphon random graph models (Lovász & Szegedy, 2006) to learn E-invariant representations for the SCM defined in Definitions 1 and 2, whose DAG is illustrated in Figure 1.

3.1. An Approximately E-Invariant Graph Representations for Our Model

Let $\mathcal{G}^*_{N^*}$ denote either an N^{tr}-sized train or N^{te}-sized test graph from the SCM in Definitions 1 and 2. For a given k-vertex graph F_k $(k < N^*)$, let $\text{ind}(F_k, \mathcal{G}^*_{N^*})$ be the number of induced homomorphisms of F_k into $\mathcal{G}^*_{N^*}$, informally, the number of mappings from $V(F_k)$ to $V(\mathcal{G}^*_{N^*})$ such that the corresponding subgraph induced in $\mathcal{G}^*_{N^*}$ is isomorphic to F_k. The induced homomorphism density is defined as

$$t_{\text{ind}}(F_k, \mathcal{G}^*_{N^*}) = \frac{\text{ind}(F_k, \mathcal{G}^*_{N^*})}{N^*!/(N^* - k)!}, \quad (3)$$

where the denominator is the number of possible mappings. Let $\mathcal{F}_{\leq k}$ be the set of all connected vertex-attributed graphs of size $k' \leq k$. Using the subgraph densities (induced homomorphism densities) $\{t_{\text{ind}}(F_{k'}, \mathcal{G}^*_{N^*})\}_{F_{k'} \in \mathcal{F}_{\leq k}}$ we will construct a (feature vector) representation for $\mathcal{G}^*_{N^*}$, similar to Hancock & Khoshgoftaar (2020); Pinar et al. (2017),

$$\Gamma_{\text{1-hot}}(\mathcal{G}^*_{N^*}) = \sum_{F_{k'} \in \mathcal{F}_{\leq k}} t_{\text{ind}}(F_{k'}, \mathcal{G}^*_{N^*}) \mathbf{1}_{\text{one-hot}}\{F_{k'}, \mathcal{F}_{\leq k}\}, \quad (4)$$

where $\mathbf{1}_{\text{one-hot}}\{F_{k'}, \mathcal{F}_{\leq k}\}$ assigns a unique one-hot vector to each distinct graph $F_{k'}$ in $\mathcal{F}_{\leq k}$. For instance, for $k = 4$, the one-hot vectors could be (1,0,. . . ,0)= , (0,1,. . . ,0)= , (0,0,. . . ,1,. . . ,0)= , (0,0,. . . ,1)= , etc.. In Section 3.2 we show that the (feature vector) representation in Equation (4) is approximately environment-invariant in our SCM model.

An alternative approach is to replace the one-hot vector representation with learnable graph representation models. We first use Graph Neural Networks (GNNs) (Kipf & Welling, 2017; Hamilton et al., 2017; You et al., 2019) to learn representations that can capture information from vertex attributes. Simply speaking, GNNs proceed by vertices passing messages, amongst each other, through a learnable function such as an MLP, and repeating $L \in \mathbb{Z}_{\geq 1}$ layers.

Consider the following simple GNN example. Let V^* be the set of vertices. At each iteration $l \in \{1, 2, \ldots, L\}$, all vertices $v \in V^*$ are associated with a learned vector $\boldsymbol{h}_v^{(l)}$. Specifically, we begin by initializing a vector as $\boldsymbol{h}_v^{(0)} = X_v$ for every vertex $v \in V^*$. Then, we recursively compute an update such as the following $\forall v \in V^*$,

$$\boldsymbol{h}_v^{(l)} = \text{MLP}^{(l)}\Big(\boldsymbol{h}_v^{(l-1)}, \text{READOUT}_{\text{Neigh}}((\boldsymbol{h}_u^{(l-1)})_{u \in \mathcal{N}(v)})\Big), \tag{5}$$

where $\mathcal{N}(v) \subseteq V^*$ denotes the neighborhood set of v in the graph, $\text{READOUT}_{\text{Neigh}}$ is a permutation-invariant function (e.g. sum) of the neighborhood learned vectors, $\text{MLP}^{(l)}$ denotes a multi-layer perceptron and whose superscript l indicates that the MLP at each recursion layer may have different learnable parameters. There are other alternatives to Equation (5) that we will also test in our experiments.

Then, we arrive to the following representation of $\mathcal{G}^*_{N^*}$:

$$\Gamma_{\text{GNN}}(\mathcal{G}^*_{N^*}) = \sum_{F_{k'} \in \mathcal{F}_{\leq k}} t_{\text{ind}}(F_{k'}, \mathcal{G}^*_{N^*}) \text{READOUT}_\Gamma(\text{GNN}(F_{k'})), \tag{6}$$

where READOUT_Γ is a permutation-invariant function that maps the vertex-level outputs of a GNN to a graph-level representation (e.g. by summing all vertex embeddings). Unfortunately, GNNs are not most-expressive representations of graphs (Morris et al., 2019; Murphy et al., 2019; Xu et al., 2019) and thus $\Gamma_{\text{GNN}}(\cdot)$ is less expressive than $\Gamma_{\text{1-hot}}(\cdot)$. A representation with greater expressive power is

$$\Gamma_{\text{GNN}^+}(\mathcal{G}^*_{N^*}) = \sum_{F_{k'} \in \mathcal{F}_{\leq k}} t_{\text{ind}}(F_{k'}, \mathcal{G}^*_{N^*}) \text{READOUT}_\Gamma(\text{GNN}^+(F_{k'})), \tag{7}$$

where GNN^+ is a most-expressive k'-vertex graph representation, which can be achieved by any of the methods of Vignac et al. (2020); Maron et al. (2019a); Murphy et al. (2019). Since GNN^+ is most expressive, GNN^+ can ignore attributes and map each $F_{k'}$ to a one-hot vector $\mathbf{1}_{\text{one-hot}}\{F_{k'}, \mathcal{F}_{\leq k}\}$; therefore, $\Gamma_{\text{GNN}^+}(\cdot)$ generalizes $\Gamma_{\text{1-hot}}(\cdot)$ of Equation (4). But *note that greater expressiveness does not imply better extrapolation*.

More importantly, GNN and GNN^+ representations allow us to increase their E-invariance by adding a penalty for having different representations of two graphs $F_{k'}$ and $H_{k'}$ with the same topology but different vertex attributes (say, $F_{k'} =$ and $H_{k'} =$), as long as these differences do not significantly impact downstream model accuracy in the training data. Note that this is more powerful than simply masking vertex attributes, since it allows same-topology graphs with distinct vertex attributes to have different representations if it is important to distinguish them for the target prediction (see Section 5.2). We will discuss more about these theoretical underpinnings in the next section. Hence, for each k'-sized vertex-attributed graph $F_{k'}$, we consider the set $\mathcal{H}(F_{k'})$ of all k'-sized vertex-attributed graphs having the same underlying topology as $F_{k'}$ but with all possible different vertex attributes. We then define the regularization penalty

$$\frac{1}{|\mathcal{F}_{\leq k}|} \sum_{F_{k'} \in \mathcal{F}_{\leq k}} \mathbb{E}_{H_{k'} \in \mathcal{H}(F_{k'})} \|\text{READOUT}_\Gamma(\text{GNN}^*(F_{k'})) - \text{READOUT}_\Gamma(\text{GNN}^*(H_{k'}))\|_2, \tag{8}$$

where $\text{GNN}^* = \text{GNN}$ if we choose the representation Γ_{GNN}, or $\text{GNN}^* = \text{GNN}^+$ if we choose the representation Γ_{GNN^+}. In practice, we assume $H_{k'}$ is uniformly sampled from $\mathcal{H}(F_{k'})$ and we sample one $H_{k'}$ for each $F_{k'}$ in order to obtain an unbiased estimator of Equation (8).

Practical considerations. Efficient algorithms exist to obtain *induced* homomorphism densities over all possible *connected* k-vertex subgraphs (Ahmed et al., 2016; Bressan et al., 2017; Chen & Lui, 2018; Chen et al., 2016; Rossi et al., 2019; Wang et al., 2014). For unattributed graphs and $k \leq 5$, we use ESCAPE (Pinar et al., 2017) to obtain *exact* densities. For attributed graphs or unattributed graphs with $k > 5$, exact counting becomes intractable, so we use R-GPM (Teixeira et al., 2018) to obtain unbiased estimates of densities. Finally, Proposition 2 in Appendix C shows that certain biased estimators can also be used if READOUT_Γ is the sum of vertex embeddings.

3.2. Theoretical Description of our E-Invariant Graph Representations

In this section, we show that the graph representations seen in the previous section are approximately environment-invariant in our SCM model under mild assumptions.

Theorem 1 (Approximately E-invariant Graph Representation)**.** *Let $\mathcal{G}^{tr}_{N^{tr}}$ and $\mathcal{G}^{te}_{N^{te}}$ be two samples of graphs of sizes N^{tr} and N^{te} from the training and test distributions, respectively, both defined over the same graphon variable W and satisfying Definitions 1 and 2. Assume the vertex attribute function $g_X(\cdot, \cdot)$ of Definitions 1 and 2 is invariant to E^{tr} and E^{te} (the reason for this assumption will be clear later). Let $\|\cdot\|_\infty$ denote the L-infinity norm. For any integer $k \leq \min(N^{tr}, N^{te})$, and any constant $0 < \epsilon < 1$,*

$$\text{P}(\|\Gamma_{\textit{1-hot}}(\mathcal{G}^{tr}_{N^{tr}}) - \Gamma_{\textit{1-hot}}(\mathcal{G}^{te}_{N^{te}})\|_\infty > \epsilon) \leq 2|\mathcal{F}_{\leq k}|(\exp(-\frac{\epsilon^2 N^{tr}}{8k^2}) + \exp(-\frac{\epsilon^2 N^{te}}{8k^2})). \tag{9}$$

Theorem 1 shows how the graph representations given in Equation (4) are approximately E-invariant. Note that for unattributed graphs, we can define $g_X(\cdot,\cdot) = \emptyset$ as the null attribute, which is invariant to any environment by construction. For graphs with attributed vertices, $g_X(\cdot,\cdot)$ being invariant to E^{tr} and E^{te} means that for any two environments $e \in \text{supp}(E^{\text{tr}}), e^\dagger \in \text{supp}(E^{\text{te}}), g_X(e,\cdot) = g_X(e^\dagger,\cdot)$.

Theorem 1 shows that for $k \ll \min(N^{\text{tr}}, N^{\text{te}})$, the representations $\Gamma_{\text{1-hot}}(\cdot)$ of two possibly different-sized graphs with the same W are nearly identical, indicating $\Gamma_{\text{1-hot}}(\mathcal{G}^*_{N^*})$ is an approximately E-invariant representation.

Theorem 1 also exposes a trade-off, however. If the observed graphs tend to be relatively small, the required k for approximately E-invariant representations can be small, and then the expressiveness of $\Gamma_{\text{1-hot}}(\cdot)$ gets compromised. That is, the ability of $\Gamma_{\text{1-hot}}(\mathcal{G}^*_{N^*})$ to extract information about W from $\mathcal{G}^*_{N^*}$ reduces as k decreases. Finally, this guarantees that for appropriate k, passing the representation $\Gamma_{\text{1-hot}}(\mathcal{G}^*_{N^*})$ to a downstream classifier provably approximates the classifier in Equation (2) of Proposition 1.

Note that when the vertex attributes are not invariant to the environment variable, $\Gamma_{\text{1-hot}}(\cdot)$ is not E-invariant and we can not extrapolate using $\Gamma_{\text{1-hot}}(\cdot)$. Thankfully, for the GNN-based graph representations $\Gamma_{\text{GNN}}(\mathcal{G}^*_{N^*})$ and $\Gamma_{\text{GNN}^+}(\mathcal{G}^*_{N^*})$ in Equations (6) and (7), respectively, the regularization penalty in Equation (8) pushes the graph representation to be more E-invariant, making it more likely to satisfy the conditions of E-invariance in Theorem 1. Equation (8) is inspired by the *asymmetry learning* procedure of Mouli & Ribeiro (2021), which induces symmetry priors in the neural network, which can be broken (making the neural network asymmetric) only when imposing the symmetry significantly increases the training loss.

To understand the effect of our *asymmetry learning* in regularizing towards topology, consider the attributed SBM example in Section 2. The environment operates by changing the distributions of attributes assigned within each block. If we are going to achieve E-invariance (and correctly predict cross-block edge probabilities in the test data (see Section 5.2)), we need graph representations that treat attributes assigned to the same block as equivalent. By regularizing the GNN-based graph representations towards focusing only on topology rather than vertex attributes, the regularization forces the GNN to treat all within-block vertex attributes as equivalent, and achieve an approximately E-invariant representation in this setting. And since treating the across-block vertex attributes as equivalent hurts the training loss in this setting, these will not be considered equivalent by the GNN.

4. Related Work

This section presents an overview of the related work. Due to space constraints, a more in-depth discussion with further references is given in Appendix E.

OOD extrapolation in graph classification and size extrapolation in GNNs. Our work ascertains a causal relationship between graphs and their target labels. We are unaware of existing work on this topic. Xu et al. (2021) is interested on a geometric (non-causal) definition of extrapolation for a class of graph algorithms. Hu et al. (2020) introduces a large graph dataset presenting significant challenges of OOD extrapolation, however, their shift is on the two-dimensional structural framework distribution of the molecules, and no causal model is provided. The parallel work of Yehudai et al. (2021) improves size extrapolation in GNNs using self-supervised and semi-supervised learning on both the training and test domain, which is orthogonal to our problem. Previous works also examine empirically the ability of graph neural networks to extrapolate in various applications, such as physics (Battaglia et al., 2016; Sanchez-Gonzalez et al., 2018), mathematical and abstract reasoning (Santoro et al., 2018; Saxton et al., 2019), and graph algorithms (Bello et al., 2017; Nowak et al., 2017; Battaglia et al., 2018; Joshi et al., 2020; Veličković et al., 2020; Tang et al., 2020). These works do not provide guarantees of test extrapolation performance, a causal model, or a proof that the tasks require extrapolation over different environments.

Causal reasoning and invariances. Recent efforts have brought counterfactual inference to machine learning models, including *Independence of causal mechanism (ICM)* methods (Bengio et al., 2020; Besserve et al., 2018; Johansson et al., 2016; Louizos et al., 2017; Parascandolo et al., 2018; Raj et al., 2020; Schölkopf, 2019), *Causal Discovery from Change (CDC)* methods (Tian & Pearl, 2001), and *representation disentanglement* methods (Bengio et al., 2020; Goudet et al., 2017; Locatello et al., 2019). Invariant risk minimization (IRM) (Arjovsky et al., 2019) is a type of ICM (Schölkopf, 2019). Risk Extrapolation (REx) (Krueger et al., 2021) optimizes by focusing on the training environments that have the largest impact on training.

Broadly, the above efforts look for representations (or mechanism descriptions) that are invariant across multiple environments observed in the training data. In our work, we are interested in techniques that can work with a single training environment and when the test support is not a subset of the train support — a common case in graph data. To the best of our knowledge, the only representation learning work considering single environment extrapolations is Mouli & Ribeiro (2021). However, none of these methods is specifically designed for graphs, and it is unclear how they can be efficiently adapted for graph tasks. Finally, we also note that domain adaptation techniques and recent work on domain-predictors (Chuang et al., 2020) aim to learn invariances that can be used for the predictions. However, these require access to test data during training, which is not our scenario.

Graph classification using induced homomorphisms. A related set of works looks at induced homomorphism densities as graph features for a kernel (Shervashidze et al., 2009; Yanardag & Vishwanathan, 2015; Wale et al., 2008). These methods can perform poorly in some tasks (Kriege et al., 2018). Recent work has also shown an interest in induced subgraphs, which are used to improve predictions of GNNs (Bouritsas et al., 2020) or treated as inputs for newly-proposed architectures (Toenshoff et al., 2021). None of these methods focus on invariant representations or extrapolations.

Expressiveness of graph representations. The expressiveness of a graph representation method is a measure of model family bias (Morris et al., 2019; Xu et al., 2019; Gärtner et al., 2003; Maron et al., 2019a; Murphy et al., 2019). That is, given enough training data, a neural network from a more expressive family can achieve smaller generalization error over the training distribution than a neural network from a less expressive family, assuming appropriate optimization. However, this power is a measure of generalization capability over the training distribution, not OOD extrapolation. Hence, the question of representation expressiveness is orthogonal to our work.

5. Empirical Results

This section is dedicated to the empirical evaluation of our theoretical claims, including the ability of the representations in Equations (4), (6) and (7) to extrapolate as predicted by Proposition 1 for tasks that abide by Definitions 1 and 2. Due to space constraints, our results are summarised here, while further details are relegated to Appendix F. Our code is also available[1].

We explore the extrapolation power of $\Gamma_{\text{1-hot}}$, Γ_{GIN} and Γ_{RPGIN} of Equations (4), (6) and (7) using the Graph Isomorphism Network (GIN) (Xu et al., 2019) as our base GNN model, and Relational Pooling GIN (RPGIN) (Murphy et al., 2019) as a more expressive GNN. The graph representations are then passed to a L-hidden layer feedforward neural network (MLP) with softmax outputs that give the predicted classes, $L \in \{0, 1\}$. As described in Section 3.1, we obtain induced homomorphism densities of *connected* graphs. For practical reasons, we focus only on densities of graphs of size *exactly* k, which is treated as a hyperparameter. Note that the number of parameters for our Γ_{GNN} and $\Gamma_{\text{GNN+}}$ does not depend on k (for $\Gamma_{\text{1-hot}}$ it does), and the forward pass on the k-sized graphs can be performed in parallel.

Baselines. Our baselines include the Graphlet Counting kernel (GC Kernel) (Shervashidze et al., 2009), which uses the $\Gamma_{\text{1-hot}}$ representation as input to a downstream classifier. We report $\Gamma_{\text{1-hot}}$ separately from GC Kernel since $\Gamma_{\text{1-hot}}$ differs from GC Kernel in that we add the same feedforward neural network (MLP) classifier used in the Γ_{GNN} model. We also include GIN (Xu et al., 2019), GCN (Kipf & Welling, 2017) and PNA (Corso et al., 2020), considering the sum, mean, and max READOUTs as proposed by Xu et al. (2021) for extrapolations (which we denote as *XU-READOUT* to not confuse with our READOUT_Γ). We also examine a more-expressive GNN, RPGIN (Murphy et al., 2019), and the WL Kernel (Shervashidze et al., 2011). We do not use the method of Yehudai et al. (2021) as a baseline since it is a covariate shift adaptation approach that requires samples from $\mathrm{P}(\mathcal{G}^{\text{te}}_{N^{\text{te}}})$, which are not available in our setting.

Experiments with single and multiple graph sizes in training. Our single-environment experiments consist of a single graph size in training, and different sizes in test (different from the training size). Whenever multiple environments are available in training —multiple environments implies different graph sizes—, we employ Invariant Risk Minimization (IRM), considering the penalty proposed by Arjovsky et al. (2019) for each environment (defined empirically as a range of training examples with similar graph sizes).

For each task, we report (a) *training* accuracy (b) *validation* accuracy, which are new examples sampled from $\mathrm{P}(Y, \mathcal{G}^{\text{tr}}_{N^{\text{tr}}})$; and (c) *extrapolation test* accuracy, which are new OOD examples sampled from $\mathrm{P}(Y, \mathcal{G}^{\text{te}}_{N^{\text{te}}})$. In our experiments we perform early stopping as per Hu et al. (2020).

5.1. Size extrapolation tasks for unattributed graphs

Schizophrenia task. We use the fMRI brain graph data on 71 schizophrenic patients and 74 controls for classifying individuals with schizophrenia (De Domenico et al., 2016). Vertices represent brain regions (voxels) with edges as functional connectivity. We process the graph differently between training and test data, where training graphs have exactly 264 vertices (a single environment) and control-group graphs in test have around 40% fewer vertices. We employ a 5-fold cross-validation for hyperparameter tuning.

Erdős-Rényi task. We simulate Erdős-Rényi graphs (Gilbert, 1959; Erdős & Rényi, 1959) as a simple graphon random graph model. The task is to classify the edge probability $p \in \{0.2, 0.5, 0.8\}$ of the generated graph. First we consider a single-environment version of the task, where we train and validate on graphs of size 80 and extrapolate to graphs with size 140 in test. We also consider another experiment with training/validation graph sizes uniformly selected from $\{70, 80\}$ (so we can use IRM), with the test data same as before (graphs of size 140 in test).

Results. Table 1 shows that all methods perform well in validation (generalization over the training distribution). How-

[1] `https://github.com/PurdueMINDS/size-invariant-GNNs`

Table 1. Extrapolation performance over *unattributed* graphs **shows clear advantage of our environment-invariant representations, with or without GNN, over standard methods or IRM in extrapolation test accuracy**. Table shows mean (standard deviation) accuracy. Bold emphasises the best test average. NA value indicates IRM is not applicable (when training data has a single graph size).

	Accuracy in Schizophrenia Task			Accuracy in Erdős-Rényi Task					
	Training has a single graph size			**Training has a single graph size**			**Training has two graph sizes**		
	Train [$P(Y, G^{\text{TR}}_{N^{\text{TR}}})$]	Val. [$P(Y, G^{\text{TR}}_{N^{\text{TR}}})$]	**Test** ($\uparrow$) [$P(Y, G^{\text{TE}}_{N^{\text{TE}}})$]	Train [$P(Y, G^{\text{TR}}_{N^{\text{TR}}})$]	Val. [$P(Y, G^{\text{TR}}_{N^{\text{TR}}})$]	**Test** ($\uparrow$) [$P(Y, G^{\text{TE}}_{N^{\text{TE}}})$]	Train [$P(Y, G^{\text{TR}}_{N^{\text{TR}}})$]	Val. [$P(Y, G^{\text{TR}}_{N^{\text{TR}}})$]	**Test** ($\uparrow$) [$P(Y, G^{\text{TE}}_{N^{\text{TE}}})$]
PNA	0.99 (0.00)	0.76 (0.08)	0.61 (0.08)	1.00 (0.00)	1.00 (0.00)	0.65 (0.12)	1.00 (0.00)	1.00 (0.00)	0.64 (0.12)
PNA (mean xu-readout)	0.99 (0.00)	0.77 (0.07)	0.53 (0.10)	1.00 (0.00)	1.00 (0.00)	0.62 (0.12)	1.00 (0.00)	1.00 (0.00)	0.51 (0.19)
PNA (max xu-readout)	0.99 (0.00)	0.75 (0.07)	0.42 (0.06)	1.00 (0.00)	1.00 (0.00)	0.59 (0.16)	0.99 (0.01)	1.00 (0.00)	0.57 (0.15)
PNA + IRM	NA	NA	NA	NA	NA	NA	1.00 (0.00)	1.00 (0.00)	0.65 (0.13)
GCN	0.74 (0.04)	0.74 (0.08)	0.55 (0.09)	0.99 (0.01)	1.00 (0.00)	0.88 (0.10)	0.98 (0.01)	1.00 (0.00)	0.87 (0.10)
GCN (mean xu-readout)	0.72 (0.04)	0.73 (0.08)	0.65 (0.08)	0.99 (0.01)	1.00 (0.00)	0.79 (0.15)	0.98 (0.02)	1.00 (0.00)	0.75 (0.20)
GCN (max xu-readout)	0.86 (0.07)	0.75 (0.07)	0.54 (0.06)	0.99 (0.01)	1.00 (0.00)	0.90 (0.07)	0.96 (0.04)	1.00 (0.00)	0.87 (0.09)
GCN + IRM	NA	NA	NA	NA	NA	NA	0.98 (0.02)	1.00 (0.00)	0.88 (0.08)
GIN	0.72 (0.02)	0.74 (0.05)	0.36 (0.09)	1.00 (0.00)	1.00 (0.00)	0.64 (0.12)	1.00 (0.00)	1.00 (0.00)	0.65 (0.12)
GIN (mean xu-readout)	0.78 (0.02)	0.72 (0.05)	0.43 (0.05)	1.00 (0.00)	1.00 (0.00)	0.63 (0.09)	1.00 (0.00)	1.00 (0.00)	0.61 (0.09)
GIN (max xu-readout)	0.85 (0.02)	0.72 (0.05)	0.35 (0.06)	0.99 (0.01)	1.00 (0.00)	0.65 (0.12)	1.00 (0.00)	1.00 (0.00)	0.65 (0.07)
GIN + IRM	NA	NA	NA	NA	NA	NA	1.00 (0.00)	1.00 (0.00)	0.66 (0.08)
RPGIN	0.70 (0.02)	0.74 (0.05)	0.37 (0.06)	1.00 (0.00)	1.00 (0.00)	0.61 (0.16)	1.00 (0.00)	1.00 (0.00)	0.60 (0.16)
WL Kernel	1.00 (0.00)	0.63 (0.07)	0.40 (0.00)	1.00 (0.00)	1.00 (0.00)	0.01 (0.00)	1.00 (0.00)	1.00 (0.00)	0.30 (0.00)
GC Kernel	0.61 (0.00)	0.61 (0.06)	0.60 (0.00)	1.00 (0.00)	1.00 (0.00)	**1.00 (0.00)**	1.00 (0.00)	1.00 (0.00)	**1.00 (0.00)**
$\Gamma_{\text{1-hot}}$	0.71 (0.01)	0.72 (0.05)	**0.72 (0.04)**	1.00 (0.00)	1.00 (0.00)	**1.00 (0.00)**	1.00 (0.00)	1.00 (0.00)	**1.00 (0.00)**
Γ_{GIN}	0.75 (0.05)	0.70 (0.04)	**0.68 (0.07)**	1.00 (0.00)	1.00 (0.00)	**1.00 (0.00)**	1.00 (0.00)	1.00 (0.00)	**1.00 (0.00)**
Γ_{RPGIN}	0.69 (0.01)	0.71 (0.06)	**0.71 (0.03)**	1.00 (0.00)	1.00 (0.00)	**1.00 (0.00)**	1.00 (0.00)	1.00 (0.00)	**1.00 (0.00)**

Table 2. Extrapolation performance over *attributed* graphs **shows clear advantage of environment-invariant representations with GNNs and the attribute regularization in Equation (8)**. Table shows mean (standard deviation) accuracy. Bold emphasises the best test average. NA value indicates IRM is not applicable (when training data has a single graph size).

	Training has a single graph size 20			**Training has two graph sizes: 14 and 20**			**Training has two graph sizes: 20 and 30**		
	Train [$P(Y, G^{\text{TR}}_{N^{\text{TR}}})$]	Val. [$P(Y, G^{\text{TR}}_{N^{\text{TR}}})$]	**Test** ($\uparrow$) [$P(Y, G^{\text{TE}}_{N^{\text{TE}}})$]	Train [$P(Y, G^{\text{TR}}_{N^{\text{TR}}})$]	Val. [$P(Y, G^{\text{TR}}_{N^{\text{TR}}})$]	**Test** ($\uparrow$) [$P(Y, G^{\text{TE}}_{N^{\text{TE}}})$]	Train [$P(Y, G^{\text{TR}}_{N^{\text{TR}}})$]	Val. [$P(Y, G^{\text{TR}}_{N^{\text{TR}}})$]	**Test** ($\uparrow$) [$P(Y, G^{\text{TE}}_{N^{\text{TE}}})$]
PNA	1.00 (0.00)	1.00 (0.00)	0.65 (0.10)	0.96 (0.06)	0.94 (0.03)	0.57 (0.19)	0.99 (0.01)	1.00 (0.00)	0.69 (0.19)
PNA (mean xu-readout)	1.00 (0.00)	1.00 (0.00)	0.86 (0.13)	0.97 (0.02)	0.95 (0.02)	0.64 (0.11)	0.99 (0.01)	1.00 (0.00)	0.70 (0.15)
PNA (max xu-readout)	0.99 (0.01)	0.97 (0.02)	0.83 (0.13)	0.94 (0.04)	0.93 (0.03)	0.80 (0.12)	0.95 (0.05)	0.95 (0.05)	0.80 (0.15)
PNA + IRM	NA	NA	NA	0.95 (0.05)	0.94 (0.03)	0.58 (0.19)	0.99 (0.01)	1.00 (0.00)	0.70 (0.20)
GCN	0.99 (0.01)	0.98 (0.02)	0.62 (0.09)	0.95 (0.02)	0.96 (0.02)	0.55 (0.17)	1.00 (0.00)	1.00 (0.00)	0.73 (0.17)
GCN (mean xu-readout)	0.94 (0.03)	0.99 (0.01)	0.61 (0.12)	0.93 (0.05)	0.94 (0.02)	0.69 (0.20)	1.00 (0.00)	1.00 (0.00)	0.84 (0.13)
GCN (max xu-readout)	0.99 (0.01)	1.00 (0.00)	0.76 (0.07)	0.95 (0.04)	0.98 (0.02)	0.61 (0.17)	0.98 (0.02)	1.00 (0.00)	0.70 (0.20)
GCN + IRM	NA	NA	NA	0.93 (0.05)	0.97 (0.03)	0.65 (0.19)	1.00 (0.00)	1.00 (0.00)	0.84 (0.17)
GIN	0.97 (0.02)	1.00 (0.00)	0.64 (0.17)	0.95 (0.03)	0.96 (0.04)	0.66 (0.20)	0.98 (0.02)	1.00 (0.00)	0.74 (0.19)
GIN (mean xu-readout)	1.00 (0.00)	1.00 (0.00)	0.85 (0.14)	0.97 (0.01)	0.99 (0.01)	0.75 (0.18)	0.99 (0.01)	1.00 (0.00)	0.80 (0.15)
GIN (max xu-readout)	0.95 (0.02)	0.97 (0.03)	0.67 (0.18)	0.93 (0.06)	0.94 (0.03)	0.67 (0.17)	0.99 (0.01)	1.00 (0.00)	0.69 (0.15)
GIN + IRM	NA	NA	NA	0.95 (0.03)	0.97 (0.04)	0.64 (0.19)	0.98 (0.02)	1.00 (0.00)	0.75 (0.19)
RPGIN	0.98 (0.02)	1.00 (0.00)	0.49 (0.15)	0.96 (0.03)	0.99 (0.01)	0.54 (0.12)	0.99 (0.01)	1.00 (0.00)	0.50 (0.13)
WL Kernel	1.00 (0.00)	0.95 (0.00)	0.57 (0.00)	0.99 (0.00)	0.90 (0.00)	0.62 (0.00)	1.00 (0.00)	1.00 (0.00)	0.57 (0.00)
GC Kernel	1.00 (0.00)	0.90 (0.00)	0.43 (0.00)	1.00 (0.00)	0.80 (0.00)	0.43 (0.00)	0.99 (0.00)	0.90 (0.00)	0.43 (0.00)
$\Gamma_{\text{1-hot}}$	1.00 (0.00)	0.90 (0.00)	0.50 (0.07)	0.97 (0.03)	0.85 (0.05)	0.50 (0.07)	0.98 (0.00)	0.96 (0.02)	0.45 (0.05)
Γ_{GIN}	1.00 (0.00)	1.00 (0.00)	**0.98 (0.02)**	0.96 (0.02)	0.95 (0.01)	**0.95 (0.06)**	1.00 (0.00)	1.00 (0.00)	0.88 (0.12)
Γ_{RPGIN}	1.00 (0.00)	1.00 (0.00)	**1.00 (0.00)**	0.97 (0.03)	0.95 (0.02)	**0.95 (0.05)**	1.00 (0.00)	1.00 (0.00)	**0.93 (0.05)**

ever, only $\Gamma_{\text{1-hot}}$ (GC Kernel and our simple classifier), Γ_{GIN}, Γ_{RPGIN} are able to extrapolate, while displaying very similar —often identical— accuracies in validation (sampled from $\mathrm{P}(\mathcal{G}^{\text{tr}}_{N^{\text{tr}}})$) and test (sampled from $\mathrm{P}(\mathcal{G}^{\text{te}}_{N^{\text{te}}})$) in all experiments, as predicted by combining the theoretical results in Proposition 1 and Theorem 1. Using IRM in the Erdős-Rényi task shows no improvement over not using IRM in the multi-environment setting.

5.2. Size/attribute extrapolation for attributed graphs

We now define a Stochastic Block Model (SBM) task with vertex attributes. The SBM has two blocks. Our goal is to classify the cross-block edge probability $\boldsymbol{P}_{1,2} = \boldsymbol{P}_{2,1} \in \{0.1, 0.3\}$ of a sampled graph. Vertex attribute distributions depend on the blocks. In block 1 vertices are randomly assigned red and blue attributes, while in block 2 vertices are randomly assigned green and yellow attributes (see **SBM with vertex attributes** in Section 2).

The change in environments between training and test introduces a joint attribute-and-size distribution shift: In training, the vertices are 90% red (resp. green) and 10% blue (resp. yellow) in block 1 (resp. block 2). While in test, the distribution is flipped and vertices are 10% red (resp. green) and 90% blue (resp. yellow) in block 1 (resp. block 2). We consider three scenarios, with the same test data made of graphs of size 40: (a) A single-environment case, where all training graphs have size 20; (b) A multi-environment case, where training graphs have sizes 14 and 20; (c) A multi-environment case, where training graphs have sizes 20 and 30. These differences in training data will check whether having graphs of sizes closer to the test graph sizes improves the performance of traditional graph representation methods.

Results. Table 2 shows how traditional graph representations and $\Gamma_{\text{1-hot}}$ (both GC Kernel and our neural classifier) tap into the easy correlation between Y and the density of red and green vertex attributes in the training graphs, while Γ_{GIN} and Γ_{RPGIN}, with their attribute regularization (Equation (8)), are approximately E-invariant, resulting in higher test accuracy that more closely matches their validation accuracy. Moreover, applying IRM has no beneficial impact, while adding larger graphs in training (closer to test graph sizes) increases the extrapolation accuracy of most methods.

5.3. Experiments with real-world datasets that violate our causal model

Finally, we test our E-invariant representations on datasets that violate Definitions 1 and 2 and the conditions of Theorem 1. We consider four vertex-attributed datasets (NCI1, NCI109, DD, PROTEINS) from Morris et al. (2020), and split the data as proposed by Yehudai et al. (2021). As

Table 3. Extrapolation performance over real-world graph datasets with OOD tasks violating Definitions 1 and 2 and conditions of Theorem 1. Always one of our E-invariant representations Γ_{GIN} and Γ_{RPGIN} is amongst the top 4 best methods in all datasets except NCI109. Table shows mean (standard deviation) Matthews correlation coefficient (MCC) of the classifiers over the OOD test data. Bold emphasises the top-4 models (in average MCC) for each dataset.

DATASETS	NCI1	NCI109	PROTEINS	DD
RANDOM	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
PNA	0.21 (0.06)	**0.24 (0.06)**	**0.26 (0.08)**	0.24 (0.10)
PNA (MEAN XU-READOUT)	0.12 (0.05)	0.21 (0.04)	0.25 (0.06)	**0.29 (0.08)**
PNA (MAX XU-READOUT)	0.16 (0.05)	0.18 (0.07)	0.20 (0.05)	0.12 (0.14)
PNA + IRM	0.21 (0.07)	**0.27 (0.08)**	**0.26 (0.10)**	**0.26 (0.08)**
GCN	0.20 (0.06)	0.15 (0.06)	0.21 (0.09)	0.23 (0.05)
GCN (MEAN XU-READOUT)	0.20 (0.04)	0.15 (0.09)	0.23 (0.07)	0.19 (0.06)
GCN (MAX XU-READOUT)	0.20 (0.04)	0.19 (0.07)	0.20 (0.14)	0.09 (0.08)
GCN + IRM	0.12 (0.05)	**0.22 (0.06)**	0.20 (0.07)	0.23 (0.07)
GIN	**0.25 (0.06)**	0.18 (0.05)	0.23 (0.05)	0.25 (0.09)
GIN (MEAN XU-READOUT)	0.16 (0.05)	0.14 (0.05)	0.24 (0.05)	**0.27 (0.12)**
GIN (MAX XU-READOUT)	0.15 (0.08)	0.18 (0.08)	**0.28 (0.11)**	0.19 (0.07)
GIN + IRM	0.18 (0.08)	0.16 (0.04)	**0.26 (0.06)**	0.21 (0.09)
RPGIN	0.15 (0.04)	0.19 (0.05)	0.24 (0.09)	0.22 (0.09)
WL KERNEL	**0.39 (0.00)**	0.21 (0.00)	0.00 (0.00)	0.00 (0.00)
GC KERNEL	0.02 (0.00)	0.01 (0.00)	**0.29 (0.00)**	0.00 (0.00)
$\Gamma_{\text{1-HOT}}$	0.17 (0.08)	**0.25 (0.06)**	0.12 (0.09)	0.23 (0.08)
Γ_{GIN}	**0.24 (0.04)**	0.18 (0.04)	**0.29 (0.11)**	**0.28 (0.06)**
Γ_{RPGIN}	**0.26 (0.05)**	0.20 (0.04)	0.25 (0.12)	0.20 (0.05)

mentioned earlier, Yehudai et al. (2021) is not part of our baselines since it requires samples from the test distribution $\mathrm{P}(\mathcal{G}^{\text{te}}_{N^{\text{te}}})$.

Training and test data are created as follows: Graphs with sizes smaller than the 50-th percentile are assigned to training, while graphs with sizes larger than the 90-th percentile are assigned to test. A validation set for hyperparameter tuning consists of 10% held out examples from training.

Results. Table 3 shows the test results using the Matthews correlation coefficient (MCC) — MCC was chosen due to significant class imbalances in the OOD shift of our test data, see Appendix F for more details. We observe that always one of our E-invariant representations Γ_{GIN} and Γ_{RPGIN} is amongst the top 4 best methods in all datasets except NCI109. We also note that the WL KERNEL performs really well at NCI1 and very poorly (random) on PROTEINS and DD, showcasing the importance of consistency across datasets.

Comments on Table 3. Counterfactual-driven extrapolations have their representation methods tailored to a specific extrapolation mechanism. Unlike in-distribution tasks (and covariate shift adaptation tasks, where one sees test distribution examples of the input graphs), counterfactual-driven extrapolations rely on being robust to the distribution-shift mechanism given by the causal model. Hence, it is expected that the causal extrapolation mechanism that works for a molecular task may not work as well for a social network (unless they share a universal graph-formation mechanism). The schizophrenia task (Section 5.1) has the same mechanism as our causal model (hence, good performance). Further research may show that every single dataset in this subsection has its own distinct extrapolation mechanism. We think that although these datasets violate our assumptions, this subsection is important (and we hope will be copied by future work) to show which datasets may need different extrapolation mechanisms.

6. Conclusions

In this work we looked at the task of out-of-distribution (OOD) graph classification, where train and test data have different distributions. By introducing a structural causal model inspired by graphon models (Lovász & Szegedy, 2006), we defined a representation that is approximately invariant to the train/test distribution changes of our causal model, empirically showing its benefits on both synthetic and real-world datasets against standard graph classification baselines. Finally, our work contributed a blueprint for defining graph extrapolation tasks through causal models.

Acknowledgements

This work was funded in part by the National Science Foundation (NSF) awards CAREER IIS-1943364 and CCF-1918483, the Frederick N. Andrews Fellowship, and the Wabash Heartland Innovation Network. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors. We would like to thank our reviewers, who gave excellent suggestions to improve the paper. Further, we would like to thank Ryan Murphy for many insightful discussions, and Mayank Kakodkar and Carlos H. C. Teixeira for their invaluable help with the subgraph function estimation.

References

Abuoda, G., Morales, G. D. F., and Aboulnaga, A. Link prediction via higher-order motif features. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 412–429. Springer, 2019.

Ahmed, N. K., Willke, T. L., and Rossi, R. A. Estimation of local subgraph counts. In *2016 IEEE International Conference on Big Data (Big Data)*, pp. 586–595. IEEE, 2016.

Airoldi, E. M., Costa, T. B., and Chan, S. H. Stochastic blockmodel approximation of a graphon: Theory and consistent estimation. In *Advances in Neural Information Processing Systems*, pp. 692–700, 2013.

Aldous, D. J. Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11(4):581–598, 1981.

Alon, U. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6):450–461, 2007.

Anonymous. Incremental learning on growing graphs. In *Submitted to International Conference on Learning Representations*, 2021. under review.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Armstrong, J. S., Collopy, F., and Yokum, J. T. Decomposition by causal forces: a procedure for forecasting complex time series. *International Journal of forecasting*, 21(1):25–36, 2005.

Arvind, V., Fuhlbrück, F., Köbler, J., and Verbitsky, O. On weisfeiler-leman invariance: subgraph counts and related graph properties. *Journal of Computer and System Sciences*, 2020.

Atwood, J. and Towsley, D. Diffusion-convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1993–2001, 2016.

Balke, A. and Pearl, J. Probabilistic evaluation of counterfactual queries. In *Proceedings of AAAI*, 1994.

Bascompte, J. and Melián, C. J. Simple trophic modules for complex food webs. *Ecology*, 86(11):2868–2873, 2005.

Battaglia, P., Pascanu, R., Lai, M., Rezende, D. J., et al. Interaction networks for learning about objects, relations and physics. In *Advances in neural information processing systems*, pp. 4502–4510, 2016.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Belkin, M. and Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, pp. 585–591, 2002.

Bello, I., Pham, H., Le, Q. V., Norouzi, M., and Bengio, S. Neural combinatorial optimization with reinforcement learning. In *International Conference on Learning Representations*, 2017.

Bengio, Y., Deleu, T., Rahaman, N., Ke, N. R., Lachapelle, S., Bilaniuk, O., Goyal, A., and Pal, C. A meta-transfer objective for learning to disentangle causal mechanisms. In *International Conference on Learning Representations*, 2020.

Benson, A. R., Gleich, D. F., and Leskovec, J. Higher-order organization of complex networks. *Science*, 353(6295): 163–166, 2016.

Besserve, M., Shajarisales, N., Schölkopf, B., and Janzing, D. Group invariance principles for causal generative models. In *International Conference on Artificial Intelligence and Statistics*, pp. 557–565, 2018.

Borgwardt, K. M. and Kriegel, H.-P. Shortest-path kernels on graphs. In *Fifth IEEE international conference on data mining (ICDM'05)*, pp. 8–pp. IEEE, 2005.

Borgwardt, K. M., Ong, C. S., Schönauer, S., Vishwanathan, S., Smola, A. J., and Kriegel, H.-P. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1): i47–i56, 2005.

Bouritsas, G., Frasca, F., Zafeiriou, S., and Bronstein, M. M. Improving graph neural network expressivity via subgraph isomorphism counting. *arXiv preprint arXiv:2006.09252*, 2020.

Bressan, M., Chierichetti, F., Kumar, R., Leucci, S., and Panconesi, A. Counting graphlets: Space vs time. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM'17)*, pp. 557–566. ACM, 2017.

Chami, I., Ying, Z., Ré, C., and Leskovec, J. Hyperbolic graph convolutional neural networks. In *Advances in neural information processing systems*, pp. 4868–4879, 2019.

Chami, I., Abu-El-Haija, S., Perozzi, B., Ré, C., and Murphy, K. Machine learning on graphs: A model and comprehensive taxonomy. *arXiv preprint arXiv:2005.03675*, 2020.

Chen, L., Qu, X., Cao, M., Zhou, Y., Li, W., Liang, B., Li, W., He, W., Feng, C., Jia, X., et al. Identification of breast cancer patients based on human signaling network motifs. *Scientific reports*, 3:3368, 2013.

Chen, X. and Lui, J. C. Mining graphlet counts in online social networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(4):1–38, 2018.

Chen, X., Li, Y., Wang, P., and Lui, J. A general framework for estimating graphlet statistics via random walk. *VLDB Endowment*, 2016.

Chen, Z., Chen, L., Villar, S., and Bruna, J. Can graph neural networks count substructures? In *Advances in Neural Information Processing Systems*, 2020.

Chuang, C.-Y., Torralba, A., and Jegelka, S. Estimating generalization under distribution shifts via domain-invariant representations, 2020.

Corso, G., Cavalleri, L., Beaini, D., Lio, P., and Veličković, P. Principal neighbourhood aggregation for graph nets.

In *Advances in Neural Information Processing Systems*, 2020.

D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M. D., et al. Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*, 2020.

De Domenico, M., Sasai, S., and Arenas, A. Mapping multiplex hubs in human functional brain networks. *Frontiers in neuroscience*, 10:326, 2016.

Dey, A. K., Gel, Y. R., and Poor, H. V. What network motifs tell us about resilience and reliability of complex networks. *Proceedings of the National Academy of Sciences*, 116(39):19368–19373, 2019.

Diaconis, P. and Freedman, D. On the statistics of vision: the julesz conjecture. *Journal of Mathematical Psychology*, 24(2):112–138, 1981.

Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pp. 2224–2232, 2015.

Eckles, D., Karrer, B., and Ugander, J. Design and analysis of experiments in networks: Reducing bias from interference. *Journal of Causal Inference*, 5(1), 2016.

Erdős, P. and Rényi, A. On random graphs i. *Publ. math. debrecen*, 6(290-297):18, 1959.

Fey, M. and Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Gao, J. and Ribeiro, B. On the equivalence between temporal and static graph representations for observational predictions. *arXiv preprint arXiv:2103.07016*, 2021.

Garg, V. K., Jegelka, S., and Jaakkola, T. Generalization and representational limits of graph neural networks. In *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2020.

Gärtner, T., Flach, P., and Wrobel, S. On graph kernels: Hardness results and efficient alternatives. In *Learning theory and kernel machines*, pp. 129–143. Springer, 2003.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Gilbert, E. N. Random graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144, 1959.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

Goudet, O., Kalainathan, D., Caillou, P., Guyon, I., Lopez-Paz, D., and Sebag, M. Causal generative neural networks. *arXiv preprint arXiv:1711.08936*, 2017.

Grover, A. and Leskovec, J. node2vec: Scalable feature learning for networks. In *Proc. of KDD*, pp. 855–864. ACM, 2016.

Haffner, P. Escaping the convex hull with extrapolated vector machines. In *Advances in Neural Information Processing Systems*, pp. 753–760, 2002.

Hagberg, A. A., Schult, D. A., and Swart, P. J. Exploring network structure, dynamics, and function using networkx. In Varoquaux, G., Vaught, T., and Millman, J. (eds.), *Proceedings of the 7th Python in Science Conference*, pp. 11 – 15, Pasadena, CA USA, 2008.

Hagmann, P., Kurant, M., Gigandet, X., Thiran, P., Wedeen, V. J., Meuli, R., and Thiran, J.-P. Mapping human whole-brain structural networks with diffusion mri. *PloS one*, 2 (7):e597, 2007.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pp. 1024–1034, 2017.

Hamilton, W. L. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159, 2020.

Hancock, J. T. and Khoshgoftaar, T. M. Survey on categorical data for neural networks. *Journal of Big Data*, 7:1–41, 2020.

Hastie, T., Tibshirani, R., and Friedman, J. *The elements of statistical learning*, volume 1. Springer series in statistics, 2012.

Hemminger, R. L. On reconstructing a graph. *Proceedings of the American Mathematical Society*, 20(1):185–187, 1969.

Hernández-García, A. and König, P. Data augmentation instead of explicit regularization. *arXiv preprint arXiv:1806.03852*, 2018.

Hoover, D. N. Relations on probability spaces and arrays of random variables. *Technical Report, Institute for Advanced Study, Princeton, NJ*, 2, 1979.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, 2020.

Johansson, F., Shalit, U., and Sontag, D. Learning representations for counterfactual inference. In *International conference on machine learning*, pp. 3020–3029, 2016.

Joshi, C. K., Cappart, Q., Rousseau, L.-M., Laurent, T., and Bresson, X. Learning tsp requires rethinking generalization. *arXiv preprint arXiv:2006.07054*, 2020.

Kallenberg, O. *Probabilistic symmetries and invariance principles*. Springer Science & Business Media, 2006.

Kashima, H., Tsuda, K., and Inokuchi, A. Marginalized kernels between labeled graphs. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pp. 321–328, 2003.

Kazemi, S. M., Goel, R., Jain, K., Kobyzev, I., Sethi, A., Forsyth, P., and Poupart, P. Representation learning for dynamic graphs: A survey. *Journal of Machine Learning Research*, 21(70):1–73, 2020.

Kelly, P. J. et al. A congruence theorem for trees. *Pacific Journal of Mathematics*, 7(1):961–968, 1957.

King, G. and Zeng, L. The dangers of extreme counterfactuals. *Political Analysis*, 14(2):131–159, 2006.

Kipf, T. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Kipf, T. N. and Welling, M. Variational graph auto-encoders. *NIPS Workshop on Bayesian Deep Learning*, 2016.

Klicpera, J., Groß, J., and Günnemann, S. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2020.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Beery, S., et al. Wilds: A benchmark of in-the-wild distribution shifts. *arXiv preprint arXiv:2012.07421*, 2020.

Kriege, N. M., Morris, C., Rey, A., and Sohler, C. A property testing framework for the theoretical expressivity of graph kernels. In *IJCAI*, pp. 2348–2354, 2018.

Kriege, N. M., Johansson, F. D., and Morris, C. A survey on graph kernels. *Applied Network Science*, 5(1):1–42, 2020.

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Priol, R. L., and Courville, A. Out-of-distribution generalization via risk extrapolation (rex), 2021.

Kumar, S., Zhang, X., and Leskovec, J. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '19, pp. 1269–1278, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016.

Lample, G. and Charton, F. Deep learning for symbolic mathematics. In *International Conference on Learning Representations*, 2020.

Lee, J. B., Rossi, R. A., Kong, X., Kim, S., Koh, E., and Rao, A. Higher-order graph convolutional networks. *arXiv preprint arXiv:1809.07697*, 2018.

Li, X., Wei, W., Feng, X., Liu, X., and Zheng, Z. Representation learning of graphs using graph convolutional multilayer networks based on motifs. *arXiv preprint arXiv:2007.15838*, 2020.

Liu, Q., Nickel, M., and Kiela, D. Hyperbolic graph neural networks. In *Advances in Neural Information Processing Systems*, pp. 8230–8241, 2019.

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pp. 4114–4124, 2019.

Louizos, C., Shalit, U., Mooij, J. M., Sontag, D., Zemel, R., and Welling, M. Causal effect inference with deep latent-variable models. In *Advances in Neural Information Processing Systems*, pp. 6446–6456, 2017.

Lovász, L. *Large networks and graph limits*, volume 60. American Mathematical Soc., 2012.

Lovász, L. and Szegedy, B. Limits of dense graph sequences. *Journal of Combinatorial Theory, Series B*, 96(6):933–957, 2006.

Mangan, S. and Alon, U. Structure and function of the feed-forward loop network motif. *Proceedings of the National Academy of Sciences*, 100(21):11980–11985, 2003.

Maron, H., Ben-Hamu, H., Serviansky, H., and Lipman, Y. Provably powerful graph networks. In *Advances in Neural Information Processing Systems*, pp. 2156–2167, 2019a.

Maron, H., Ben-Hamu, H., Shamir, N., and Lipman, Y. Invariant and equivariant graph networks. In *International Conference on Learning Representations*, 2019b.

McKay, B. D. Small graphs are reconstructible. *Australasian Journal of Combinatorics*, 15:123–126, 1997.

Meng, C., Mouli, S. C., Ribeiro, B., and Neville, J. Subgraph pattern neural networks for high-order graph evolution prediction. In *AAAI*, pp. 3778–3787, 2018.

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. Network motifs: simple building blocks of complex networks. *Science*, 298(5594): 824–827, 2002.

Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4602–4609, 2019.

Morris, C., Kriege, N. M., Bause, F., Kersting, K., Mutzel, P., and Neumann, M. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020.

Mouli, S. C. and Ribeiro, B. Neural networks for learning counterfactual g-invariances from single environments. *ICLR*, 2021.

Munch, E. A user's guide to topological data analysis. *Journal of Learning Analytics*, 4(2):47–61, 2017.

Murphy, R., Srinivasan, B., Rao, V., and Ribeiro, B. Relational pooling for graph representations. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Neyman, J. Sur les applications de la theorie des probabilites aux experiences agricoles: essai des principes (masters thesis); justification of applications of the calculus of probabilities to the solutions of certain questions in agricultural experimentation. excerpts english translation (reprinted). *Stat Sci*, 5:463–472, 1923.

Niepert, M., Ahmed, M., and Kutzkov, K. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pp. 2014–2023, 2016.

Nowak, A., Villar, S., Bandeira, A. S., and Bruna, J. A note on learning algorithms for quadratic assignment with graph neural networks. In *Proceeding of the 34th International Conference on Machine Learning (ICML)*, volume 1050, pp. 22, 2017.

Orbanz, P. and Roy, D. M. Bayesian models of graphs, arrays and other exchangeable random structures. *IEEE transactions on pattern analysis and machine intelligence*, 37(2):437–461, 2014.

Ou, M., Cui, P., Pei, J., Zhang, Z., and Zhu, W. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1105–1114, 2016.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519, 2017.

Parascandolo, G., Kilbertus, N., Rojas-Carulla, M., and Schölkopf, B. Learning independent causal mechanisms. In *International Conference on Machine Learning*, pp. 4036–4044. PMLR, 2018.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Pearl, J. *Causality*. Cambridge university press, 2009.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Perozzi, B., Al-Rfou, R., and Skiena, S. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, 2014.

Pinar, A., Seshadhri, C., and Vishal, V. Escape: Efficiently counting all 5-vertex subgraphs. In *Proceedings of the 26th International Conference on World Wide Web*, pp. 1431–1440, 2017.

Pržulj, N. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183, 2007.

Qiu, J., Dong, Y., Ma, H., Li, J., Wang, K., and Tang, J. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 459–467, 2018.

Raj, A., Bauer, S., Soleymani, A., Besserve, M., and Schölkopf, B. Causal feature selection via orthogonal search. *arXiv preprint arXiv:2007.02938*, 2020.

Rieck, B., Bock, C., and Borgwardt, K. A persistent weisfeiler-lehman procedure for graph classification. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5448–5458, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Rosenfeld, E., Ravikumar, P., and Risteski, A. The risks of invariant risk minimization. *arXiv preprint arXiv:2010.05761*, 2020.

Rossi, R. A., Ahmed, N. K., and Koh, E. Higher-order network representation learning. In *Companion Proceedings of the The Web Conference 2018*, pp. 3–4, 2018.

Rossi, R. A., Ahmed, N. K., Carranza, A., Arbour, D., Rao, A., Kim, S., and Koh, E. Heterogeneous network motifs. *arXiv preprint arXiv:1901.10026*, 2019.

Rubin, D. B. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

Sanchez-Gonzalez, A., Heess, N., Springenberg, J. T., Merel, J., Riedmiller, M. A., Hadsell, R., and Battaglia, P. Graph networks as learnable physics engines for inference and control. In *International Conference on Machine Learning*, 2018.

Santoro, A., Hill, F., Barrett, D., Morcos, A., and Lillicrap, T. Measuring abstract reasoning in neural networks. In *International Conference on Machine Learning*, pp. 4477–4486, 2018.

Sato, R. A survey on the expressive power of graph neural networks. *arXiv preprint arXiv:2003.04078*, 2020.

Saxton, D., Grefenstette, E., Hill, F., and Kohli, P. Analysing mathematical reasoning abilities of neural models. In *International Conference on Learning Representations*, 2019.

Schölkopf, B. Causality for machine learning. *arXiv preprint arXiv:1911.10500*, 2019.

Scott Armstrong, J. and Collopy, F. Causal forces: Structuring knowledge for time-series extrapolation. *Journal of Forecasting*, 12(2):103–115, 1993.

Shajarisales, N., Janzing, D., Schoelkopf, B., and Besserve, M. Telling cause from effect in deterministic linear dynamical systems. In *International Conference on Machine Learning*, pp. 285–294. PMLR, 2015.

Shen-Orr, S. S., Milo, R., Mangan, S., and Alon, U. Network motifs in the transcriptional regulation network of escherichia coli. *Nature genetics*, 31(1):64–68, 2002.

Shervashidze, N., Vishwanathan, S., Petri, T., Mehlhorn, K., and Borgwardt, K. Efficient graphlet kernels for large graph comparison. In *Artificial Intelligence and Statistics*, pp. 488–495, 2009.

Shervashidze, N., Schweitzer, P., Leeuwen, E. J. v., Mehlhorn, K., and Borgwardt, K. M. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(Sep):2539–2561, 2011.

Sitawarin, C., Bhagoji, A. N., Mosenia, A., Mittal, P., and Chiang, M. Rogue signs: Deceiving traffic sign recognition with malicious ads and logos. *CoRR*, abs/1801.02780, 2018.

Snijders, T. A. and Nowicki, K. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of classification*, 14(1):75–100, 1997.

Sporns, O. and Kötter, R. Motifs in brain networks. *PLoS biology*, 2(11):e369, 2004.

Stone, L. and Roberts, A. Competitive exclusion, or species aggregation? *Oecologia*, 91(3):419–424, 1992.

Stone, L., Simberloff, D., and Artzy-Randrup, Y. Network motifs and their origins. *PLoS computational biology*, 15 (4):e1006749, 2019.

Sugiyama, M., Krauledat, M., and Müller, K.-R. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5), 2007.

Sugiyama, M., Ghisu, M. E., Llinares-López, F., and Borgwardt, K. graphkernels: R and python packages for graph comparison. *Bioinformatics*, 34(3):530–532, 2017.

Sun, H., Dhingra, B., Zaheer, M., Mazaitis, K., Salakhutdinov, R., and Cohen, W. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4231–4242, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

Tang, H., Huang, Z., Gu, J., Lu, B.-L., and Su, H. Towards scale-invariant graph-related problem solving by iterative homogeneous gnns. In *Advances in Neural Information Processing Systems*, volume 33, pp. 15811–15822, 2020.

Teixeira, C. H., Cotta, L., Ribeiro, B., and Meira, W. Graph pattern mining and learning through user-defined relations. In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 1266–1271. IEEE, 2018.

Teru, K. K., Denis, E., and Hamilton, W. L. Inductive relation prediction by subgraph reasoning. In *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2020.

Tian, J. and Pearl, J. Causal discovery from changes. *UAI*, 2001.

Toenshoff, J., Ritzert, M., Wolf, H., and Grohe, M. Graph learning with 1d convolutions on random walks. *arXiv preprint arXiv:2102.08786*, 2021.

Tweedie, M. C. Inverse statistical variates. *Nature*, 155 (3937):453–453, 1945.

Ulam, S. M. A collection of mathematical problems. *Wiley, New York*, 29, 1960.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. *ICLR*, 2018.

Veličković, P., Ying, R., Padovano, M., Hadsell, R., and Blundell, C. Neural execution of graph algorithms. In *International Conference on Learning Representations (ICLR)*, 2020.

Vignac, C., Loukas, A., and Frossard, P. Building powerful and equivariant graph neural networks with structural message-passing. In *Advances in Neural Information Processing Systems*, 2020.

Wale, N., Watson, I. A., and Karypis, G. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*, 14 (3):347–375, 2008.

Wang, L., Zhao, H., Li, J., Xu, Y., Lan, Y., Yin, W., Liu, X., Yu, L., Lin, S., Du, M. Y., et al. Identifying functions and prognostic biomarkers of network motifs marked by diverse chromatin states in human cell lines. *Oncogene*, 39(3):677–689, 2020a.

Wang, P., Lui, J. C., Ribeiro, B., Towsley, D., Zhao, J., and Guan, X. Efficiently estimating motif statistics of large networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(2):1–27, 2014.

Wang, Y., Wang, W., Liang, Y., Cai, Y., and Hooi, B. Graphcrop: Subgraph cropping for graph classification. *arXiv preprint arXiv:2009.10564*, 2020b.

Wedeen, V. J., Hagmann, P., Tseng, W.-Y. I., Reese, T. G., and Weisskoff, R. M. Mapping complex tissue architecture with diffusion spectrum magnetic resonance imaging. *Magnetic resonance in medicine*, 54(6):1377–1386, 2005.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.-i., and Jegelka, S. Representation learning on graphs with jumping knowledge networks. volume 80 of *Proceedings of Machine Learning Research*, pp. 5453–5462, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

Xu, K., Zhang, M., Li, J., Du, S. S., Kawarabayashi, K.-I., and Jegelka, S. How neural networks extrapolate: From feedforward to graph neural networks. In *International Conference on Learning Representations*, 2021.

Yanardag, P. and Vishwanathan, S. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1365–1374. ACM, 2015.

Ye, W., Askarisichani, O., Jones, A., and Singh, A. Deepmap: Learning deep representations for graph classification. *arXiv preprint arXiv:2004.02131*, 2020.

Yehudai, G., Fetaya, E., Meirom, E., Chechik, G., and Maron, H. From local structures to size generalization in graph neural networks. *arXiv preprint arXiv:2010.08853*, 2021.

You, J., Ying, R., and Leskovec, J. Position-aware graph neural networks. volume 97 of *Proceedings of Machine Learning Research*, pp. 7134–7143, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Yu, W., Zheng, C., Cheng, W., Aggarwal, C. C., Song, D., Zong, B., Chen, H., and Wang, W. Learning deep network representations with adversarially regularized autoencoders. In *Proc. of AAAI*, pp. 2663–2671. ACM, 2018.

Zhang, M. and Chen, Y. Link prediction based on graph neural networks. In *Advances in Neural Information Processing Systems*, pp. 5165–5175, 2018.

Zhang, Z., Cui, P., and Zhu, W. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2020.