Causal Lifting and Link Prediction

Leonardo Cotta [*†1], Beatrice Bevilacqua [2], Nesreen Ahmed [3], and Bruno Ribeiro [2]

[1]Vector Institute
[2]Purdue University
[3]Intel Labs

July 28, 2023

Abstract

Existing causal models for link prediction assume an underlying set of inherent node factors —an innate characteristic defined at the node's birth— that governs the causal evolution of links in the graph. In some causal tasks, however, link formation is *path-dependent*: The outcome of link interventions depends on existing links. Unfortunately, these existing causal methods are not designed for path-dependent link formation, as the cascading functional dependencies between links (arising from *path dependence*) are either unidentifiable or require an impractical number of control variables. To overcome this, we develop the first causal model capable of dealing with path dependencies in link prediction.

In this work we introduce the concept of causal lifting, an invariance in causal models of independent interest that, on graphs, allows the identification of causal link prediction queries using limited interventional data. Further, we show how structural pairwise embeddings exhibit lower bias and correctly represent the task's causal structure, as opposed to existing node embeddings, *e.g.*, graph neural network node embeddings and matrix factorization. Finally, we validate our theoretical findings on three scenarios for causal link prediction tasks: knowledge base completion, covariance matrix estimation and consumer-product recommendations.

1 Introduction

Predicting links between entities via latent factors has captivated the scientific community ever since Charles Spearman published *The Abilities of Man* [67] in 1927, where he described the mathematical tools to uncover latent *common factors* of intelligence just by observing a subject i perform a task j successfully ($A_{ij} = 1$) or unsuccessfully ($A_{ij} = 0$) over n subjects and m tasks. Spearman's work started a revolution that gave us, among other things, matrix and tensor factorizations, Principal Component Analysis (PCA), and Independent Component Analysis (ICA). Simultaneously, *The Abilities of Man* also warned us about interpreting the factors of subject i as innate rather than acquired abilities. For instance, regarding Woolley and Fischer's observation that "boys are *enormously superior* [to girls] at [...] spatial relations" (*i.e.*, in how objects relate in space) [76], Spearman warns that "*evidence of this difference being really innate* [*rather than acquired*] *is still dubious*".

Today, we can describe Spearman's warning as being about two competing causal hypotheses that describe link formation between young children and their abilities. A *path-dependent hypothesis* where past links influence future links [46] and an *innate factors* hypothesis where link formation is just a manifestation of latent innate factors [75]. In Woolley and Fischer's experiments, both hypotheses are able to describe the data: Either boys are innately better than girls at spatial reasoning (innate factors hypothesis), or boys in 1914 just happened to have had more playtime with spatial tasks than girls, with each task further improving

*Work partially done at Purdue University and Intel Labs.
†Correspondence to: leonardo.cotta@vectorinstitute.ai

their skills (path-dependent hypothesis). If we were to hide the performance of girl i on task j (*i.e.*, hide A_{ij}), both hypotheses could give equally accurate predictive models for the missing association A_{ij}. Observing boys and girls perform these tasks over time does not disambiguate these two hypotheses, since inner factors can also evolve over time.

Under a causal task, however, incorrect innate factor assumptions may lead to incorrect predictions about the effect of interventions. Encouraging girls to perform spatial tasks in playtime (an intervention) could improve their spatial reasoning skills under the path-dependent hypothesis but not under an innate factors hypothesis. Similarly, performing an intervention to probe into whether a consumer will buy a computer keyboard (through an ad or a top search ranking) [39, 75], who then buys it, will lead to assuming that (a) maybe the consumer no longer needs (another) keyboard under a path-dependent model; or (b) the consumer will keep buying more keyboards (if we keep showing more ads and top search results) [37] under an innate factor hypothesis. Current causal models for link prediction operate under the assumption of innate factors [60, 75]. Path-dependent causal models are able to react to the evolution of the graph structure. Part of the reason for the absence of path-dependent models in the literature are the relational cross-dependencies of path dependency that are hard to address with existing causal methods.

The key insight of our work is observing that graph symmetries can encode the relationship between an evolving graph structure and a link formation process that reacts to it. To illustrate this, let us look at the social network example in Figure 1. Consider the graph formed by the observed friendships between users (black edges). We then intervene by recommending a friendship between users 1 and 8. Next, we observe a new friendship (link) appearing (green edge). Finally, we wonder what would have happened had we recommended a friendship between users 3 and 9 instead (counterfactual query). From the figure, we can see that both pairs of users are symmetric (isomorphic), *i.e.*, indistinguishable if we do not consider their identifiers (cf. Appendix A). Thus, answering the counterfactual query about 3 and 9 with the same outcome observed in 1 and 8 seems intuitively correct. However, without causal modeling assumptions it is not possible to answer counterfactual queries in general [4]. Hence, our work establishes sufficient causal modeling assumptions for this intuition to hold in practice, *i.e.*, for graph symmetries to govern the causal link formation process of a graph. Note how previous interventions in the graph formation process will be reflected in the structural symmetries, which itself governs the link formation process. As such, this can be a path-dependent model. Finally, we generalize this setting to a supervised learning task, where graph embedding models are trained to predict intervention outcomes from the observed graph structure. We show that graph embeddings capturing such symmetries, *i.e.*, assigning the same embeddings to symmetric node pairs, are the unbiased estimators capturing the underlying causal mechanisms of the task. Next, we formalize these notions and detail our contributions.

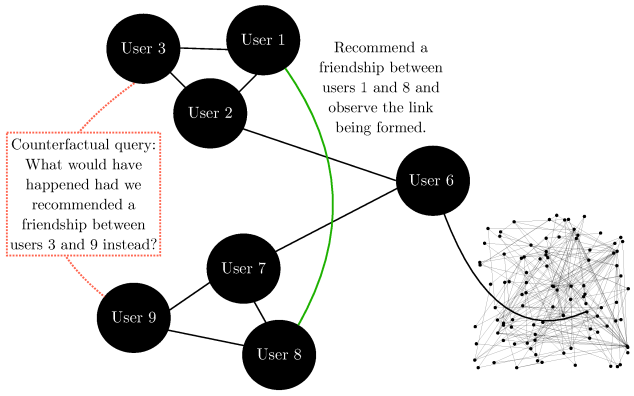

Figure 1: We illustrate the role of symmetry in a social network. Given that we observe the social graph formed by the black edges, we intervene and recommend a friendship between users 1 and 8. After observing that a new link is created between them, we ask whether a link would have been created had we recommended a friendship between users 3 and 9 instead.

Task setup. We consider first observing at time t_0 a (possibly static and attributed) graph $G^{(t_0)}$. Next, at a later time t_1 we are able to intervene in a pair of nodes $\mathcal{E}^{(t_1)}$ and observe whether the link appears in $G^{(t_1)}$. Finally, we are interested in answering counterfactual queries of the form: *What would have happened had we intervened in this other pair of nodes?* We outline the needed assumptions on the causal generating process $\mathcal{C}$ of $G^{(t_0)}$ such that we can intervene on more pairs and train a graph embedding model to answer the counterfactual queries. The causal model considered is (possibly) path-dependent, *i.e.*, the link formation in $G^{(t_1)}$ can depend on how $G^{(t_0)}$ was created, including previous interventions (before t_0) that we do not observe. Moreover, the graph embedding is trained in a supervised learning fashion using the observational data $G^{(t_0)}$ to predict the interventional data $G^{(t_1)}$, *i.e.*, the targets are the links and non-links in which we intervened.

Contributions. Our contributions are centered around using invariances to both i) define a set of sufficient causal modeling assumptions for causal identification and to ii) define the needed graph embedding for unbiased estimation of causal links. Regarding identification, we develop the concept of causal lifting, an invariance property of causal models that is able to serve link prediction under interventions in both path-dependent and innate factor models. Causal lifting serves more than a causal link prediction tool, it is also an experimental design tool and identification strategy for invariant data. The key insight of causal lifting is to identify causal quantities by assuming invariances in the causal mechanisms of the task, rather than relying on variable controls or covariate-based adjustments common in do-calculus and potential outcomes analyses. Further, on the estimation side, we show how structural pairwise embeddings, a type of graph embedding that incorporates the known causal invariances of the task, achieves lower bias and variance than node embedding methods. Finally, we validate our theoretical findings on four datasets under three different scenarios of causal link prediction tasks.

A Family of Causal Link Prediction Tasks

In our task, we observe graph data at some (pre-trial) time t_0 from an unknown (causal) graph generation process $\mathcal{C}$ with potential path dependencies. As such, future links and probes might depend on previous states of the graph. We denote the observed graph at pre-trial time t_0 by $G^{(t_0)}$, with corresponding adjacency matrix $a^{(t_0)}$, node set $V^{(t_0)}$ of size $n^{(t_0)} := |V^{(t_0)}|$ and edge set $E^{(t_0)}$. Without loss of generality, we refer to graphs at any other time points of interest $t \in \mathbb{N}$ using the same notation $(G^{(t)}, a^{(t)}, V^{(t)}, n^{(t)}, E^{(t)})$. Moreover, we use capital letters to denote the corresponding random variable of an observation, *e.g.*, $A^{(t_0)}$ is the random variable of the observed adjacency $a^{(t_0)}$. Finally, unless otherwise stated, our results consider an adjacency $a^{(t_0)} \in \mathbb{A}^{n^{(t_0)} \times n^{(t_0)}}$ with arbitrary finite domain $\mathbb{A}$. Thus, $a^{(t_0)}$ possibly contains information about time, node and edge features —which implies that $a^{(t_0)}$ is not necessarily a matrix (it can be a tensor representing this heterogeneous node adjacency).

After observing the graph $G^{(t_0)}$, we are often interested in *probing into a certain relation*. More specifically, we probe into the relation of $(I, J) \sim \mu(G^{(t_0)})$, where $p\big((I, J); G^{(t_0)}\big)$ is a distribution over the node pairs in $G^{(t_0)}$. We refer to $\mu(G^{(t_0)})$ as the probe policy, *i.e.*, it (stochastically) defines which relation from $G^{(t_0)}$ we wish to probe into. For instance, in online social networks we can probe into a friendship between two users by making a friendship recommendation. Note that the act of probing induces an intervention in the graph formation process, *i.e.*, despite of being friends two users might not add each other at that point of time without an explicit recommendation from the system. We will refer to interventions of this type, *i.e.*, probing into a relation, simply by **probes**. Finally, note that a probe can be seen as an experimental trial as well, *e.g.*, in the *recommendations as treatments* framework [39] products are seen as treatments that the recommender system chooses or not to administer to each of its users.

We consider the act of probing and its outcome happening simultaneously at a posterior time $t_1 > t_0$. That is, the difference in time between an intervention and its outcome is small enough that variables associated with other pairs are not impacted by the intervention. As mentioned, the probe is an intervention in the graph formation process. Hence, we define as $\mathcal{E}^{(t_1)}$ the random variable of the graph process we make interventions in. We can then define the random variable of the outcome of a probe in $(I, J) \sim \mu(G^{(t_0)})$ as

$$\underbrace{Y_{IJ}^{(t_1)}}_{\substack{\text{Outcome of}\\ \text{probe in } (I,J).}} := A_{\mathcal{E}^{(t_1)}}^{(t_1)}\underbrace{\left(\mathcal{E}^{(t_1)} = (I,J)\right)}_{\text{Probe in } (I,J).} \mid G^{(t_0)}, \tag{1}$$

where t_1 is both the time of the probe and when we see its effect (post-trial time), $A^{(t_1)}$ is the adjacency of the graph $G^{(t_1)}$ (after the probe), and $\mathcal{E}^{(t_1)}$ is the probe random variable. In our work we will use the potential outcomes notation [63] and Pearl's Causal Hierarchy [5] framework to describe causal tasks.

In real-world systems, we can actively probe and observe an outcome for Equation (1), *e.g.*, make a recommendation and observe whether two users add each other in an online social network. However, we also would like to keep probes to a minimum. Either because they are expensive to perform or because they interfere with users' experience [21] or the system's normal operation. Rather, we describe our task as the following **idealized learning task**: Perform one probe in $(I,J) \sim \mu(G^{(t_0)})$, then predict what would have happened had we probed into the relation of a different pair $(U,V) \sim \mu(G^{(t_0)}), (U,V) \neq (I,J)$. We refer to this task as **causal link prediction**, and more formally define it as estimating the counterfactual quantity

$$P\Big(\underbrace{A_{\mathcal{E}^{(t_1)}}^{(t_1)}\left(\mathcal{E}^{(t_1)} = (U,V)\right)}_{\substack{\text{What would have}\\ \text{happened had we probed}\\ \text{in } (U,V) \text{ instead?}}} \mid A_{\mathcal{E}^{(t_1)}}^{(t_1)}(\mathcal{E}^{(t_1)} = (I,J)), G^{(t_0)}\Big) \equiv P\Big(Y_{UV}^{(t_1)} \mid Y_{IJ}^{(t_1)}\Big). \tag{2}$$

Note that $\mathcal{E}^{(t_1)}$ is an individual quantity, *i.e.*, for an observation $G^{(t_0)}$ it can only take one value, thus Equation (2) is, without further assumptions, a strict counterfactual query (see [5] for a precise definition).

Equation (2)'s query is of interest to a wide range of applications, for instance (we will present experiments in Section 6): (i) determining what would have been the outcome of recommending product V to user U, given user I was recommended product J and bought ($A_{\mathcal{E}^{(t_1)}}^{(t_1)}(\mathcal{E}^{(t_1)} = (I,J)) = 1$) or didn't buy ($A_{\mathcal{E}^{(t_1)}}^{(t_1)}(\mathcal{E}^{(t_1)} = (I,J)) = 0$) it; (ii) knowledge base completion, where we manually check the relation between a pair of entities (I,J) in a knowledge base and then use this experiment to predict what would have happened had we manually checked the relation between entities (U,V); (iii) in refining estimations of the covariance matrix between random variables from experiments, *i.e.*, if $A_{\mathcal{E}^{(t_1)}}^{(t_1)}(\mathcal{E}^{(t_1)} = (I,J)) \approx \text{cov}(I,J)$ is the empirical covariance between two random variables I and J, and our policy $\mu(G^{(t_0)})$ decides to collect more field data to improve the covariance estimates between I and J, Equation (2) allows us to predict what would have been the improved covariance estimate of $\text{cov}(U,V)$ had the policy chosen (U,V) as the pair of random variables.

We highlight that our counterfactual query is with respect to the probing policy $\mu(G^{(t_0)})$, *i.e.*, we require interventional data collected under $\mu(G^{(t_0)})$ and can only answer counterfactual queries about pairs sampled from $\mu(G^{(t_0)})$. A possible interesting extension of this work is showing how to merge data from different policies to answer queries about one of them. Finally, we note that counterfactual questions such as "what would have happened had we not intervened?" are also out of the scope of this work. These questions involve a different causal quantity ($A_{\mathcal{E}^{(t_1)}}^{(t_1)} \mid A_{\mathcal{E}^{(t_1)}}^{(t_1)}(\mathcal{E}^{(t_1)} = (I,J))$) than the one in Equation (2) and would need either further causal assumptions or different data. Moreover, we can see this query as one about a different probing policy, *i.e.*, where edges would have been generated according to the graph's natural evolution process.

Learning from multiple probes. For ease of exposition, we have so far referred to an idealized task where we learn from a single probe $Y_{IJ}^{(t_1)}$ at time t_1 —learning predictions from a single intervention would result in high variance estimates. To consider a more practical solution, we first define the random variable of the outcome of a probe at time $t_M \geq t_1$ in $(I^{(t_M)}, J^{(t_M)}) \sim \mu(G^{(t_0)})$ as

$$Y_{I^{(t_M)}J^{(t_M)}}^{(t_M)} := A_{\mathcal{E}^{(t_M)}}^{(t_M)}\left(\mathcal{E}^{(t_M)} = (I^{(t_M)}, J^{(t_M)})\right) \mid Y_{I^{(t_1)}J^{(t_1)}}^{(t_1)}, \ldots, Y_{I^{(t_{M-1})}J^{(t_{M-1})}}^{(t_{M-1})}, G^{(t_0)}, \tag{3}$$

where each probe outcome $Y_{I^{(t_m)}J^{(t_m)}}^{(t_m)}, 1 \leq m \leq M$ is defined recursively in the same way. Now, we are ready to define the random variable of a sequence of M probes in $G^{(t_0)}$ at times $t_1, \ldots, t_M$ as

$$\mathbf{Y}^{(M)} := \left(Y_{I^{(t_m)}J^{(t_m)}}^{(t_m)}\right)_{m=1}^{M}. \tag{4}$$

Then, we are finally left with the generalized task of estimating the counterfactual quantity

$$P\Big(\underbrace{A^{(t_1)}_{\mathcal{E}^{(t_1)}}\big(\mathcal{E}^{(t_1)}=(U,V)\big)}_{\substack{\text{What would have}\\ \text{happened had we probed}\\ \text{in } (U,V) \text{ at time } t_1 \text{ instead?}}} \mid \bigwedge_{m=1}^{M} A^{(t_m)}_{\mathcal{E}^{(t_m)}}\Big(\mathcal{E}^{(t_m)}=(I^{(t_m)},J^{(t_m)})\Big), G^{(t_0)}\Big) \equiv P\Big(Y^{(t_1)}_{UV} \mid \mathbf{Y}^{(M)}\Big). \tag{5}$$

Challenges. Estimating Equation (5) brings the following challenges:

1. The (causal) data generating process $\mathcal{C}$ of $G^{(t_0)}$ is often unknown. Thus, there might have been unknown probes prior to t_0 and the (causal) evolution of the graph may be path-dependent (*e.g.*, new links may depend on existing ones [2, 19, 55, 58]).
2. Due to spillover [3, 13, 17, 34, 35, 48, 62, 72, 79] and carryover effects [42], subsequent probes might be affected by previous ones. As such, multiple probes ($\mathbf{Y}^{(M)}$) cannot be treated as i.i.d. data.
3. Observing the outcome of a probe in $(I^{(t_m)}, J^{(t_m)})$ might change our belief of what would have been the outcome of a probe in (U, V), *i.e.*, probes ($\mathbf{Y}^{(M)}$) and our prediction (U, V) cannot be treated as i.i.d. data.

Outline of contributions. In what follows we outline this work's contributions to tackle the above challenges.

A. We define the general concept of causal lifting: A generalization of probabilistic lifting [57] to different layers of Pearl's Causal Hierarchy. These definitions are of independent interest, not tied to link prediction.

B. *We define the causal link prediction learning task. We show that lifting in causal link prediction is essentially the structural task of finding symmetries in $G^{(t_0)}$, which is more akin to finding logical rules than the positional goal of finding nearby similar relations in $G^{(t_0)}$. We then show why node embeddings are generally undesirable for these tasks and argue that structural (permutation invariant or equivariant) pairwise embeddings —a special type of joint node pair representation— present low learning bias and capture the correct causal structure from the task, as opposed to existing node embedding methods (e.g., matrix factorization and GNNs).*

C. Finally, we introduce three situations where our theoretical results can be leveraged: (i) to extrapolate knowledge in knowledge bases, (ii) to learn to improve covariance matrix estimations with partially collected data, and (iii) to make recommendations in recommender systems. Overall, results confirm that invariant pairwise embedding methods consistently outperform node embedding ones in causal link prediction tasks.

D. *Contribution in the supplement*: We present —to the best of our knowledge— the first universal family of Structural Causal Models (SCMs) for finite graphs (finite exchangeable two-dimensional arrays).

2 Existing Link Prediction Literature

Link prediction started with Spearman's 1927 work but has more recently gained traction in different applications, *e.g.*, matrix denoising [10], social networks [1], recommender systems [50] and many others [26]. In each problem domain, link prediction has been either treated as an associational (layer 1 of Pearl's Causal Hierarchy) or as a causal task (layer 2 and 3 of Pearl's Causal Hierarchy). In what follows, we review each of these perspectives, highlighting relevant related works while contrasting with our approach.

Associational link prediction methods. Since the first half of the last century, link prediction as an associational task has been studied through what can be formally described as a self-supervised learning task, where a masking process (a.k.a., noise process) hides edges as nonedges (and in some applications also vice-versa). The goal is to predict the true edges and nonedges in the adjacency matrix. By defining the task as self-supervised learning, Appendix B shows how existing methods either implicitly or explicitly assume the masking process is known in training.

To illustrate our self-supervised learning perspective, let us consider matrix factorization on the observed graph $G^{(t_0)}$. More specifically, we let our loss function be given by the mean squared error $\|a^{(t_0)} - UV^T\|_2^2 =$

$\sum_{(i,j)\in V^2}(a_{ij}^{(t_0)} - U_{i\cdot}V_{j\cdot}^T)^2$ with $U \in \mathbb{R}^{n\times d}$ and $V \in \mathbb{R}^{n\times d}$ as our learned low-rank matrices. In this scenario, as further detailed in Appendix B, we assume a uniform mask over $G^{(t_0)}$'s node pairs, while our link reconstruction model is given by a normal distribution around its latent factors' dot product, *i.e.*, $A_{(I,J)}^{(t_0)} \mid a_{-(I,J)}^{(t_0)} \sim \mathcal{N}(U_{I\cdot}V_{J\cdot}^T, 1)$. Note that in this case, apart from the usual causal restrictions imposed by latent factor models, we also have an observational limitation: At test time we have to sample node pairs uniformly at random —an unlikely setting in some applications.

State-of-the-art tensor factorization methods used in knowledge base completion [61, 69, 70] force the embeddings to encode properties of logical rules, such as symmetry/antisymmetry and composition. More recently, Graph Neural Networks (GNNs) [64] have emerged as an alternative way to perform link prediction. Unlike matrix factorization, a GNN embedding is not sensitive to permutations of the node identifiers (e.g., user id in the system). Matrix factorization and GNN node embeddings are the modern workhorses of link prediction. These node embeddings encode associations between graph topology (edges of the graph) as well as node and edge attributes, but they are designed to handle observational instead of interventional data. The framing of link prediction as a self-supervised learning task helps us understand why even state-of-the-art link prediction approaches are associational and unable to predict the query in Equation (5).

There are alternative associational approaches, often referred to as *model-based* methods, that fit a random graph model to $G^{(t_0)}$ in different ways, *e.g.*, considering hierarchical structures [14] or stochastic block models [33, 34]. These models can output the probability of any link in the observed graph. Note, however, that these methods are by definition associational, *i.e.*, they directly compute a (associational) probability distribution from which $G^{(t_0)}$ was generated. As such, they do not provide much flexibility for causal extensions outside from the process defined by the model.

Existing causal link prediction methods. *Mechanistic (network creation) methods:* Link prediction methods have also been derived from models of network formation [46]. For instance, the common neighbors mechanism [51] assumes links are more likely created by nodes that have many common neighbors. The vast majority of other mechanisms such as the Jaccard, the Adamic-Adar [1], the Katz [40], and the preferential attachment [51] indices are all variations of the common neighbors method. All of such methods assume a fixed network formation process and predict links based on it. In the context of this work such methods can be interpreted as defining the interventional policy $\mu(G^{(t_0)})$. The effect of interventions is only captured by these methods if the network formation process follows their assumptions. Otherwise, one needs to estimate a method's effects by extensively performing experiments or, as we propose here, learning to estimate the effects from a few experiments (probes) using the method as our policy $\mu(G^{(t_0)})$.

Inner factors literature. Recent works have proposed to merge matrix factorization and interventional data in recommender systems [6, 75, 77]. In all of such works, both the interventional policy and the outcomes of interventions are given by inner (latent) factor models. What is missing in these methods is a causal structure where the link creation reacts to the current state of the graph (path-dependent models). It is understandable that these existing works choose not to model such co-dependence since it is unclear how to encode it in a causal model and still have a practical method. Later in this work we show how, under mild assumptions, one can develop a practical causal predictor that is robust to this co-evolution. More recently, the authors of [83] proposed to learn the essential factors determining the existence of links by asking the question "would the link still exist had the graph structure been different?". Note how this is essentially a different causal question than the one we approach in this work (see Equation (2)).

Recommendations as treatments literature. Estimating the effect of recommendations, an application of Equation (1), has recently gained traction in the literature denoted as *recommendations as treatments* [39]. Under this framework, we can see $\mu(G^{(t_0)})$ as a (probabilistic) recommendation algorithm and the expected value of Equation (1) as its effectiveness, *i.e.*, a higher expected value corresponds to links with higher probability of recommendation being formed. Solutions to estimate such effectiveness often comes in two flavors: Online and off-line A/B testing. *Online A/B testing*, or simply A/B testing, is the most widely used method to evaluate recommender systems in industry. In this context, A/B testing can be seen as Randomized Controlled Trials (RCTs), where one part of the population (graph) is probed according to $\mu(G^{(t_0)})$ while the another is not, *i.e.*, it is the control group. Unfortunately, this method is often expensive and time consuming, while possibly

exposing the population to unwanted risks. To avoid running into these problems, methods under the umbrella of *off-line A/B testing* propose to evaluate $\mu(G^{(t_0)})$ based on past user experience data, *i.e.*, without running new experiments. The essence of these methods lies in the use of inverse propensity scores [39], which re-weights the importance of previous interventions according to the policy used at the time and a new evaluated policy. Although such methods do not explicitly encode causal modeling assumptions, they implicitly make strong causal assumptions: Both the graph evolution process and the effect of probing into it are time-homogeneous, *i.e.*, the probability of a link being created, under a probe or not, is the same at any point in time. Furthermore, all previous probes are observed, *i.e.*, we have all past interventional and observational data. In this work, we consider a more general scenario where we only have past observational data and no assumptions about time homogeneity in the past.

Causal effect estimation on networks. Another relevant set of works aims at estimating the difference in potential outcomes of two treatments given in a network —also known as causal effect estimation. In this scenario, a given treatment might not just affect the treated individual, but also their neighbors. Recent work in [38] formulates the problem as a multi-task one, which is then solved by a GNN-based framework. The authors of [28] propose to use the network's structure to control confounding bias and learn individual treatment effects. More recently, the work of [47] focuses on hypergraphs, in order to represent group interactions, and learns how to control for confounders and model higher-order interactions to estimate treatments' effects. All of such works fundamentally differ from ours for a main reason: They do not intervene in the graph structure, but rather on the nodes of a given graph. To the best of our knowledge, [65] is the only method that investigates interventions on the graph structure, but, differently to our approach, it aims to study the effect of changes resulting from creating or damaging network ties, rather than probing into relations.

3 Causal Lifting

We commence our discussion by presenting one of our key contributions: *Causal lifting*, a general concept of independent interest beyond link prediction. *Causal lifting* is a causal extension to the associational definition of lifting in probabilistic inference [57, 73]. Here, instead of defining symmetries in the associational distribution, we define them in different layers of Pearl's Causal Hierarchy (associational, interventional and counterfactual). Let us first recall the classical definition of lifting in associational distributions.

Definition 1 (Associational lifting [57, 73])**.** *Let X be our random variable of interest and $\mathcal{G}$ a group where $\cdot$ is the left action of $\mathcal{G}$ onto* $\mathrm{supp}(X)$ *(i.e., the function $\cdot : \mathcal{G} \times \mathrm{supp}(X) \to \mathrm{supp}(X)$ satisfies the following axioms: (i) $e \cdot x = x$, where $e \in \mathcal{G}$ is the identity element; (ii) $g \cdot (h \cdot x) = (gh) \cdot x$, where $g, h \in \mathcal{G}$). We say that $\mathcal{G}$ lifts the associational distribution of X if, $\forall x \in \mathrm{supp}(X), \forall g \in \mathcal{G}$,*

$$P\Big(X = x\Big) = P\Big(X = g \cdot x\Big). \tag{6}$$

Now, since an intervention defines an interventional distribution (see Appendix A), we can directly extend Definition 1 to interventional distributions in Definition 2.

Definition 2 (Interventional lifting)**.** *Consider X and $\mathcal{G}$ as in Definition 1, while Y is a target random variable of interest. We say that $\mathcal{G}$ lifts the interventional distribution of X and Y if, $\forall x \in \mathrm{supp}(X), \forall g \in \mathcal{G}$,*

$$P\Big(Y(X = x)\Big) = P\Big(Y(X = g \cdot x)\Big). \tag{7}$$

Finally, following Definition 2 it is also straightforward to extend lifting to counterfactual distributions as we do in Definition 3 below.

Definition 3 (Counterfactual lifting)**.** *Consider X, Y and $\mathcal{G}$ as in Definition 2. We say that $\mathcal{G}$ lifts the counterfactual distribution of X and Y if, $\forall x, x' \in \mathrm{supp}(X), \forall g \in \mathcal{G}$,*

$$P\Big(Y(X = x) \mid X = x'\Big) = P\Big(Y(X = g \cdot x) \mid X = x'\Big). \tag{8}$$

Definition 1 is used in probabilistic inference algorithms [57, 73] to avoid unnecessary computations: One can replace the need to estimate $|\mathcal{G}|$ quantities in the marginal probability $\sum_{g\in\mathcal{G}} P\big(X = g \cdot x\big)$ by estimating a single quantity $P\big(X = x\big)$. While in probabilistic inference we are interested in efficiently computing associational distributions, in causal inference our main challenge is to *identify a causal quantity*: Without causal lifting, our data may not be enough to answer the causal query. Next, we show how interventional lifting can play a key role in identifying causal link prediction tasks.

4 Interventional Lifting for Link Prediction

Our goal in this section is to show how interventional lifting can serve as an identification strategy and experimental design tool for the causal link prediction task (Equation (5)). To provide the reader with the basic tools used in our solution, in Section 4.1 we introduce a universal family of structural causal models (SCMs), and in Section 4.2 we describe the needed invariance assumptions in the SCM mechanisms that allow us to apply interventional lifting for link prediction. Then, Section 4.3 presents our main result (Theorem 2) in the context of our idealized single probe task (Equation (2)) with (I, J) and (U, V) as isomorphic node pairs. Finally, in Sections 4.4 and 4.5 we show extra causal mechanism assumptions that are sufficient to learn the general task of Equation (5) with multiple probes.

4.1 A universal family of causal models for graphs

Our work considers an underlying causal model that generates edges (together with edge and node attributes) and nonedges sequentially, *i.e.* at time step $t \in \mathbb{N}$ it decides whether an specific pair of nodes (i, j) has an edge (and its value) or not (node attributes are recorded in the pairs (i, i), $i \in V^{(t_0)}$). The causal mechanism deciding which pair of nodes will be assigned an edge or a nonedge at time t is denoted by $f_{\mathcal{E}}^{(t)}$ while the mechanism deciding whether it is an edge vs. nonedge is given by $f_{\mathcal{X}}^{(t)}$. Note that such a model can generate attributed graphs by outputting the edge attribute with $f_{\mathcal{X}}^{(t)}$. Generally speaking, both mechanisms take as input the sequence of all previously generated edges and nonedges, *i.e.*, a path-dependent causal model execution. At observation time (t_0) the (observed) node identifiers are uniformly permuted from the hidden true identifiers and, if the graph is undirected, $a^{(t_0)}$ is symmetrized. In Figure 2 we can see a running example of such an SCM generating a graph of four nodes.

Path-dependency. Note that since the mechanisms $f_{\mathcal{E}}^{(t)}, f_{\mathcal{X}}^{(t)}$ generating (non)edges at any time t take as input all previously generated variables, a causal model $\mathcal{C}$ in this class can be path-dependent. We say that $\mathcal{C}$ *can be* path-dependent since we are not specifying its mechanisms, thus it might be that $f_{\mathcal{E}}^{(t)}, f_{\mathcal{X}}^{(t)}$ ignore the input and use, *e.g.*, a fixed set of inner factors. This set of inner factors might also be dependent on t, which implies that $\mathbb{C}$ contains both path-dependent and (temporal) inner factor models. As we will see in Section 4.2, our mechanism assumptions are not implying independence of previously generated variables, thus the class of causal models we finally consider remains able to represent both path-dependent and (temporal) inner factor models.

The set of SCMs $\mathbb{C}$ denotes the class of all SCMs taking this form, with any mechanisms $(f_{\mathcal{X}}^{(t)}, f_{\mathcal{E}}^{(t)})$ and exogenous variable distributions. Definitions 13 and 14 in Appendix C introduce a formal and complete description of the above family of SCMs. We now show that $\mathbb{C}$ is a universal family of exchangeable graph models, that is, any exchangeable distribution over (countable) graphs can be obtained by some SCM $\mathcal{C} \in \mathbb{C}$.

Theorem 1 (Universality of our graph SCM)**.** *Let $\mathbb{C}$ be the family of SCMs as defined in Definitions 13 and 14 in Appendix C and $\mathbb{A}$ be the domain of the entries of the adjacency matrices of the graphs generated by it. Then,*

i. For every SCM $\mathcal{C} \in \mathbb{C}$ at an arbitrary observation time $t_0 \geq 0$, $\mathcal{C}$ always generates observed graphs $G^{(t_0)}$ where $P(A^{(t_0)} = a) = P(A^{(t_0)} = a')$ for any two isomorphic graphs with adjacencies $a, a' \in \mathbb{A}$;

ii. For all finite (jointly) exchangeable graph distributions $P(A^{(t_0)})$, if $\mathbb{A}$ is a countable set there exists an SCM $\mathcal{C} \in \mathbb{C}$ and an observation time $t_0 \geq 0$ that induces it.

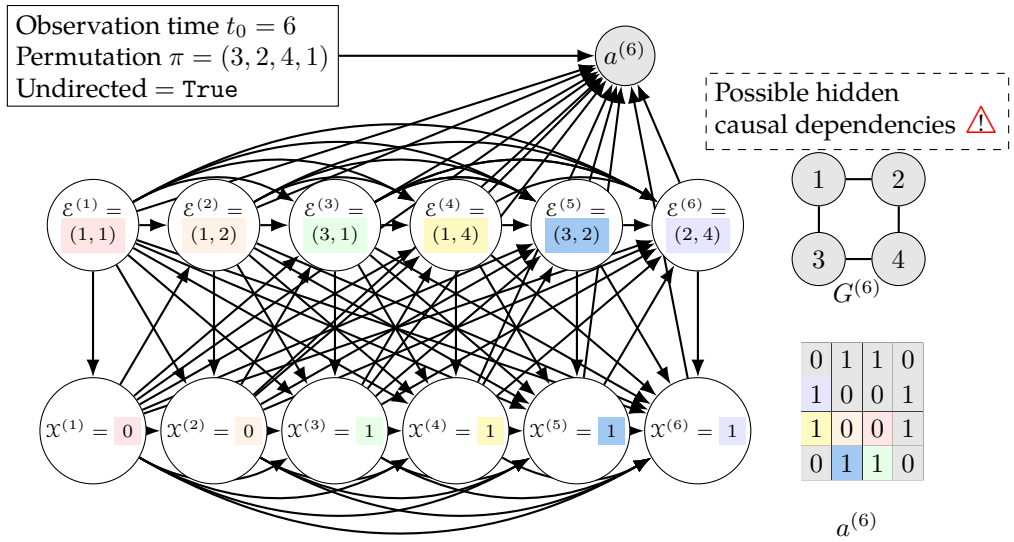

Figure 2: We show the execution of an SCM as described in Appendix C where, together with the observation parameters $t_0 = 6$ and $\pi = (3, 2, 4, 1)$, generate the observed graph $G^{(6)}$ with adjacency matrix $a^{(6)}$. We color the entries in $a^{(6)}$ according to the corresponding variables in the unobserved SCM that generated them. Gray nodes represent observed random variables, while white nodes represent unobserved ones. Note that for simplicity of exposition we omit the exogenous variables in the causal model.

4.2 Sufficient mechanism invariances for interventional lifting in link prediction

After introducing a universal family of causal graph models $\mathbb{C}$ in Section 4.1, we now restrict this family with four causal modeling assumptions that will allow us to apply interventional lifting in our causal link prediction task. We highlight how these are not causal independence (structural) assumptions as it is commonly assumed in causal identification procedures. Instead, we define four mechanism invariances present in the underlying SCM of the task. The mechanism invariance assumptions are a key feature of our work since, by avoiding causal independence assumptions, our models consider possible path dependencies. These invariances are presented informally in the main text; we refer readers to Appendix D for a corresponding formal mathematical description.

Assumption 1 (Time gap ignorability (informal))**.** *We say that our SCM satisfies time gap ignorability if the mechanism $f_{\mathfrak{X}}^{(t_1)}$ is invariant to the SCM intermediate states between the time the intervention probe is performed t_0 and the instant before we see its effect in t_1.*

Assumption 2 (Time exchangeability (informal))**.** *We say that our SCM satisfies time exchangeability if the mechanism $f_{\mathfrak{X}}^{(t_1)}$ is invariant to the order in which edges and nonedges have been generated.*

Assumption 3 (Non-link ignorability (informal))**.** *We say that our SCM satisfies non-link ignorability if the mechanism $f_{\mathfrak{X}}^{(t_1)}$ is invariant to which pairs of nodes were generated as non-links or were not generated at all at time t_0.*

Assumption 4 (Identifier exchangeability (informal))**.** *We say that our SCM satisfies identifier exchangeability if the mechanism $f_{\mathfrak{X}}^{(t_1)}$ is invariant to permutations of the node identifiers.*

We now discuss how the above mechanism invariance assumptions induce a simplified causal DAG. First, note that $t_1 \geq t_0 + 1$, which means that the observation time (t_0) and the time we see the effect of the probe (t_1) are not necessarily consecutive (in the model execution time). In theory, links and non-links generated between the observation of the graph and the probe could influence the outcome of the probe. Since, by having only access to $G^{(t_0)}$, we cannot account for the graph evolution in the interval (t_0, t_1), we need time gap ignorability (Assumption 1) —which assumes that the outcome of the probe is invariant to whatever

happened between t_0 and the moment before t_1. This assumption is pervasive in causal inference since it is not possible to identify causal queries without observing the true distribution of outcomes [9].

Time exchangeability (Assumption 2) is the assumption that the order in which links and non-links have been created until time t_0 does not affect the probe. This is necessary because we only observe a static graph $G^{(t_0)}$ at time t_0. Note that if the order is an important aspect of the task, Assumption 2 would still hold if we represent $G^{(t_0)}$ as a temporal graph (with time stamps as edge attributes in the static graph $G^{(t_0)}$, see [25]).

Assumption 3 is important since at t_0 we are not aware of whether an observed non-link is indeed a non-link generated by the causal model or a node pair not-yet-executed, hence we need the outcome of the probe to be invariant to this difference. By assuming non-link ignorability (Assumption 3), we do not need to distinguish not-yet-executed from non-links. For instance, in recommender systems Assumption 3 makes the simplifying assumption that new purchases are causally influenced by past purchases, not by what one has been exposed but chose not to buy.

It is important to note how Assumption 4 is not equivalent to $\mathcal{C}$ being an exchangeable SCM (Theorem 1 i.). The mechanism $f_{\mathcal{X}}^{(t_1)}$ can use the node identifiers as input and because they are shuffled by π the observed graph is finite exchangeable regardless of $f_{\mathcal{X}}^{(t_1)}$ or any of its mechanisms. Thus, Assumption 4 guarantees that not only the observational distribution is finite exchangeable, but also that its graph generating process also is exchangeable to node ids (at time t_1). This assumption is a reasonable one, but *if we believe identifiers are relevant to the causal model mechanisms we should use them as node features, and then Assumption 4 still holds*.

Finally, we discuss the concept of i.i.d. exogenous variables between node pairs in an SCM $\mathcal{C} \in \mathbb{C}$. Independent and identical exogenous variables are central to identify our counterfactual query. The exogenous variables can be interpreted as the unknown context of a node pair. Then, the i.i.d. assumption between two node pairs holds if we believe they belong to different, non-interfering, but identical contexts. For instance, in a video streaming platform a pair containing a person and a movie and another isomorphic pair containing both a person and a movie from a different geographic location are likely to have both independent and identical contexts. Later, we will assume this for subsets of node pairs, *i.e.*, we never require that this is true for every node pair in $G^{(t_0)}$.

4.3 Causal link prediction with single probe lifting

Now we are ready to describe how interventional lifting can allow us to answer our counterfactual query of Equation (2). We say that two node pairs $(i, j), (u, v)$ from $G^{(t_0)}$ are isomorphic if there exists a permutation $\pi \in \mathrm{Aut}(G^{(t_0)})$ in the automorphism group of $G^{(t_0)}$ such that $(u, v) = \pi \cdot (i, j)$, *i.e.*, (i, j) and (u, v) are structurally indistinguishable in $G^{(t_0)}$. Having this notion in mind, under certain conditions we will show we can obtain the interventional lifting (Definition 2)

$$P\Big(A^{(t_1)}_{\mathcal{E}^{(t_1)}}\Big(\mathcal{E}^{(t_1)} = (I, J)\Big) \mid G^{(t_0)}\Big) = P\Big(A^{(t_1)}_{\mathcal{E}^{(t_1)}}\Big(\mathcal{E}^{(t_1)} = \pi \cdot (I, J)\Big) \mid G^{(t_0)}\Big), \forall \pi \in \mathrm{Aut}(G^{(t_0)}). \tag{9}$$

In words, the above equation states that node pairs isomorphic in $G^{(t_0)}$ (*i.e.*, pairs indistinguishable without node ids) must have the same distribution of probe outcomes. Intuitively, this means that under certain invariance conditions on the causal mechanisms, the observed graph $G^{(t_0)}$ is sufficiently expressive of the causal link formation process. In fact, as we state in Theorem 2, these conditions are precisely the ones discussed in Section 4.2. Theorem 2 shows that if (u, v) is in (i, j)'s orbit in $G^{(t_0)}$, under the causal mechanism invariance conditions from Assumptions 1 to 4 and identical exogenous variables, the probe outcome $Y_{ij}^{(t_1)}$ has the same distribution as the probe outcome $Y_{uv}^{(t_1)}$ (conditioned on $G^{(t_0)}$). That is, the invariances in causal mechanisms coupled with identical exogenous variables ensure interventional lifting, *i.e.*, identical distributions in the orbit. The lifting is better observed in the equivalent causal DAG shown in Figure 3. We then leverage i.i.d. exogenous variables to ensure independence between probes in (i, j) and in (u, v) to then build an estimator in Corollary 1.

Theorem 2 (Invariances for interventional lifting in link prediction)**.** *Let $\mathbb{C}$ be the family of SCMs as defined in Definitions 13 and 14 in Appendix C and $\mathbb{A}$ be the domain of the entries of the adjacency matrices of the graphs generated by it. Then,*

i. if $\mathcal{C} \in \mathbb{C}$ has the mechanism invariances described in Assumptions 1 to 4 and exogenous variable independence at time $t_0 + 1$, i.e., $\mathcal{U}_{\mathcal{X}}^{(t_0+1)} \perp\!\!\!\perp (\mathcal{U}_{\mathcal{X}}^{(t)})_{t=1}^{t_0} \mid (I, J)$ in Definition 13. Then, the effect of an intervention in $\mathcal{C}$ can

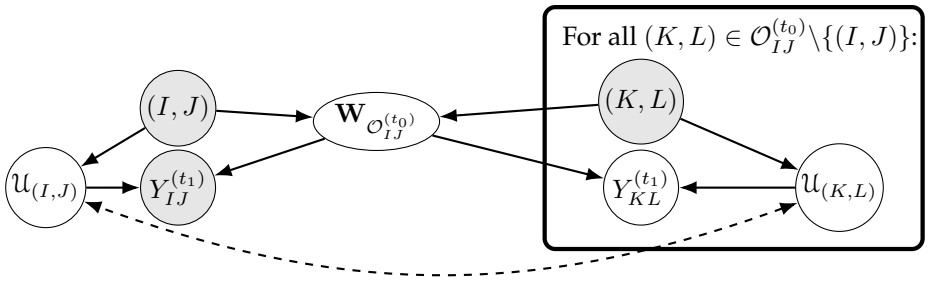

Figure 3: (Theorem 2(i)) Causal DAG of an equivalent data generating process of a probe in (i,j) (*left*) and in its orbit (*right*). As usual, we represent observed and unobserved variables with gray and white nodes respectively.

be equivalently described by Figure 3's causal DAG, where $(I,J) \sim \mu(G^{(t_0)})$ *is the node pair we intervene in,* $\mathcal{O}_{IJ}^{(t_0)} := \{\pi \cdot (I,J) \colon \pi \in \mathrm{Aut}(G^{(t_0)})\}$ *is the set of all structurally indistinguishable pairs to* (I,J) *in* $G^{(t_0)}$*, and* $\mathbf{W}_{\mathcal{O}_{IJ}^{(t_0)}}$ *is a latent variable tied to* $\mathcal{O}_{IJ}^{(t_0)}$ *(the orbit of* (I,J) *in* $G^{(t_0)}$*).*

ii. Under the conditions in (i) and assuming the extra symmetry $P(\mathcal{U}_{(I,J)}) = P(\mathcal{U}_{\pi\cdot(I,J)})$ *in Figure 3's DAG, we have the following interventional lifting result (Definition 2), where* $\forall \pi \in \mathrm{Aut}(G^{(t_0)})$

$$P\left(Y_{IJ}^{(t_1)}\right) = P\Big(A_{\mathcal{E}^{(t_1)}}^{(t_1)}\Big(\mathcal{E}^{(t_1)} = (I,J)\Big) \mid G^{(t_0)}\Big) = P\Big(A_{\mathcal{E}^{(t_1)}}^{(t_1)}\Big(\mathcal{E}^{(t_1)} = \pi\cdot(I,J)\Big) \mid G^{(t_0)}\Big)$$
$$= P\left(Y_{\pi\cdot(I,J)}^{(t_1)}\right).$$

In summary, Theorem 2 outlines a procedure to derive causal modeling conditions in which a probe in (i,j) can be used as an unbiased estimate of Equation (2) on a subset of pairs: (i,j)'s orbit. This is an interventional lifting for link prediction (see Equation (9)). Then, finally, if the exogenous variables of probes in the same orbit $\mathcal{U}_{(i,j)}$ and $\mathcal{U}_{(u,v)}$ are i.i.d., we can introduce the following result.

Corollary 1 (Symmetries for interventional learning)**.** *Under conditions of Theorem 2, let* $(i,j) \in V^{(t_0)} \times V^{(t_0)}$ *be an arbitrary node pair and* $(u,v) \in \mathcal{O}_{ij}^{(t_0)}$ *(i.e.,* (u,v) *is structurally indistinguishable from* (i,j) *in* $G^{(t_0)}$*). Further, assume* $\mathcal{U}_{(i,j)}$ *and* $\mathcal{U}_{(u,v)}$ *are i.i.d.. Then, by Theorem 2(ii)'s DAG in Figure 3*

$$P(Y_{uv}^{(t_1)} \mid Y_{ij}^{(t_1)}) = \int_{\mathbf{W}_{\mathcal{O}_{ij}^{(t_0)}}} P(Y_{uv}^{(t_1)} \mid \mathbf{W}_{\mathcal{O}_{ij}^{(t_0)}}) P(\mathbf{W}_{\mathcal{O}_{ij}^{(t_0)}} \mid Y_{ij}^{(t_1)})\, d\mathbf{W}_{\mathcal{O}_{ij}^{(t_0)}}. \tag{10}$$

If $P(\mathbf{W}_{\mathcal{O}_{ij}^{(t_0)}} \mid Y_{ij}^{(t_1)})$ *has a single mode and low variance, the above equation can be approximated by*

$$P(Y_{uv}^{(t_1)} \mid Y_{ij}^{(t_1)}) \approx P(Y_{uv}^{(t_1)} \mid \mathbf{W}^{\star}_{\mathcal{O}_{ij}^{(t_0)}}),$$

where $\mathbf{W}^{\star}_{\mathcal{O}_{ij}^{(t_0)}} = \mathrm{argmax}_{\mathbf{W}_{\mathcal{O}_{ij}^{(t_0)}}} P(\mathbf{W}_{\mathcal{O}_{ij}^{(t_0)}} \mid Y_{ij}^{(t_1)})$ *is a Maximum A Posteriori (MAP) estimate.*

Theorem 2 together with Corollary 1 provide an identification and estimation procedure for causal link prediction. In Figure 3 we depict how the hidden parameters associated with the orbit $\mathcal{O}_{ij}^{(t_0)}$ of a node pair (i,j) in $G^{(t_0)}$ d-separates the probe from the generating process of $G^{(t_0)}$. As such, we can see Equation (10) as a sequence of *abduction, action, and prediction* (Theorem 7.1.7 in Pearl [53]), using structural information to compose the admissible set of variables in the task. Note, however, that such a procedure learns from a single probe in an isomorphic node pair. Next, we will extend our causal assumptions to allow us to learn from multiple probes in different orbits.

4.4 Lifting in causal link prediction with multiple probes

Corollary 1 shows how a probe in (i, j) can serve as unbiased estimate of the counterfactual query (Equation (2)) on a pair (u, v) in (i, j)'s orbit. A single training example (probe), however, is not an ideal learning setting: We have the outcome of a probe in a single pair and we can answer counterfactual queries only about its orbit. Now, we show how to extend our causal assumptions to learn to answer counterfactual queries about *any pair* using *multiple probes* as training data (Equation (5)).

We start with the condition that allows us to use multiple probes to estimate the expected value in, for instance, Corollary 1: Non-interfering probes (c.f. Assumption 5, formalized in Appendix D). In summary, non-interference in probes means that the outcome of other probes does not interfere in each others' results. This assumption is widely used in causal inference for graph data, see for instance the work of Eckles et al. [17]. In Assumption 5, however, we provide —to the best of our knowledge— the first formalization of non-interference with respect to the the underlying causal mechanisms of the task. Note that non-interfering probes is an invariance condition depending on both the causal mechanisms and on the pairs $(I^{(t_m)}, J^{(t_m)})$'s we probe at each time t_m. Regarding the mechanisms, it needs to be taken as a causal modeling assumption. As of how to choose probes that don't interfere with each other, we can treat it as an experimental design choice. For instance, non-interference tends to be satisfied with higher probability if the probes are performed in pairs far away in the graph [44]. Finally, we refer the reader to [17] for a thorough analysis of experimental design choices that result in non-interfering probes.

Assumption 5 (Non-interfering probes (informal))**.** *We say that a sequence of probes in M pairs $\left((I^{(t_m)}, J^{(t_m)})\right)_{m=1}^{M}$ is non-interfering if every mechanism $f_{\mathcal{X}}^{(t_m)} \colon 1 < m \leq M$ is invariant to probes performed between t_1 and $t_m - 1$.*

Now, we can extend Corollary 1 to use $\mathbf{Y}^{(M)}$, where under non-interference, multiple probes in the orbit of (i, j) can be used to estimate the counterfactual query about any pair (u, v) also in (i, j)'s orbit.

Corollary 2 (Symmetries for interventional learning with multiple non-interfering interventions)**.** *Under the same conditions as Theorem 2(ii), let $\mathbf{Y}^{(M)} := (Y^{(t_m)}_{i^{(t_m)}j^{(t_m)}})_{m=1}^{M}$ be a sequence of outcomes of probes in node pairs in the same orbit $(\mathcal{O}^{(t_0)}_{i^{(1)}j^{(1)}} = \mathcal{O}^{(t_0)}_{i^{(2)}j^{(2)}} = \ldots = \mathcal{O}^{(t_0)}_{i^{(M)}j^{(M)}})$. Then, for $(u, v) \in \mathcal{O}^{(t_0)}_{i^{(1)}j^{(1)}}$, if the exogenous variables $\{\mathcal{U}_{(i^{(t_m)},j^{(t_m)})} \colon m \in [M]\} \cup \{\mathcal{U}_{(u,v)}\}$ are i.i.d. and $\mathbf{Y}^{(M)}$ is a sequence of non-interfering interventions, then*

$$P(Y^{(t_1)}_{uv} \mid \mathbf{Y}^{(M)}) = \int_{\mathbf{W}_{\mathcal{O}^{(t_0)}_{ij}}} P(Y^{(t_1)}_{uv} \mid \mathbf{W}_{\mathcal{O}^{(t_0)}_{ij}}) P(\mathbf{W}_{\mathcal{O}^{(t_0)}_{ij}} \mid \mathbf{Y}^{(M)})\, d\mathbf{W}_{\mathcal{O}^{(t_0)}_{ij}}. \tag{11}$$

If $\mathbf{W}_{\mathcal{O}^{(t_0)}_{ij}} \mid \mathbf{Y}^{(M)}$ has low variance, the above equation can be well-approximated by

$$P(Y^{(t_1)}_{uv} \mid \mathbf{Y}^{(M)}) \approx P(Y^{(t_1)}_{uv} \mid \mathbf{W}^{\star}_{\mathcal{O}^{(t_0)}_{ij}}),$$

where $\mathbf{W}^{\star}_{\mathcal{O}^{(t_0)}_{ij}} = \operatorname{argmax}_{\mathbf{W}_{\mathcal{O}^{(t_0)}_{ij}}} P(\mathbf{W}_{\mathcal{O}^{(t_0)}_{ij}}) \prod_{m=1}^{M} P(Y^{(t_1)}_{i^{(m)}j^{(m)}} \mid \mathbf{W}_{\mathcal{O}^{(t_0)}_{ij}})$ is a MAP estimate.

Although we can now use multiple probes to estimate our counterfactual query, we are still restricted to probing in the orbit $\mathcal{O}^{(t_0)}_{i^{(1)}j^{(1)}}$ of the first queried pair $(i^{(1)}, j^{(1)})$. Such a setting tends to be impractical since i) both random and real-world graphs tend to be nearly asymmetric (most orbits are of size one with high probability) [18, 84] and ii) computing the orbit of a pair of nodes is as hard as solving the graph isomorphism problem [24]. Therefore, we finally consider a more practical solution, where we sample pairs according to an arbitrary policy $\mu(G^{(t_0)})$, probe into their relationships and *learn a model* capable of predicting our counterfactual query (Equation (5)) for any $(U, V) \sim \mu(G^{(t_0)})$.

Such a general solution relies on two extra assumptions: i) exogenous variables from the pairs sampled during training, $\{\mathcal{U}_{(i^{(t_m)},j^{(t_m)})}\}_{m=1}^{M}$, and the tested pair, $\mathcal{U}_{(u,v)}$, are i.i.d. and ii) the parameters of the probes' distributions are shared across all orbits. In particular, for ii) we consider a shared set of parameters $\mathbf{W}_{\Gamma^\star}$, which parameterizes a representation function $\Gamma^\star$. Note that, in order to express all possible probe distributions, the set of parameters $\mathbf{W}_{\Gamma^\star}$ must assign the same value to (i, j) and (u, v) if and only if (u, v) is in (i, j)'s orbit. We refer to such as a most-expressive representation, defined next.

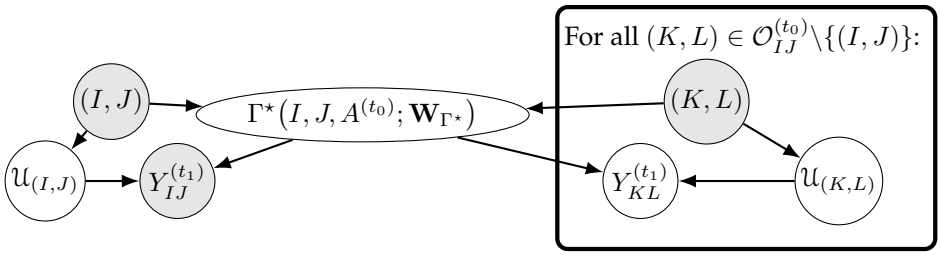

Figure 4: The causal diagram of the data generating process of a probe in (i, j) (*left*) and in its orbit (*right*) that allows us to learn from interventions in a supervised learning fashion (see Equation (13) and Proposition 1).

Definition 4 (Most-expressive pairwise representation)**.** *A most-expressive pairwise representation is given by the functional* $\Gamma^\star \colon V^{(t_0)} \times V^{(t_0)} \times \mathbb{A}^{n^{(t_0)} \times n^{(t_0)}} \times \mathbb{R}^p \to \mathbb{R}^d$*, parameterized by* $\mathbf{W}_{\Gamma^\star} \in \mathbb{R}^p$*,* $p \geq 1$*, where*

i. for any parameter $\mathbf{W}_{\Gamma^\star}$*, any input graph* $a^{(t_0)} \in \mathbb{A}^{n^{(t_0)} \times n^{(t_0)}}$ *and any pair* $(i, j) \in V^{(t_0)} \times V^{(t_0)}$*, we have that* $\Gamma^\star(i, j, A^{(t_0)}; \mathbf{W}_{\Gamma^\star}) = \Gamma^\star(k, l, A^{(t_0)}; \mathbf{W}_{\Gamma^\star}), \forall\ (k, l) \in \mathcal{O}_{i,j}^{(t_0)}$*, and*

ii. $\exists\ \mathbf{W}'_{\Gamma^\star} \in \mathbb{R}^p$ *such that for any input graph* $a^{(t_0)} \in \mathbb{A}^{n^{(t_0)} \times n^{(t_0)}}$ *and any pair* $(i, j) \in V^{(t_0)} \times V^{(t_0)}$*, we have that* $\Gamma^\star(i, j, A^{(t_0)}; \mathbf{W}'_{\Gamma^\star}) \neq \Gamma^\star(r, s, A^{(t_0)}; \mathbf{W}'_{\Gamma^\star}), \forall\ (r, s) \in (V^{(t_0)} \times V^{(t_0)}) \setminus \mathcal{O}_{i,j}^{(t_0)}$*.*

Leveraging Definition 4, we depict this modification in the SCM in Figure 4. Now, we are ready to state our main learning result and summarize its practical implications in experimental design and assumptions.

Proposition 1. *Under conditions of Theorem 2(ii), if* $\mathbf{Y}^{(M)}$ *is a sequence of outcomes of non-interfering interventions (Assumption 5), the exogenous variables* $\{\mathcal{U}_{(u,v)}\} \cup \{\mathcal{U}_{(i^{(t_m)}, j^{(t_m)})} \colon 1 \leq m \leq M\}$ *are i.i.d., and* $\Gamma^\star(\cdot, \cdot, \cdot; \mathbf{W}_{\Gamma^\star}) \colon V^{(t_0)} \times V^{(t_0)} \times \mathbb{A}^{n^{(t_0)} \times n^{(t_0)}} \to \mathbb{R}^d$ *be a most-expressive pairwise representation function (cf. Definition 4), we have that the causal DAG in Figure 3 can be equivalently described by the causal DAG in Figure 4, which by Corollary 1 yields*

$$P(Y_{uv}^{(t_1)} \mid \mathbf{Y}^{(M)}) = \int_{\mathbf{W}_{\Gamma^\star}} P(Y_{uv}^{(t_1)} \mid \mathbf{W}_{\Gamma^\star}) P(\mathbf{W}_{\Gamma^\star} \mid \mathbf{Y}^{(M)})\, d\mathbf{W}_{\Gamma^\star}. \tag{12}$$

Note that in practice we will approximate the above equation by

$$P(Y_{uv}^{(t_1)} \mid \mathbf{Y}^{(M)}) \approx P(Y_{uv}^{(t_1)} \mid \mathbf{W}^\star_{\Gamma^\star}),$$

where $\mathbf{W}^\star_{\Gamma^\star} = \operatorname{argmax}_{\mathbf{W}_{\Gamma^\star}} P(\mathbf{W}_{\Gamma^\star}) \prod_{m=1}^{M} P(Y_{i^{(m)} j^{(m)}}^{(t_1)} \mid \mathbf{W}_{\Gamma^\star})$ is a MAP estimate and the prior $P(\mathbf{W}_{\Gamma^\star})$ is a hyperparameter of our model.

Proposition 1 in practice. Let us now summarize the practical conditions in which the estimator presented in Proposition 1 (leveraged in our final solution) can be used. We can break down the assumptions into three sets: i) causal model and mechanism invariances (Appendix C, Assumptions 1 to 4); ii) parameter-sharing in orbit representation (Definition 4); and iii) non-interference and i.i.d. exogenous variables (Assumption 5).

The causal modeling assumptions (i) are world models that we must believe to be true so we can identify the causal quantities in the task at hand —as generally needed in causal inference settings [53]. Regarding (ii), the assumption is related to the intuition we maintained throughout the paper: Structure induces link formation. That is, in Proposition 1 we are assuming that two isomorphic node pairs have the same probe distribution and that their orbit representations are given by a common set of parameters. In theory, since $\Gamma^\star$ is most-expressive, we can assign arbitrary and unique representations to each orbit and thus they are not necessarily related. However, in practice the parameter-sharing will induce similar predictions to node pairs that have similar, but not necessarily identical, structure. Therefore, generally speaking, under the assumption that structure can predict the (interventional) formation of links, this assumption is justified. One way to test this in practice is to probe into node pairs with similar structure and (statistically) test whether the observed probe distributions are the same. In Appendix F we do this for real-world recommender systems data and confirm the hypothesis.

Finally, we have the assumptions about non-interference and i.i.d. exogenous variables (iii). These are assumptions may not hold in practice but, unlike (i,ii), they can be enforced by experimental design. Non-interference can be induced by sampling training (interventional) data from distant (*e.g.*, different clusters) parts of the graph, as extensively discussed in Eckles et al. [17]. Non-interference is important for our counterfactual query since it allows us to use multiple interventions as training data. Note that our query is about t_1 and we can only interfere sequentially. As for exogenous variables, we can first think of them as the hidden, *i.e.*, unobserved, contexts of each pair. Having this in mind, it is clear that independent contexts can expected to hold by sampling training (interventional) data from distant parts of the graph. Lastly, having identical context (exogenous) distributions is a challenge often arising in causal inference [39]. A common way to accomplish it through experimental design is to stratify, *i.e.*, train a model for each population stratum expected to have similar context (exogenous) distributions, *e.g.* user demographics.

4.5 Causal lifting induces a supervised learning solution to causal link prediction

Proposition 1 describes when it is possible to estimate our general causal link prediction task (Equation (5)) with multiple probes in different orbits. Let us now leverage such result to present our *final practical solution to the task*. We are interested in a supervised learning-based approach, where the examples are the probed pairs $(i^{(t_m)}, j^{(t_m)})$ with their observed outcomes $(\mathbf{y}_m^{(M)})$ as their labels. We are also interested in graph embedding models, *i.e.*, our prediction of Equation (5) is given by $\rho\big(\Gamma(U, V, A^{(t_0)}; \mathbf{W}_\Gamma^\star); \mathbf{W}_\rho^\star\big)$, where $\Gamma(U, V, A^{(t_0)}; \mathbf{W}_\Gamma^\star)$ outputs a pairwise representation (embedding) of (U, V) in $G^{(t_0)}$ and $\rho(\cdot; \mathbf{W}_\rho)$ is a link function (*e.g.*, a downstream neural network) where the parameters $\mathbf{W}_\rho^\star, \mathbf{W}_\Gamma^\star$ are learned by solving

$$\mathbf{W}_\rho^\star, \mathbf{W}_\Gamma^\star := \underset{\mathbf{W}_\rho, \mathbf{W}_\Gamma}{\operatorname{argmin}} \frac{1}{M} \sum_{m=1}^{M} \mathcal{L}\Big(\mathbf{y}_m^{(M)}, \rho\big(\Gamma(i^{(t_m)}, j^{(t_m)}, A^{(t_0)}; \mathbf{W}_\Gamma); \mathbf{W}_\rho\big)\Big), \tag{13}$$

with $(I^{(t_m)}, J^{(t_m)}) \sim \mu(G^{(t_0)})$, where $\mu(G^{(t_0)})$ is an arbitrary distribution over node pairs, $\mathbf{y}_m^{(M)}$ is the outcome of probe $(i^{(t_m)}, j^{(t_m)})$ at time t_m and $\mathcal{L}$ a nonnegative loss function that optimizes $\mathbf{W}_\rho^\star, \mathbf{W}_\Gamma^\star$ towards the MLE estimates of the task.

We can now see how our solution in Equation (13) is estimating $\mathbf{W}_\Gamma^\star$ according to Proposition 1 — where we assume a non-informative prior over $\mathbf{W}_\Gamma$. We can also interpret $\rho(\cdot, \mathbf{W}_\rho^\star)$ as our estimation of the mechanism that takes $\Gamma(I, J, A^{(t_0)}; \mathbf{W}_\Gamma^\star)$ and outputs $A_{IJ}^{(t_1)}$. Finally, we highlight that Equation (13) relies on the assumptions from Proposition 1 to guarantee that $\rho(\cdot, \mathbf{W}_\rho^\star)$ is an unbiased estimate of our counterfactual query from Equation (5).

With the myriad of existing graph embedding methods, we are finally left with the question: What are good choices for Γ? As usual in machine learning, a choice of Γ is better than another if it achieves lower error with the same (or less) number of samples. Next, we show how classical node embedding methods, such as matrix factorization and graph neural networks, fail at either achieving low error or capturing the correct causal structure of the task. As an alternative, we show how structural (joint) pairwise embeddings can overcome the existing issues with node embeddings.

5 Graph Embeddings for Causal Link Prediction

We now examine our general identification result (Proposition 1) when using structural and (strictly) positional node embeddings —two widely-used family of graph embedding predictors— and structural (joint) pairwise embeddings. Node embedding methods build Γ by separately computing the node embeddings of the two nodes in the represented pair. Then, they are merged by some binding function. For instance, if the graph is undirected, such binding can be done using a Hadamard product of the two node embeddings. We note that our following analysis is agnostic to the choice of the binding function. Therefore, without loss of generality, in node embedding methods we will consider Γ as the concatenation of the two embeddings. The (arbitrary) binding operation is then incorporated by the link function ρ.

Overall, we find that **structural (joint) pairwise embeddings present lower bias than structural node embeddings and, unlike strictly positional node embeddings, correctly represents the causal structure of**

the task. Having the causal model from Figure 4 in mind, it is natural to turn to structural representations as a candidate graph embedding choice.

5.1 Structural (joint) pairwise embeddings are the ideal graph embeddings for causal link prediction tasks

In Proposition 1, and its causal DAG in Figure 4, we see that the natural representation for our causal link prediction task is a most-expressive pairwise representation as in Definition 4, which for any input pair $(i, j) \in V^{(t_0)} \times V^{(t_0)}$ it is a (possibly unique) representation of the orbit $\mathcal{O}_{ij}^{(t_0)}$. That is, under the conditions of Theorem 2(ii), most-expressive pairwise embeddings capture all (causal) invariances in the task. In practice, however, one generally chooses a less expressive model family, since most-expressive graph models necessarily incur high computational costs (since they must solve the graph isomorphism task [12]).

Composing structural node embeddings is a popular attempt to design structural representations of node pairs [36]. However, as we later show in Section 5.2, even most-expressive structural node embeddings can induce model bias in causal link prediction tasks. Thus, to distinguish structural representations acting jointly on the node pair from structural representations acting separately on its nodes, we next define structural joint pairwise embeddings.

Definition 5 (Structural joint pairwise embeddings)**.** *A structural joint pairwise embedding is given by the functional* $\Gamma^{(joint)} \colon V^{(t_0)} \times V^{(t_0)} \times \mathbb{A}^{n^{(t_0)} \times n^{(t_0)}} \times \mathbb{R}^p \to \mathbb{R}^d$*, parameterized by* $\mathbf{W}_{\Gamma^{(joint)}} \in \mathbb{R}^p$*,* $p \geq 1$*, where for some graph* $a^{(t_0)} \in \mathbb{A}^{n^{(t_0)} \times n^{(t_0)}}$ *and some parameter* $\mathbf{W}_{\Gamma^{(joint)}} \in \mathbb{R}^p$ *the joint representation encodes more than the node orbits of* $i \in V^{(t_0)}$ *and* $j \in V^{(t_0)}$ *in* $a^{(t_0)}$ *separately, it encodes* (i, j)*'s the joint orbit* $\mathcal{O}_{(i,j)}^{(t_0)}$*. More precisely,* $\Gamma^{(joint)}$ *is such that*

i. for any parameter $\mathbf{W}_{\Gamma^{(joint)}}$*, any input graph* $a^{(t_0)} \in \mathbb{A}^{n^{(t_0)} \times n^{(t_0)}}$ *and any pair* $(i, j) \in V^{(t_0)} \times V^{(t_0)}$*, we have that* $\Gamma^{(joint)}(i, j, A^{(t_0)}; \mathbf{W}_{\Gamma^{(joint)}}) = \Gamma^{(joint)}(k, l, A^{(t_0)}; \mathbf{W}_{\Gamma^{(joint)}}), \forall\ (k, l) \in \mathcal{O}_{i,j}^{(t_0)}$*, and*

ii. $\exists \mathbf{W}'_{\Gamma^{(joint)}} \in \mathbb{R}^p$*,* $\exists a^{(t_0)} \in \mathbb{A}^{n^{(t_0)} \times n^{(t_0)}}$*, and* $\exists (i, j) \in V^{(t_0)} \times V^{(t_0)}$ *such that it does* ***not*** *exist a function over the set of node orbits* $(\bigcup_{v \in V} \{\mathcal{O}_v^{(t_0)}\})$ *that can be equivalent to* $\Gamma^{(joint)}$*, that is, for all* $f \colon \bigcup_{v \in V} \{\mathcal{O}_v^{(t_0)}\} \times \bigcup_{v \in V} \{\mathcal{O}_v^{(t_0)}\} \to \mathbb{R}^d$ *we have that* $f(\mathcal{O}_i^{(t_0)}, \mathcal{O}_j^{(t_0)}) \neq \Gamma^{(joint)}(i, j, a; \mathbf{W}'_{\Gamma^{(joint)}})$.

Note how the above definition guarantees that for some input graph $G^{(t_0)}$, its structural pairwise embedding could not have been computed from the nodes' individual orbits. At first sight it might seem that only most-expressive pairwise representations are encompassed by Definition 5, but this is not true. For instance, pairwise distances cannot be computed from node orbits (see [45]), thus any structural pairwise representation capturing distances would be a joint embedding as in Definition 5. In practice, we have seen how even for observational link prediction tasks (*e.g.*, see the OGB [36] suite[1]) architectures under Definition 5 consistently outperform others using only structural node representations. This serves as evidence that joint properties, *i.e.*, as in Definition 5(ii), such as distances and common neighbors indeed play a central role in link formation mechanisms. Generally speaking, the benefits enjoyed by estimators using Definition 5 are tied to how much structural properties govern the (interventional) link formation process. In Appendix F we test this assumption in practice. Finally, for simplicity we will often refer to representations satisfying Definition 5 simply as *structural pairwise embeddings*.

5.2 Structural node embeddings are undesirable for causal link prediction due to model bias

Unlike joint embeddings (Definition 5), Graph Neural Networks (GNNs) [64] are the most common permutation-invariant[2] node embeddings (a.k.a. structural node embeddings), where each node gets its own representation as described in Definition 6. A structural node embedding outputs the same node

[1] `https://ogb.stanford.edu/docs/linkprop/`

[2] Invariant node embeddings can also be seen as equivariant embeddings, where the input of the model is the adjacency a and the output is a matrix of embeddings $H \in \mathbb{R}^{n \times d}$. Then, equivariance is achieved by having the action of permutation π in the input $\pi \cdot a$ resulting in permuting the rows of the output matrix of embeddings accordingly, *i.e.*, $\pi \cdot H$.

representations to any two isomorphic nodes in $G^{(t_0)}$. That is, if $\pi \in \mathrm{Aut}(G^{(t_0)})$ is an automorphism of $G^{(t_0)}$, then for any $i \in V^{(t_0)}$ we have that $\pi \cdot a^{(t_0)} = a^{(t_0)}$ and thus nodes i and $\pi \cdot i$ are assigned the same representation in $a^{(t_0)}$. Overall, structural node embeddings encompass not only GNNs but also classical structural node roles [31].

Definition 6 (Structural (permutation-invariant) node embeddings [68])**.** *A structural node embedding is given by the functional* $Z \colon V^{(t_0)} \times \mathbb{A}^{n^{(t_0)} \times n^{(t_0)}} \times \mathbb{R}^p \to \mathbb{R}^d$ *parameterized by* $\mathbf{W}_Z \in \mathbb{R}^p, p \geq 1$, *where* $Z(i, a^{(t_0)}; \mathbf{W}_Z) = Z(\pi \cdot i, \pi \cdot a^{(t_0)}; \mathbf{W}_Z)\ \forall\ \pi \in \mathbb{S}_{n^{(t_0)}}, \forall\ a^{(t_0)} \in \mathbb{A}^{n^{(t_0)} \times n^{(t_0)}}, \forall\ i \in V^{(t_0)}$.

We now highlight that Definitions 4 and 6 are not equivalent. In fact, next we prove that the family of all models satisfying Definition 5 have lower bias than those satisfying Definition 6. We first define the bias of a graph embedding solution to our problem below.

Definition 7 (Graph embedding bias)**.** *We define the bias of a representation* $\Gamma(\cdot; \mathbf{W}_\Gamma)$ *together with a link function* $\rho(\cdot; \mathbf{W}_\rho)$ *for a fixed* t_0 *as the expectation*

$$\mathcal{B}_{\Gamma,\rho} := \min_{\mathbf{W}_\Gamma, \mathbf{W}_\rho} \mathbb{E}\left[\mathcal{L}\Big(Y_{IJ}^{(t_1)}, \rho\big(\Gamma(I, J, A^{(t_0)}; \mathbf{W}_\Gamma); \mathbf{W}_\rho\big)\Big)\right]$$

taken over (I, J), $Y_{IJ}^{(t_1)}$ *and* $A^{(t_0)}$ *with* $\mathcal{L}$ *being a loss function as in Equation* (13).

Now, we define a theoretically-relevant class of graphs in link prediction tasks: Pairwise symmetric graphs (c.f. Definition 8). In essence, pairwise symmetric graphs are symmetric graphs that contain pairs of nodes that are node-wise isomorphic. That is, there exist permutations that map nodes individually from one pair to the other, but there are at least two of these pairs that are not isomorphic, *i.e.*, it does not exist a single permutation bringing one pair to the other. These graphs are important since structural node embeddings will mistakenly assign symmetries to non-isomorphic pairs —and thus they fail to fulfill Definition 5(ii). Instead, we need to consider pairwise symmetries by using structural pairwise embeddings as in Definition 5.

Definition 8 (Pairwise symmetric graphs)**.** *We say that a symmetric graph* a *is pairwise symmetric if* $\exists\ \pi, \pi' \in \mathrm{Aut}(a), \exists\ i, j, u, v \in V$ *such that* $u = \pi \cdot i, v = \pi' \cdot j$, *but* $\nexists\ \pi^\star \in \mathrm{Aut}(A^{(t_0)})$ *such that* $(u, v) = \pi^\star \cdot (i, j)$.

We are now finally ready to state when structural pairwise embeddings have lower bias than structural node embeddings in Theorem 3.

Theorem 3. *Let* $\mathcal{B}_{Z,\rho}$ *and* $\mathcal{B}_{\Gamma^{(joint)},\rho}$ *be the respective biases (c.f. Definition 7) of models using structural node embeddings (c.f. Definition 6) and structural pairwise embeddings (c.f. Definition 5) in our causal link prediction task as described in Proposition 1. Then, we have that if* $\mathrm{supp}(A^{(t_0)})$ *contains at least one pairwise symmetric graph (cf. Definition 8),*

$$\mathcal{B}_{Z,\rho} \geq \mathcal{B}_{\Gamma^{(joint)},\rho} = 0.$$

The core of Theorem 3's proof (c.f. Appendix E) relies on showing (i) how structural node embeddings cannot achieve zero error in pairwise symmetric graphs and (ii) that an SCM satisfying the conditions of Proposition 1 can generate pairwise symmetric graphs —thus, structural node embeddings cannot achieve zero-bias in causal link prediction tasks. We highlight how this result encompasses even most-expressive structural node embeddings, *i.e.*, representations unique to each node orbit. A natural follow-up question is: What about matrix factorization-based methods? Since they do not explicitly capture invariances, they do not seem to suffer from the same higher-bias problems as structural node embeddings. Next, we show how matrix factorization and other types of (strictly) positional node embeddings, unlike structural pairwise embeddings, do not capture the correct causal structure of our task.

5.3 (Strictly) positional node embeddings (*e.g.*, matrix factorization embeddings) incorrectly encode the causal structure for interventional lifting

Beyond structural node embeddings, it is also natural to consider positional node embeddings as our choice for building Γ. The formal and most general definition of positional node embeddings is given in [68], where the embeddings are samples from an equivariant distribution.

Definition 9 (Positional node embeddings [68])**.** *The positional node embeddings* $(\theta_i)_{i=1}^{n^{(t_0)}}$ *of a graph* $G^{(t_0)}$ *with adjacency* $A^{(t_0)}$ *are defined as joint samples* $(\theta_i)_{i=1}^{n^{(t_0)}} \mid A^{(t_0)} = a^{(t_0)} \sim P(\boldsymbol{\theta} \mid A^{(t_0)} = a^{(t_0)})$ *with* $\operatorname{supp}(\theta_i) \subseteq \mathbb{R}^d$ *and* $P(\pi \cdot \boldsymbol{\theta} \mid A^{(t_0)} = a^{(t_0)}) = P(\boldsymbol{\theta} \mid A^{(t_0)} = \pi \cdot a^{(t_0)})$.

Deterministic algorithms for matrix factorization may not appear to follow the above definition, however, once we consider that node identifiers are arbitrarily assigned, even deterministic factorization methods follow Definition 9. We refer the reader to Srinivasan & Ribeiro [68] for a more comprehensive discussion. At this point, the attentive reader has noted how the presented definition of positional node embeddings is quite general. As such, as previously noted, the embeddings possibly do not encode any graph symmetry, *e.g.*, assign the same representation to isomorphic pairs in the graph. However, the definition is so general that the opposite might also be true, *i.e.*, structural node embeddings can be framed as positional in this framework. Thus, to capture the positional node embeddings that do not encapsulate any symmetry, we next define strictly positional node embeddings.

Definition 10 (Strictly positional node embeddings)**.** *Given a symmetric graph* $G^{(t_0)}$ $(\mid \operatorname{Aut}(G^{(t_0)}) \mid > 1)$*, we say that the positional node embeddings* $(\theta_i^{(pos+)})_{i=1}^{n}$ *of nodes in* $G^{(t_0)}$ *are strictly positional if for every* $(i,j) \in V^{(t_0)} \times V^{(t_0)}, i \neq j$ *we have that* $\theta_i^{(pos+)} \neq \theta_j^{(pos+)}$ *almost everywhere.*

In words, Definition 10 is defining the set of positional node embedding distributions that always assign distinct representations to every node in a symmetric graph. Note that we focus on symmetric graphs here since in asymmetric graphs ($\mid \operatorname{Aut}(G^{(t_0)}) \mid = 1$) even structural node embeddings can assign distinct representations to all nodes. We start by further noting how, just like structural pairwise embeddings, strictly positional node embeddings can achieve zero-bias in any causal link prediction task. It follows from Theorem 2 in [68] that, with a powerful enough link function, strictly positional embeddings can in expectation recover most-expressive pairwise representations (cf. Definition 4) —which achieve zero-bias as shown in Theorem 3.

Although strictly positional node embeddings do not necessarily impose a model bias to the causal link prediction task, they do suffer from a different challenge: Figure 5 shows that because strictly positional node embeddings generally assigns different embeddings for pairs of nodes in the same orbit, they do not capture the correct causal DAG of Figure 2. Further, having the causal model from Figure 2 in mind, we can see how

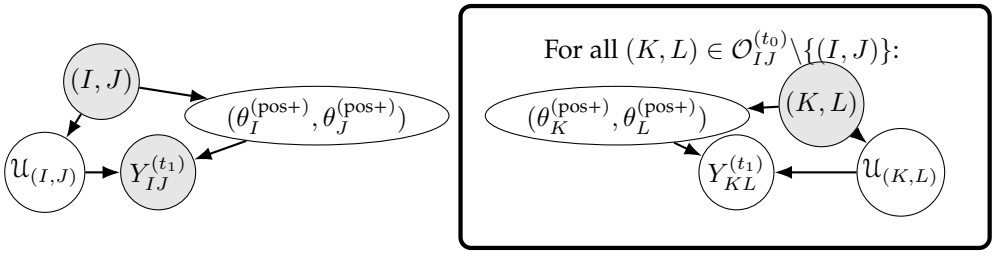

Figure 5: Strictly positional node embedding's incorrect causal structure of probes in (i,j) (*left*) and other pairs in the orbit of $\mathcal{O}_{(i,j)}^{(t_0)}$ (*right*).

strictly positional node embeddings make it difficult to generalize to unseen node pairs. In practice, unless we choose or learn a link function (a link function is the function that takes the embeddings and outputs a link prediction) that assigns the same prediction to the different embeddings of node pairs in the same orbit, the lifting does not occur.

Matrix factorization gives strictly positional node embeddings. When considering (strictly) positional node embeddings, we are interested in matrix factorization and other factor models that reflect some notion of node distances in the graph in their node embeddings, such as metric embeddings [11, 20] and similar traditional (associational) link prediction methods [27, 56, 59]. Thus, to illustrate how the analysis of strictly positional node embeddings is insightful for factor models, we now prove that, in a wide family of graphs, *SVD behaves as a strictly positional embedding*.

SVD node embeddings use the eigenvectors of $a^{(t_0)}{a^{(t_0)}}^T$ and ${a^{(t_0)}}^T a^{(t_0)}$ —here we consider concatenating both right and left eigenvectors. SVD justifies the positional node embedding name since it does reflect

the distances between nodes in the graph in its embedding space. As isomorphic nodes can be arbitrarily distant in a graph, it is easy to see that SVD can assign them different embeddings. Thus, it has been believed and recently conjectured in [85][Theorem 4.2] that SVD in general does not assign the same embedding to isomorphic nodes. To fill this gap in literature, we next present the first result —to the best of our knowledge— on the exact invariances of SVD: When SVD assigns the same embedding to a node pair. The proof of Theorem 4 is presented in Appendix E.

Theorem 4 (The invariances of SVD)**.** *Let G be a graph with adjacency $a \in \{0,1\}^{n\times n}$ and $\theta_i^{(SVD)}$ the SVD embedding of node $i \in V$. Then, two nodes $i, j \in V$ get the same SVD embedding $\theta_i^{(SVD)} = \theta_j^{(SVD)}$ if and only if*

(*a*) *the nodes are isomorphic $i \cong j$; and*

(*b*) *they have the exact same neighborhood: $a_{iv} = a_{jv}, a_{vi} = a_{vj},\ \forall v \in V$.*

From Theorem 4 we can directly derive Corollary 3, in which we outline the exact set of symmetric graphs where SVD embeddings are strictly positional. These symmetric graphs are quite specific: they have duplicate nodes, *i.e.*, nodes that are not only symmetric, but that **carry the exact same information: equal neighborhood**.

Corollary 3. *Given a symmetric and unattributed ($\mathbb{A} = \{0,1\}$) graph $G^{(t_0)}$, its SVD embeddings $(\theta_i^{(SVD)})_{i=1}^{n^{(t_0)}}$ are strictly positional if and only if there exist two nodes $i, j \in V^{(t_0)}$ such that $j \in \mathcal{O}_i^{(t_0)}$ and $a_{iv}^{(t_0)} = a_{jv}^{(t_0)},\ \forall v \in V^{(t_0)}$.*

Note that even in graphs with duplicate nodes, SVD will assign different embeddings to all other nodes. Therefore, although it can capture some degree of invariance from very specific nodes, SVD in general behaves as a strictly positional embedding. As such, it encodes the causal process from Figure 5, *i.e.*, it does not encapsulates the known invariance of the task: two isomorphic pairs must be assigned the same prediction.

Apart from SVD, we can also relate strictly positional node embeddings to neural matrix factorization [16]. Neural matrix factorization can be seen as using a one-hot encoding of the node identifier as the node embedding. Then, it uses a multi-layer perceptron in the link function —in general if the graph is undirected there exists some level of weight-sharing between the two node embeddings in the link function as well. Since node identifiers are by definition unique, it is straightforward to see that *neural matrix factorization produces strictly positional node embeddings*.

Overall, our theory implies that the use of strictly positional node embeddings, such as SVD and neural matrix factorization, entails an incorrect causal structure under our assumptions that does not result in a causal lifting for the causal link prediction task in Equation (2). As a result, models using strictly positional node embeddings will not generalize as well as models using pairwise structural embeddings. More specifically, the sensitivity to training data here is attached to the fact that positional node embeddings might overfit to the training probes while not capturing its symmetry (invariance) properties of their orbits. Next, in the experimental section, we show how this result indeed translates into practice: (Strictly) Positional node embeddings struggle with generalizing to node pairs not probed during training.

6 Experimental Results

We now evaluate our theoretical findings on identifying and estimating the counterfactual query from Equation (5) with experimental data using our supervised learning solution from Equation (13). Concretely, we investigate how accurately (strictly) positional node embeddings (Definition 10), structural node embeddings (Definition 6), and pairwise invariant embeddings (Definition 5) can estimate our counterfactual query from Equation (5). We are interested in empirically answering the following four questions.

(Q1) How do structural node embeddings perform when our test set contains tuples (U, V) that are node-wise isomorphic to the ones used in training $(I^{(t_1)}, J^{(t_1)}), \ldots, (I^{(t_M)}, J^{(t_M)})$? In Theorem 3 we proved that structural node embeddings are not unbiased estimators of Equation (5) in this setting. We are now interested in evaluating the practical impact of this result.

(**Q2**) How do (strictly) positional node embeddings perform when our test set contains tuples (U, V) such that neither U or V participated in the probes $(I^{(t_1)}, J^{(t_1)}), \ldots, (I^{(t_M)}, J^{(t_M)})$ used for training? In Figure 5 we can see that (strictly) positional node embeddings cannot learn a single representation for multiple pairs. Does such a property translate into poor practical performance when new pairs are tested? That is, in an inductive setting where we have to extrapolate our knowledge from training, do we need to explore symmetries?

(**Q3**) How do (strictly) positional and structural (node and pairwise) embedding solutions perform in real-world scenarios where we are not aware of the underlying causal model? Here we are interested in testing all the assumptions involved in Proposition 1 in real-world data.

(**Q4**) (Appendix F) In Figure 4 we can see a central assumption in Proposition 1: The (interventional) link formation process can be retrieved by a common set of parameters representing the structure of the node pairs. Here we ask: Is this assumption reasonable with real-world data? To answer this, we test whether structurally similar node pairs have the same probe outcome distribution (Equation (1)).

We evaluate **Q1** and **Q2** in a knowledge base completion task, **Q2** in a covariance matrix estimation task and **Q2**, **Q3** and **Q4** in two user-item recommendation tasks. For each scenario, we present the problem as the counterfactual query in causal link prediction and discuss the conditions for its identification. Next, we briefly introduce the graph embedding methods we evaluate in the proposed tasks.

Node embedding methods. As representatives of (strictly) **positional node embedding** methods we use Nonnegative Matrix Factorization (NMF) [43], SVD [30] and positional GCN [41], which is obtained by using node identifiers as node features. As for **structural node embeddings**, we choose the classical GCN [41] model and the more recent UniMP [66] GNN. Specifically for the knowledge base experiment, we also choose some popular **knowledge graph embeddings**, namely TransE [7], DistMult [78] and ComplEX [71]. For all methods, we use multi-layer perceptrons as link functions (ρ in Equation (13)). Implementation details can be found in Appendix G.

Structural pairwise embedding methods. The importance of structural pairwise embeddings has been showed only recently in [68] and thus its literature is still underexplored. Here, we use SEAL [81] and Neo-GNNs [80] as representative methods. SEAL extends GNNs to output structural pairwise embeddings. The idea behind SEAL was generalized and named labeling trick [82]. Essentially, methods of this kind apply a GNN over the graph but mark (only) the two nodes in the represented pair. In its simplest form, this marking can be done by a single bit added to the node features —indicating whether the node is in the pair. Here, we also consider what we call Label-GCN —an approximate and more scalable application of the labeling trick. Originally, the labeling trick computes the GNN representation of each pair separately. This process is considerably less scalable than a standard GNN model where only one GNN computation is performed to represent all pairs. To reduce the computational burden of the labeling trick, in Label-GCN we mark all the pairs in a mini-batch of our (stochastic) optimization procedure. By using a small mini-batch size, we expect representations from a sparse graph to not interfere with each other —providing a good approximation of the labeling trick method. Note that during test we also need to use the same small mini-batch size. Implementation details can be found in Appendix G.

6.1 Impact on knowledge graph queries

Here we consider the causal link prediction task in knowledge graphs. In this scenario, we are interested in questions as those described earlier:

> "Given our current knowledge about the world and some fact-finding mission that added a new piece of information, *i.e.* new relations, what other pieces would have been added had we investigated these relations in other parts of the graph?"

To answer queries of this type, we construct a knowledge base which comprises 100 non-isomorphic family trees, built using the methods provided in Hohenecker et al. [32], whose procedure we follow verbatim. In total, the dataset contains 28 relation types. The `parentOf` relation represents the current knowledge about the world and it is used to construct the observed graph $G^{(t_0)}$. The goal is to predict the other family relations,

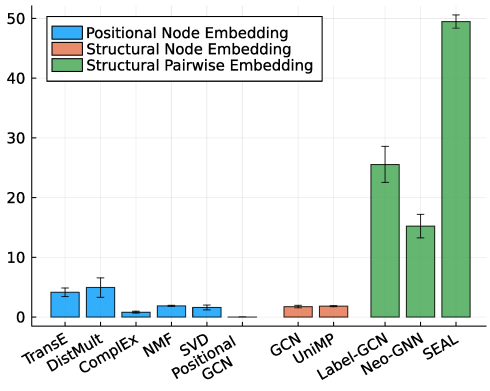

Figure 6: Results for the family tree dataset. We present the average and the standard deviation over five runs for the Hits@500 (out of 10,538 potential edges) metric in each method. Note that the larger ↑ the better the performance.

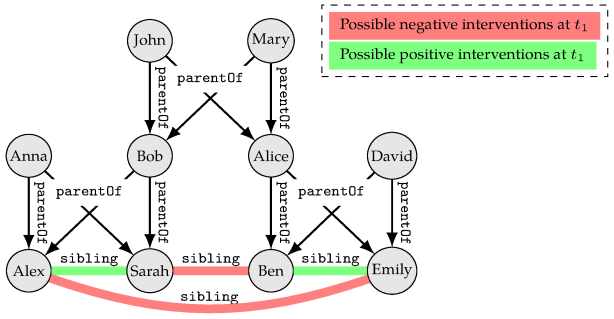

Figure 7: Synthetic illustrative example of a knowledge graph with two isomorphic subtrees as the knowledge base observed at (pre-trial) time t_0. In green and red we highlight possible new relations or non-existent relations after an intervention at (post-trial) time t_1.

which can be entirely determined just from the `parentOf` relations, *i.e.* the observed graph $G^{(t_0)}$. Thus, it is straightforward that our model completely satisfies all the conditions from Theorem 2 and Proposition 1.

In this dataset, 30% of the family trees contain two isomorphic subtrees, that we split between train and test such that all the relations within one subtree form the training probes and all the relations in the isomorphic subtree form the test relations. Structural node embeddings assign the same representation to isomorphic nodes and therefore will not distinguish, for example, `girlCousinOf` and `sisterOf` relations, see Alex and Sarah vs. Alex and Emily in the pairwise symmetric graph from Figure 7. This last example is indeed the type of situation from Theorem 3 where we evaluate **Q1**. To induce this kind of problem in the task, we consider test non-links having each end-point in a different subtree. Strictly positional node embeddings can distinguish those two relations, but they may not make the same prediction for isomorphic pairs due to higher variance (Figure 5, **Q2**), and therefore the predictions in one subtree might be different from the ones in the other isomorphic subtree, impacting generalization. As predicted by Theorem 3, structural pairwise embeddings do not suffer from these issues, as shown by the results in Figure 6, where both (strictly) positional node embeddings and structural node embeddings obtain poor performances when compared to structural pairwise embeddings. We show the results using the Hits@500 metric where we get the 500 node pairs with the highest probability of forming a link given by the model and compute the ratio that do form a link according to the data, *i.e.*, their labels are positive.

6.2 Impact on covariance matrix estimation

We now present a task mostly unexplored in the causal link prediction literature. Consider observing samples from a joint distribution. For instance, each sample is a list of measurements from a subject (e.g., a patient) and the distribution is over their corresponding (medical) attributes. We can then use these samples to construct an estimated covariance matrix for the attribute variables. We now present the following counterfactual query:

> "At a later time we want to refine our estimated covariance matrix by adding more data from more subjects but only for a subset of the attribute variables. What would have been the re-estimated covariance values for the attributes that we did not collect extra data?"

Note that in this setting the experiments are made simultaneously, since a node in a graph is an attribute variable and for a subject all pairs measurements are collected at the same time (Assumption 5). Further, since each subject is (presumably) not related to other subjects, measurements do not interfere with the ones from other subjects (Assumption 5). Here, the only assumption needed is that exogenous variables from subjects are i.i.d. Theorem 2. Such an assumption should hold if subjects are selected independently at

random, *i.e.*, they are not related. Time exchangeability (cf. Assumption 2) is also satisfied since the links are created simultaneously and the order in which they change (estimation is refined) is irrelevant due to the samples being i.i.d.. Further, since the covariance matrix forms a complete graph, the non-link ignorability assumption (cf. Assumption 3) is not needed to apply our result. Finally, the identifier of an attribute, *i.e.*, the name we give to it, does interfere in its probe outcomes (cf. Assumption 4).

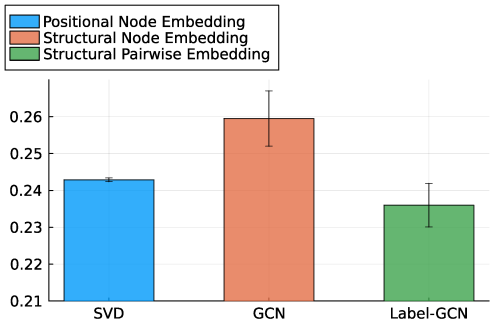

Figure 8: Results for the covariance matrix dataset. We present the average and the standard deviation over five runs for the MSE metric in each method. Note that the smaller ↓ the better the performance.

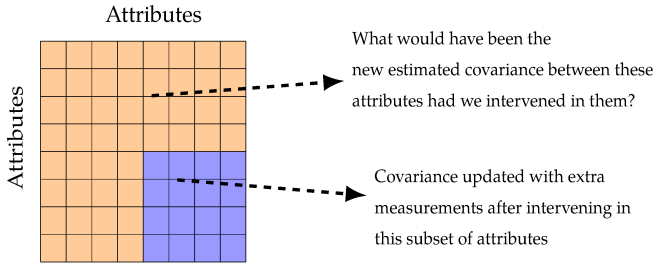

Figure 9: An illustration of the process of acquiring more data (measurements) for a submatrix —induced by a subset of attributes— of our original estimated covariance matrix. We then query how would the rest of the matrix be updated had we acquired data for the other attributes as well.

Baselines. We do not evaluate Neo-GNNs and SEAL here since these methods consider the neighborhood of a pair and, in this task, the equivalent graph $G^{(t_0)}$ is often complete. Hence, SEAL would be equivalent to Label-GCN and Neo-GNNs would be equivalent to a standard GNN. Therefore, we chose to evaluate only Label-GCN as a structural pairwise embedding method. Moreover, the sparsity assumption to train Label-GCN does not hold and, hence, we compute each pair representation separately in the mini-batch —which is equivalent to performing the labeling trick. Our experiments compare Label-GCN to one method of each other node embedding type.

6.3 Impact on recommender systems

Here we consider two user-item recommendation datasets, the Amazon Electronics (AE) [74] and the Last FM (LFM) [49] datasets. We built the observed graph $G^{(t_0)}$ from user-item interactions occurring between 11/24/2015 and 12/24/2015 in AE and between 2007 and 2013 in LFM. Now, given these observed graphs we face the following counterfactual problem:

> " At a later time, we can probe by exposing a subgroup of users to items. After observing their interactions, we wonder what would the other users consume had we exposed the items to them?"

In both datasets we consider the subgroup of male users as the one we probe in. At test time, our counterfactual queries are about female users. We use in both train and test interactions happening between 12/24/2015 and 12/31/2015 for AE and in 2014 for LFM. Note that in such datasets we have the outcome of probes that turn out to create edges, but not of probes with nonedge outcomes. To overcome this issue, we select nonedge probe examples by sampling nodes uniformly at random. Since the graph is quite sparse, with high probability the two nodes are unrelated and would have formed a nonedge had we probed in them.

In these datasets we will evaluate tasks that do not necessarily fulfill all assumptions needed for our application of causal lifting. Apart from fulfilling identifier exchangeability (cf. Assumption 4), it is unclear whether Assumptions 1 to 3 and 5 hold (**Q3**) in this application. For instance, in LFM a user listening to a song from an artist might influence their choice to listen to another song of the same artist —possibly violating Assumption 5 in the experiments. Then, given this scenario, how do (strictly) positional node embeddings and structural (node and pairwise) embeddings perform? The results in Figures 10 and 11 show that even under these assumption violations, *pairwise embedding methods consistently outperform node embedding methods* (the results use the Hits@500 and Hits@50 metrics as described in Section 6.1). In particular, testing pairs of

users unseen in training (probes) makes (strictly) positional node embeddings have poor performance in both tasks. This observed behavior can be attributed to the predictor needing to use symmetries between males and females for knowledge transfer in this task, rather than relying solely on a node's positional embedding. Interestingly, in the AE dataset (Figure 10) we see that the GCN structural node embedding, although worse than structural pairwise embeddings, still achieves reasonable performance, but its performance is poor in the LFM dataset (cf. Figure 11), a result that can be attributed to a higher bias of structural node embedding (Theorem 3) in the LFM dataset. Overall, the results clearly show that structural pairwise embeddings are superior for these tasks than node embeddings.

Baselines. Note that in user-item recommendation tasks the graph is bipartite. Thus, Neo-GNN, a method that considers the one-hop neighborhood of a pair would be equivalent to a standard GNN model. Thus, we choose not to evaluate it here.

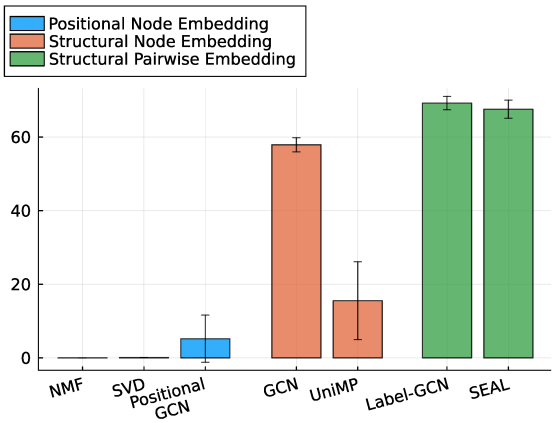

Figure 10: Results for the AE dataset. We present the average and the standard deviation over five runs for the Hits@500 metric (out of 6,138 potential edges) in each method. Note that the larger ↑ the better the performance.

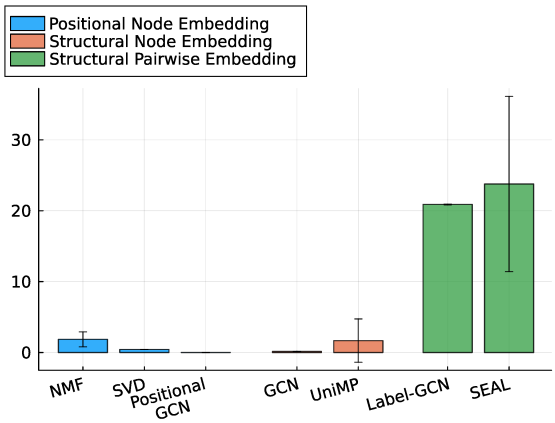

Figure 11: Results for the LFM dataset. We present the average and the standard deviation over five runs for the Hits@50 metric (out of 4,256 potential edges) in each method. Note that the larger ↑ the better the performance.

7 Conclusions

In this work we have shown how invariances can play a key role in identifying counterfactual queries with interventional data. Classical back-door adjustments for link prediction rely on over-simplifying causal independence assumptions (DAG-based constructions). Here, we take a different route: Consider a universal class of causal models and define causal mechanism invariances. By doing so, we show the SCM acquires a property that we denote *causal lifting*, which enables performing symmetry-based adjustments, *i.e.*, use orbits instead of covariates.

By approaching causal link prediction with symmetry-based adjustments, we are able to consider a wide class of *possibly path-dependent* causal models. To the best of our knowledge, our work is the first to consider such a setting and, moreover, to show the importance of invariant pairwise embeddings as estimators. We hope to shed light into this (until now) overlooked area of invariant representation learning and to incorporate symmetry-based adjustments in future causal literature works.

8 Acknowledgments

This work was funded in part by the National Science Foundation (NSF) awards CAREER IIS-1943364, CCF-1918483, and CNS-2212160 and an Amazon Research Award. L. Cotta is funded in part, by a postdoctoral fellowship provided by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute. 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.

References

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

[2] Albert, R. and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47.

[3] Bakshy, E., Eckles, D., Yan, R., and Rosenn, I. (2012). Social influence in social advertising: evidence from field experiments. In *Proceedings of the 13th ACM conference on electronic commerce*, pages 146–161.

[4] Barceló, P., Kostylev, E. V., Monet, M., Pérez, J., Reutter, J. L., and Silva, J. P. (2020). The logical expressiveness of graph neural networks. In *ICLR*.

[5] Bareinboim, E., Correa, J., Ibeling, D., and Icard, T. (2020). On Pearl's hierarchy and the foundations of causal inference. *ACM special volume in honor of Judea Pearl*.

[6] Bonner, S. and Vasile, F. (2018). Causal embeddings for recommendation. In *Proceedings of the 12th ACM conference on recommender systems*, pages 104–112.

[7] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *NeurIPS*, volume 26.

[8] Boschin, A. (2020). Torchkge: Knowledge graph embedding in python and pytorch. *arXiv preprint arXiv:2009.02963*.

[9] Brand, J. E. and Xie, Y. (2007). 11. identification and estimation of causal effects with time-varying treatments and time-varying outcomes. *Sociological methodology*, 37(1):393–434.

[10] Candes, E. J. and Plan, Y. (2010). Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936.

[11] Chen, S., Moore, J. L., Turnbull, D., and Joachims, T. (2012). Playlist prediction via metric embedding. In *Proceedings of the 18th ACM SIGKDD*, pages 714–722.

[12] Chen, Z., Villar, S., Chen, L., and Bruna, J. (2019). On the equivalence between graph isomorphism testing and function approximation with GNNs. In *NeurIPS*, pages 15868–15876.

[13] Christakis, N. A. and Fowler, J. H. (2007). The spread of obesity in a large social network over 32 years. *New England journal of medicine*, 357(4):370–379.

[14] Clauset, A., Moore, C., and Newman, M. E. (2008). Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101.

[15] Dua, D. and Graff, C. (2017). UCI machine learning repository.

[16] Dziugaite, G. K. and Roy, D. M. (2015). Neural network matrix factorization. *arXiv preprint arXiv:1511.06443*.

[17] Eckles, D., Karrer, B., and Ugander, J. (2017). Design and analysis of experiments in networks: Reducing bias from interference. *Journal of Causal Inference*, 5(1).

[18] Erdos, P. and Renyi, A. (1963). Asymmetric graphs. *Acta Mathematica Academiae Scientiarum Hungarica*, 14(3-4):295–315.

[19] Fabrikant, A., Koutsoupias, E., and Papadimitriou, C. H. (2002). Heuristically optimized trade-offs: A new paradigm for power laws in the internet. In *ICALP*, pages 110–122. Springer.

[20] Feng, S., Li, X., Zeng, Y., Cong, G., Chee, Y. M., and Yuan, Q. (2015). Personalized ranking metric embedding for next new poi recommendation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[21] Ferrara, A., Espín-Noboa, L., Karimi, F., and Wagner, C. (2022). Link recommendations: Their impact on network structure and minorities. *arXiv preprint arXiv:2205.06048*.

[22] Fey, M. and Lenssen, J. E. (2019). Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*.

[23] Fisher, R. A. (1922). On the interpretation of χ 2 from contingency tables, and the calculation of p. *Journal of the royal statistical society*, 85(1):87–94.

[24] Flum, J. and Grohe, M. (2006). Parameterized complexity theory springer.

[25] Gao, J. and Ribeiro, B. (2022). On the equivalence between temporal and static equivariant graph representations. In *ICML*.

[26] Ghasemian, A., Hosseinmardi, H., Galstyan, A., Airoldi, E. M., and Clauset, A. (2020). Stacking models for nearly optimal link prediction in complex networks. *PNAS*, 117(38):23393–23400.

[27] Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD*, pages 855–864.

[28] Guo, R., Li, J., and Liu, H. (2020). Learning individual causal effects from networked observational data. In *Proceedings of the 13th International Conference on WSDM*, WSDM '20, page 232–240, New York, NY, USA. Association for Computing Machinery.

[29] Guvenir, H. A., Acar, B., Demiroz, G., and Cekin, A. (1997). A supervised machine learning algorithm for arrhythmia analysis. In *Computers in Cardiology 1997*, pages 433–436. Ieee.

[30] Halko, N., Martinsson, P. G., and Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288.

[31] Henderson, K., Gallagher, B., Eliassi-Rad, T., Tong, H., Basu, S., Akoglu, L., Koutra, D., Faloutsos, C., and Li, L. (2012). Rolx: structural role extraction & mining in large graphs. In *Proceedings of the 18th ACM SIGKDD*, pages 1231–1239.

[32] Hohenecker, P. and Lukasiewicz, T. (2020). Ontology reasoning with deep neural networks. *Journal of Artificial Intelligence Research*, 68.

[33] Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137.

[34] Holland, P. W. and Leinhardt, S. (1981). An exponential family of probability distributions for directed graphs. *Journal of the american Statistical association*, 76(373):33–50.

[35] Hong, G. (2015). *Causality in a social world: Moderation, mediation and spill-over*. John Wiley & Sons.

[36] Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. (2020). Open graph benchmark: Datasets for machine learning on graphs. In *NeurIPS*.

[37] Ji, Y., Sun, A., Zhang, J., and Li, C. (2022). Recommender may not favor loyal users. *arXiv preprint arXiv:2204.05927*.

[38] Jiang, S. and Sun, Y. (2022). Estimating causal effects on networked observational data via representation learning. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 852–861.

[39] Joachims, T., London, B., Su, Y., Swaminathan, A., and Wang, L. (2021). Recommendations as treatments. *AI Magazine*, 42(3):19–30.

[40] Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43.

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

[42] Kohavi, R., Deng, A., Frasca, B., Longbotham, R., Walker, T., and Xu, Y. (2012). Trustworthy online controlled experiments: Five puzzling outcomes explained. In *Proceedings of the 18th ACM SIGKDD*, pages 786–794.

[43] Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

[44] Leung, M. P. (2022). Causal inference under approximate neighborhood interference. *Econometrica*, 90(1):267–293.

[45] Li, P., Wang, Y., Wang, H., and Leskovec, J. (2020). Distance encoding: Design provably more powerful neural networks for graph representation learning. In *NeurIPS*.

[46] Liben-Nowell, D. and Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031.

[47] Ma, J., Wan, M., Yang, L., Li, J., Hecht, B., and Teevan, J. (2022). Learning causal effects on hypergraphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 1202–1212, New York, NY, USA. Association for Computing Machinery.

[48] Manski, C. F. (2013). Identification of treatment response with social interactions. *The Econometrics Journal*, 16(1):S1–s23.

[49] Melchiorre, A. B., Rekabsaz, N., Parada-Cabaleiro, E., Brandl, S., Lesota, O., and Schedl, M. (2021). Investigating gender fairness of recommendation algorithms in the music domain. *Information Processing & Management*, 58(5):102666.

[50] Mnih, A. and Salakhutdinov, R. R. (2008). Probabilistic matrix factorization. In *NeurIPS*, pages 1257–1264.

[51] Newman, M. E. (2001). Clustering and preferential attachment in growing networks. *Physical review E*, 64(2):025102.

[52] 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. (2019). Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, volume 32.

[53] Pearl, J. (2009). *Causality*. Cambridge university press.

[54] 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. (2011). Scikit-learn: Machine learning in Python. *JMLR*, 12:2825–2830.

[55] Perc, M. (2014). The matthew effect in empirical data. *Journal of The Royal Society Interface*, 11(98):20140378.

[56] Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD*, pages 701–710.

[57] Poole, D. (2003). First-order probabilistic inference. In *IJCAI*, volume 3, pages 985–991.

[58] Puffert, D. J. (2002). Path dependence in spatial networks: the standardization of railway track gauge. *Explorations in Economic History*, 39(3):282–314.

[59] Quadrana, M., Cremonesi, P., and Jannach, D. (2018). Sequence-aware recommender systems. *ACM Computing Surveys (CSUR)*, 51(4):1–36.

[60] Radhakrishnan, A., Stefanakis, G., Belkin, M., and Uhler, C. (2022). Simple, fast, and flexible framework for matrix completion with infinite width neural networks. *PNAS*, 119(16):e2115064119.

[61] Ren, H., Hu, W., and Leskovec, J. (2020). Query2box: Reasoning over knowledge graphs in vector space using box embeddings. *arXiv preprint arXiv:2002.05969*.

[62] Rosenbaum, P. R. (2007). Interference between units in randomized experiments. *Journal of the american statistical association*, 102(477):191–200.

[63] Rubin, D. B. (2005). Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331.

[64] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2009). The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

[65] Sherman, E. and Shpitser, I. (2020). Intervening on network ties. In *Uncertainty in Artificial Intelligence*, pages 975–984. PMLR.

[66] Shi, Y., Huang, Z., Feng, S., Zhong, H., Wang, W., and Sun, Y. (2020). Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*.

[67] Spearman, C. (1927). *The abilities of man.* Macmillan.

[68] Srinivasan, B. and Ribeiro, B. (2020). On the equivalence between positional node embeddings and structural graph representations. In *ICLR*.

[69] Sun, Z., Deng, Z.-H., Nie, J.-Y., and Tang, J. (2019). Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*.

[70] Trouillon, T., Dance, C. R., Gaussier, É., Welbl, J., Riedel, S., and Bouchard, G. (2017). Knowledge graph completion via complex tensor factorization. *The JMLR*, 18(1):4735–4772.

[71] Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., and Bouchard, G. (2016). Complex embeddings for simple link prediction. In *ICML*, pages 2071–2080. Pmlr.

[72] Ugander, J., Karrer, B., Backstrom, L., and Kleinberg, J. (2013). Graph cluster randomization: Network exposure to multiple universes. In *Proceedings of the 19th ACM SIGKDD*, pages 329–337.

[73] Van den Broeck, G. and Niepert, M. (2015). Lifted probabilistic inference for asymmetric graphical models. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[74] Wan, M., Ni, J., Misra, R., and McAuley, J. (2020). Addressing marketing bias in product recommendations. In *Proceedings of the 13th international conference on WSDM*, pages 618–626.

[75] Wang, Y., Liang, D., Charlin, L., and Blei, D. M. (2020). Causal inference for recommender systems. In *Fourteenth ACM Conference on Recommender Systems*, pages 426–431.

[76] Woolley, H. T. and Fischer, C. R. (1914). Mental and physical measurements of working children. *The Psychological Monographs*, 18(1):i.

[77] Xu, S., Ge, Y., Li, Y., Fu, Z., Chen, X., and Zhang, Y. (2021). Causal collaborative filtering. *arXiv preprint arXiv:2102.01868*.

[78] Yang, B., Yih, W.-t., He, X., Gao, J., and Deng, L. (2014). Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

[79] Yuan, Y., Altenburger, K., and Kooti, F. (2021). Causal network motifs: Identifying heterogeneous spillover effects in a/b tests. In *Proceedings of the Web Conference 2021*, pages 3359–3370.

[80] Yun, S., Kim, S., Lee, J., Kang, J., and Kim, H. J. (2021). Neo-gnns: Neighborhood overlap-aware graph neural networks for link prediction. In *NeurIPS*, volume 34.

[81] Zhang, M. and Chen, Y. (2018). Link prediction based on graph neural networks. *arXiv preprint arXiv:1802.09691*.

[82] Zhang, M., Li, P., Xia, Y., Wang, K., and Jin, L. (2021). Labeling trick: A theory of using graph neural networks for multi-node representation learning. In *NeurIPS*, volume 34, pages 9061–9073.

[83] Zhao, T., Liu, G., Wang, D., Yu, W., and Jiang, M. (2022). Learning from counterfactual links for link prediction. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th ICML*, volume 162 of *Proceedings of Machine Learning Research*, pages 26911–26926. PMLR.

[84] Zhou, Y., Kutyniok, G., and Ribeiro, B. (2022). Ood link prediction generalization capabilities of message-passing gnns in larger test graphs. In *NeurIPS*.

[85] Zhu, J., Lu, X., Heimann, M., and Koutra, D. (2021). Node proximity is all you need: Unified structural and positional node and graph embedding. In *Proceedings of the 2021 SDM*, pages 163–171. Siam.

A Notation and Background

Graph notation. We consider a graph G with adjacency $a \in \mathbb{A}^{n\times n}$, where each entry $a_{ij} \in \mathbb{A}$ belongs to an arbitrary domain $\mathbb{A}$ with at least two elements. Without loss of generality, we consider the node set $V = [n] := \{1, \dots, n\}$. In simple unattributed graphs a is a binary matrix, while for general attributed (multi)graphs a can be seen as a tensor, where its third mode encodes edge and node attributes[3]. Note that following this definition a completely defines G. Since our exposition is mostly agnostic to its third mode, unless otherwise stated we consider $a \in \mathbb{A}^{n\times n}$. Further, we denote the value representing a non-link as $\mathbf{0} \in \mathbb{A}$. Finally, we use capital letters to denote the corresponding random variable of an observation, *e.g.*, A is a random variable of a. As such, if $a \in \mathbb{A}^{n\times n}$ was produced by some mechanism that contains some intrinsic randomness (even noise), then A describes the other possible outcomes with their respective probabilities.

Symmetry definitions. We denote the symmetric group of $[n]$ by $\mathbb{S}_n$, *i.e.*, the set of all permutations of $\{1, \dots, n\}$. Further, we let $\pi \cdot i$ be the mapping of $i \in [n]$ in permutation $\pi \in \mathbb{S}_n$ and $\pi^{-1} \cdot i$ its inverse, *i.e.*, the element that π maps to i. Finally, we let the action of π in a be denoted by $\pi \cdot a$, *i.e.*, $a_{ij} = (\pi \cdot a)_{\pi\cdot i\pi\cdot j}$. Note that $\pi \cdot a$ permutes the rows and columns of a (or first two modes if a is a tensor) according to the permutation π. We say that $G \cong H$ are isomorphic graphs if there exists a permutation $\pi \in \mathbb{S}_n$ such that the adjacency of H is equal to $\pi \cdot a$.

Apart from isomorphism between graphs, we now define isomorphism between nodes and node pairs in the same graph. For that, we need to first define the automorphism group of a graph G: $\mathrm{Aut}(G) \colon \{\pi \in \mathbb{S}_n \colon \pi \cdot a = a\}$. Note that $\mathrm{Aut}(G)$ defines the set of permutations that map the graph G to itself. Now, we say nodes i and j in G are isomorphic[4] $i \cong j$ if there exists a permutation $\pi \in \mathrm{Aut}(G)$ where $j = \pi \cdot i$. Similarly, two node pairs $(i, j), (u, v)$ in G are isomorphic $(i, j) \cong (u, v)$ if there exists a permutation $\pi \in \mathrm{Aut}(G)$ where $(u, v) = \pi \cdot (i, j)$. Finally, we denote by $\mathcal{O}_i : \{j \in V \colon j \cong i\}$ the orbit of node i in G and by $\mathcal{O}_{ij} : \{(u, v) \in V^2 \colon (u, v) \cong (i, j)\}$ the orbit of the node pair (i, j) in G. Overall, the orbit of a node (or node pair) defines the set of nodes (or node pairs) isomorphic to it (including itself). Note that the orbit of a node or of a node pair in G contains elements other than themselves only if $|\mathrm{Aut}(G)| > 1$. We call graphs that satisfy such property *symmetric graphs*.

Now that we have defined symmetry between graphs, nodes and node pairs, we can turn our attention to a symmetry in the graph distribution. The random variable of a graph is finite (jointly) exchangeable if any two isomorphic graphs are generated with the same probability. That is, node identifiers, that are what distinguish isomorphic graphs, are not relevant to the task at hand. In Definition 11 we overload this definition and define finite exchangeable graphs as graphs that come from such a distribution. Throughout this work we consider finite exchangeable graphs as input data to our problem, *i.e.*, $G^{(t_0)}$ in Equations (2) and (5). Although this is an assumption about the data distribution, finite exchangeability is also what distinguishes graph from sequence data. If node identifiers are not arbitrary, we can treat the graph's adjacency as a long sequence of edges $(a_{11}, a_{12}, \dots)$. Thus, finite exchangeability can be seen as assuming that our input data is a graph.

Definition 11 (Finite exchangeable graphs)**.** *We say that an adjacency matrix random variable A is finite (jointly) exchangeable if*

$$P(A = a) = P(A = \pi \cdot a), \quad \forall \pi \in \mathbb{S}_n, a \in \mathbb{A}^{n\times n}.$$

We overload finite (joint) exchangeability to ease notation and say that a graph G is finite exchangeable if its adjacency matrix random variable A is finite (jointly) exchangeable.

Causality definitions.

Any causal query can be seen as an inquiry about the underlying Structural Causal Model (SCM) of the task [5]. An SCM is a mathematical description of the mechanisms behind the data generating process of interest. We start by formally defining what an SCM is below.

Definition 12 (Structural Causal Model (SCM) [5])**.** *A Structural Causal Model (SCM) is a 4-tuple $\mathcal{C} = (\mathcal{U}, \mathcal{V}, \mathcal{F}, P(\mathcal{U}))$, where*

[3] The node attribute of node i can be encoded in its self-loop entry a_{ii}.

[4] In graph theory we also often say i is similar to j.

- $\mathcal{U}$ *is the set of external noise random variables —also called background or exogenous variables— which are generated by (unknown) mechanisms outside the model;*
- $\mathcal{V} = \{\mathcal{V}_1, \mathcal{V}_2 \ldots, \mathcal{V}_q\}$ *is the set of endogenous random variables, which are generated by variables inside the model* ($\mathcal{V} \cup \mathcal{U}$)*;*
- $\mathcal{F} = \{f_1, f_2 \ldots, f_q\}$ *is a set of functions defining a mapping between* $\mathcal{U}$ *and* $\mathcal{V}$ *with each* f_i *mapping from the domains of* Pa_i *and* $\mathcal{U}_{\mathcal{V}_i}$ *to* $\mathcal{V}_i$ *with* $\mathcal{U}_{\mathcal{V}_i} \subseteq \mathcal{U}, \mathrm{Pa}_i \subseteq \mathcal{V} \backslash \mathcal{V}_i$*. In practice, each* $f_i \in \mathcal{F}$ *is a mechanism that outputs the* i^{th} *endogenous variable* $\mathcal{V}_i$ *given its exogenous variables* $\mathcal{U}_{\mathcal{V}_i}$ *and its endogenous parent variables* Pa_i*, i.e.*

$$\mathcal{V}_i = f_i(\mathrm{Pa}_i, \mathcal{U}_{\mathcal{V}_i});$$

and

- $P(\mathcal{U})$ *is a probability distribution over* $\mathcal{U}$*.*

Note from above that an SCM $\mathcal{C}$ is a data generating process for $\mathcal{V}$ and as such induces an (associational) probability distribution $P(\mathcal{V} = v) = \sum_u \prod_{i|\mathcal{V}_i \in \mathcal{V}} P(v_i \mid \mathrm{pa}_i u_{\mathcal{V}_i}) P(u)$. We can then define the interventional distribution $P(\mathcal{V}(\mathcal{V}_i = v_i))$ as the distribution induced by an altered model $\mathcal{C}_{\mathcal{V}_i = v_i}$, where $\mathcal{C}_{\mathcal{V}_i = v_i}$ is exactly as $\mathcal{C}$ with the difference that $\mathcal{V}_i$ is replaced by a constant value v_i —we often denote such distribution by $P_{\mathcal{V}_i = v_i}(\mathcal{V})$. Now, the counterfactual distribution $P(\mathcal{V}(\mathcal{V}_i = v_i) \mid \mathcal{V} = v')$ can be defined as the distribution induced by the altered model $\mathcal{C}_{\mathcal{V}_i = v_i} \mid v'$, where $\mathcal{C}_{\mathcal{V}_i = v_i} \mid v'$ is exactly as $\mathcal{C}_{\mathcal{V}_i = v_i}$ with the difference that $P(\mathcal{U})$ is replaced by $P(\mathcal{U} \mid \mathcal{V} = v')$, *i.e.*, the exogenous variables distribution changes based on our observed evidence v' —we often denote such distribution by $P_{\mathcal{V}_i = v_i | v'}(\mathcal{V})$. Finally, note that the evidence v' can also be the outcome of an interventional distribution, as we have in Equation (2). In this case, what changes is how we update the exogenous variable distribution, *i.e.*, $P(\mathcal{U})$ is replaced by $P(\mathcal{U} \mid \mathcal{V}(\mathcal{V}_i = v_i) = v)$. From these definitions, we now have the tools to evaluate interventional and counterfactual quantities, as required by our task (cf. Equations (2) and (5)).

B Link Prediction through self-supervision

In what follows we formalize the **observational** task of predicting whether a node pair $(i, j) \in V^{(t_0)} \times V^{(t_0)}$ forms a link in $G^{(t_0)}$ in the context of Graph Neural Networks and Matrix Factorization. We will show how, for both methods, this task can be viewed as a self-supervised learning one. Let $A^{(t_0)}_{-ij}$ and $a^{(t_0)}_{-ij}$ be respectively the random variable $A^{(t_0)}$ and the adjacency matrix $a^{(t_0)}$ without the pair $(i, j) \in V^{(t_0)} \times V^{(t_0)}$. Note how this is different from assuming (i, j) is a nonedge (*i.e.*, that there is no edge between nodes i and j). The adjacency matrix $a^{(t_0)}_{-ij}$ has no information about $a^{(t_0)}_{ij}$, including whether it is an edge or a nonedge.

Relation to Graph Neural Networks (GNNs). Existing works in link prediction using GNN node embeddings make use of the graph structure to predict missing links [36]. To predict $A^{(t_0)}_{IJ}$, the embeddings of I and J are obtained by applying a GNN over the training graph $G^{(t_0)}$ with adjacency $a^{(t_0)}$. Thus, instead of learning the self-supervision task $P(A^{(t_0)}_{IJ} = a^{(t_0)}_{IJ} \mid A^{(t_0)}_{-IJ} = a^{(t_0)}_{-IJ})$, previous works on GNNs for link prediction are learning $P(A^{(t_0)}_{IJ} = a^{(t_0)}_{IJ} \mid A^{(t_0)} = a^{(t_0)})$. Note, however, that $A^{(t_0)}_{IJ}$ is contained in $A^{(t_0)}$, hence this is not a sound statistical learning task. Instead, we should remove the information about $A^{(t_0)}_{IJ}$ and thus use the self-supervised objective. In practice, we note that the information about $A^{(t_0)}_{IJ}$ is not directly encoded in GNN embeddings, *i.e.*, it is used implicitly via the message-passing scheme. Thus, most training procedures are not impacted when using $a^{(t_0)}$ instead of $a^{(t_0)}_{-IJ}$ to compute GNN embeddings.

Relation to matrix factorization. Matrix (or tensor) factorization methods are one of the most used tools for link prediction and clustering tasks in graphs. For undirected unattribued graphs, such methods learn a matrix of embeddings $\phi \in \mathbb{R}^{n \times d}$, where each row ϕ_i is the embedding of node i. For directed graphs, ϕ is a pair of such matrices, encoding the source and the target embedding of each node. Finally, heterogeneous graphs also add edge type embeddings in ϕ. How do these methods learn ϕ?

Matrix factorization-based models learn the joint distribution $P(A^{(t_0)} = a^{(t_0)} \mid \Phi = \phi)$. The key assumption here is edge (conditional) independence, that is

$$P(A^{(t_0)} = a^{(t_0)} \mid \Phi = \phi) = \prod_{(i,j) \in V^{(t_0)} \times V^{(t_0)}} P(A_{ij}^{(t_0)} = a_{ij}^{(t_0)} \mid \Phi = \phi). \tag{14}$$

These methods learn ϕ^* by maximizing the log-likelihood of Equation (14), a problem that can be written as

$$\phi^* = \operatorname*{argmax}_{\phi} \mathbb{E}_{(I,J)} \big[\log P(A_{IJ}^{(t_0)} = a_{IJ}^{(t_0)} \mid \Phi = \phi) \big], \tag{15}$$

with $(I, J) \sim \text{Uniform}(V^{(t_0)} \times V^{(t_0)})$.

Finally, what is the relationship between Equation (14) and the self-supervised learning task $P(A_{IJ}^{(t_0)} = a_{IJ}^{(t_0)} \mid A_{-IJ}^{(t_0)} = a_{-IJ}^{(t_0)})$? First, note that we can rewrite the self-supervised learning task when learning parameters ϕ as

$$P(A_{IJ}^{(t_0)} = a_{IJ}^{(t_0)} \mid A_{-IJ}^{(t_0)} = a_{-IJ}^{(t_0)}) = \int_{\phi} P(A_{IJ}^{(t_0)} = a_{IJ}^{(t_0)} \mid \Phi = \phi) P(\Phi = \phi \mid A_{-IJ}^{(t_0)} = a_{-IJ}^{(t_0)}) \, d\phi, \tag{16}$$

which can be expressed as the log-likelihood

$$\phi^* = \operatorname*{argmax}_{\phi} \mathbb{E}_{(I,J)} \big[\log \mathbb{E}_{\Phi = \phi \mid a_{-IJ}^{(t_0)}} [P(A_{IJ}^{(t_0)} = a_{IJ}^{(t_0)} \mid \Phi = \phi)] \big], \tag{17}$$

with $(I, J) \sim \text{Uniform}(V^{(t_0)} \times V^{(t_0)})$. We can see how the difference between the training objectives Equation (17) and Equation (15) is the expectation over the prior $P(\Phi = \phi \mid A_{-IJ}^{(t_0)} = a_{-IJ}^{(t_0)})$. Finally, if we assume a flat prior $P(\Phi = \phi \mid A_{-ij}^{(t_0)} = a_{-ij}^{(t_0)})$ over all $(i, j) \in V^{(t_0)} \times V^{(t_0)}$, the two objectives become the same. Thus, we can see how matrix factorization methods are simply ignoring the rest of the observed graph as an input signal to predict $A_{IJ}^{(t_0)}$.

C A Universal Family of Causal Models for Graphs

The conditions needed to perform counterfactual lifting in our task (Equations (2) and (5)) are with respect to its underlying Structural Causal Model (SCM), *i.e.*, the graph's causal generating process. To this end, here we present a universal family of SCMs for graphs, where we are able to define our task and derive sufficient conditions for counterfactual lifting and other identification results. Without loss of generality, we show how to generate $A^{(t_0)}$ —which completely defines the observed graph $G^{(t_0)}$. Note that unlike in usual causal models, we observe finite exchangeable graph data and thus our SCM needs to be finite exchangeable with respect to $A^{(t_0)}$. A finite (jointly) exchangeable SCM generates an observation $a^{(t_0)}$ with the same probability as $\pi \cdot a^{(t_0)}$ for any permutation $\pi \in \mathbb{S}_n$, *i.e.*, any two isomorphic graphs are generated with the same probability. Since an SCM generates every random variable as a function of its parents, causal models have an intrinsic (partial) ordering, *i.e.*, parents must be generated before their children. Thus, designing an SCM for our task implies generating (partially) ordered sequences of random variables. How can we go from (partially) ordered sequences to finite exchangeable random variables? Next, we define a family of SCMs with such property. Intuitively, our causal models achieve finite exchangeability by randomly reassigning node identifiers. Later in Theorem 1 (i)'s proof we formally show how this family of SCMs is indeed finite exchangeable.

We denote the proposed family of SCMs by $\mathbb{C}$. Each SCM $\mathcal{C} \in \mathbb{C}$ has a specific set of mechanisms $\mathcal{F}$ and exogenous variables distribution $P(\mathcal{U})$. To generate the observed graph, $\mathcal{C}$ has two stages: the data generating process (Definition 13) and the adjacency observation process (Definition 14). The data generating process outputs two infinite-size sequences $(\mathcal{X}^{(t)})_{t=1}^{\infty}, (\mathcal{E}^{(t)})_{t=1}^{\infty}$. Each $\mathcal{X}^{(t)}$ has the same support as the entries of $A^{(t_0)}$ ($\text{supp}(\mathcal{X}^{(t)}) = \mathbb{A}$), while each $\mathcal{E}^{(t)}$ defines a node pair ($\text{supp}(\mathcal{E}^{(t)}) = \mathbb{Z}^+ \times \mathbb{Z}^+$). Since it generates $(\mathcal{X}^{(t)})_{t=1}^{\infty}$ and $(\mathcal{E}^{(t)})_{t=1}^{\infty}$ sequentially, the SCM has a notion of time, where at time t it generates the value $\mathcal{X}^{(t)}$ of the interaction (or its absence) between $\mathcal{E}^{(t)}$. At a given observation time t_0, the adjacency observation process in Definition 14 generates $A^{(t_0)}$ by using $(\mathcal{E}^{(t)})_{t=1}^{t_0}$ to map $(\mathcal{X}^{(t)})_{t=1}^{t_0}$ to a matrix (or tensor) and jointly shuffling its rows and columns (or its first two modes).

Definition 13 (Data generating process)**.** *We start by defining $\mathcal{E}^{(1)} = (1, 1)$. Then, at time t, for two mechanisms $f_{\mathcal{X}}^{(t)}$ and $f_{\mathcal{E}}^{(t)}$ we define the recurrence relations*

$$\mathcal{E}^{(t)} = f_{\mathcal{E}}^{(t)}\Big((\mathcal{E}^{(r)})_{r=1}^{t-1}, (\mathcal{X}^{(r)})_{r=1}^{t-1}, \mathcal{U}_{\mathcal{E}}^{(t)}\Big),$$
$$\mathcal{X}^{(t)} = \begin{cases} f_{\mathcal{X}}^{(t)}\Big(\mathcal{U}_{\mathcal{X}}^{(t)}\Big), & \text{if } t = 1, \\ f_{\mathcal{X}}^{(t)}\Big((\mathcal{E}^{(r)})_{r=1}^{t}, (\mathcal{X}^{(r)})_{r=1}^{t-1}, \mathcal{U}_{\mathcal{X}}^{(t)}\Big), & \text{otherwise,} \end{cases}$$

where $\mathcal{U}_{\mathcal{X}}^{(t)} \sim P(\mathcal{U}_{\mathcal{X}}^{(t)} \mid \mathcal{E}^{(t)}, (\mathcal{U}_{\mathcal{X}}^{(m)})_{m=1}^{t-1})$ are sampled (given $\mathcal{E}^{(t)}$ and all previous exogenous variables) and $\mathcal{U}_{\mathcal{E}}^{(t)} \sim P(\mathcal{U}_{\mathcal{E}}^{(t)} \mid (\mathcal{U}_{\mathcal{E}}^{(m)})_{m=1}^{t-1})$ are exogenous variables sampled at time t. Note that the distribution of $\mathcal{U}_{\mathcal{X}}^{(t)}$ takes $\mathcal{E}^{(t)}$ as a parameter and thus the exogenous variable at time t is dependent not only on the previously generated exogenous variables, but also on the pair being generated[5]*. Finally,*

$$f_{\mathcal{X}}^{(1)} \colon \operatorname{supp}(\mathcal{U}_{\mathcal{X}}^{(1)}) \to \mathbb{A}$$
$$f_{\mathcal{X}}^{(t)} \colon \bigcup_{r=1}^{t} \Big(\big(\mathbb{Z}^+ \times \mathbb{Z}^+\big)^r \times \big(\mathbb{A}\big)^{r-1}\Big) \times \operatorname{supp}\Big(\mathcal{U}_{\mathcal{X}}^{(t)}\Big) \to \mathbb{A},\ t > 1,$$

are measurable maps and

$$f_{\mathcal{E}}^{(t)} \colon \Big(\mathbb{Z}^+ \times \mathbb{Z}^+\Big)^t \times \Big(\mathbb{A}\Big)^t \times \operatorname{supp}\Big(\mathcal{U}_{\mathcal{E}}^{(t)}\Big) \to \mathbb{Z}^+ \times \mathbb{Z}^+$$

is a measurable map such that $\max(\mathcal{E}^{(t)}) \leq \max\Big(\big(\max(\mathcal{E}^{(r)})\big)_{r=1}^{t-1}\Big) + 1$, i.e., the pair to be generated at time t can contain up to one node that has not yet been generated until time $t-1$.

The only mechanism restriction is a simple rule on $f_{\mathcal{E}}^{(t)}$: the pair to be generated can contain only up to one unseen node[6] and its identifier is the smallest positive integer not seen yet. This way, if n nodes have appeared in interactions at any point of time, their identifiers will be $[n] := \{1, \ldots, n\}$. Note, however, that we do not observe the sequences that Definition 13 outputs. Instead, we partially observe them as $G^{(t_0)}$, *i.e.*, a graph at some given point of time t_0. The process generating our observed data (Definition 14) takes as input $(\mathcal{E}^{(t)})_{t=1}^{t_0}, (\mathcal{X}^{(t)})_{t=1}^{t_0}$ from Definition 13 and outputs $A^{(t_0)}$. It samples a permutation π of the node identifiers and maps the sequence to the observed matrix (or tensor) $A^{(t_0)}$. Thus, t_0 and π are the observation parameters —when we observe the graph and what we see as arbitrary node identifiers. Note how in Definition 13 a pair (i, j) can be generated multiple times by $f_{\mathcal{E}}^{(t)}$. Therefore, the observation process only considers the most recent interaction occurred at time t° (cf. Definition 14).

Definition 14 (Adjacency observation process)**.** *Let $n^{(t_0)} := \max\Big(\big(\max(\mathcal{E}^{(t)})\big)_{t=1}^{t_0}\Big)$ be the number of different nodes appearing in pairs generated by $\mathcal{C}$ until time t_0. Then, we can generate $A^{(t_0)}$ by first sampling a permutation of the node identifiers*

$$\pi \sim \mathit{Uniform}(\mathbb{S}_{n^{(t_0)}})$$

and then assigning

$$A_{ij}^{(t_0)} = \begin{cases} \mathbf{0}, \text{if } (\pi^{-1} \cdot i, \pi^{-1} \cdot j) \notin \{\mathcal{E}^{(t)}\}_{t=1}^{t_0}, \\ \mathcal{X}^{(t_{ij}^\circ)}, t_{ij}^\circ := \max\big(\{t \colon \mathcal{E}^{(t)} = (\pi^{-1} \cdot i, \pi^{-1} \cdot j), 1 \leq t \leq t_0\}\big), \text{otherwise.} \end{cases}$$

Undirected graphs. *Note that if $G^{(t_0)}$ is an undirected graph we have that all mechanisms $f_{\mathcal{X}}^{(t)}, t \geq 1$ are invariant to the order of the input pairs, i.e., each $\mathcal{E}^{(t)}$ is treated as a set rather than a tuple. Finally, we have an extra step here setting $A_{ji}^{(t_0)}$ to $A_{ij}^{(t_0)}$ if $(\pi_i^{-1}, \pi_j^{-1}) \in \{\mathcal{E}^{(t)}\}_{t=1}^{t_0}$. In the case that (j, i) was also generated by the SCM, i.e., $(\pi^{-1} \cdot j, \pi^{-1} \cdot i) \in \{\mathcal{E}^{(t)}\}_{t=1}^{t_0}$, we set $A_{ji}^{(t_0)}$ to $A_{ij}^{(t_0)}$ only if $t_{ij}^\circ > t_{ji}^\circ$.*

[5]Note that in fact $\mathcal{U}_{\mathcal{X}}^{(t)}$ is not an exogenous variable, since it is modeled inside our SCM, but we refer to it as such for the sake of simplicity.

[6]An unseen node in the SCM is a node which has not appeared in any generated pairs until time t.

We thus define a graph SCM $\mathcal{C} \in \mathbb{C}$ as the coupling of Definitions 13 and 14. At this point, it is worth taking a moment to understand $\mathcal{C}$ —see Figure 2 for a sample generation of a graph with four nodes. The data generating process (Definition 13) is an underlying evolving process, deciding at each time step which node pairs will be assigned a link (and its value) or a non-link. We do not observe the execution of Definition 13 or the observation parameter π, *i.e.*, we are not aware of nodes' original identifiers. We only observe the graph's adjacency $A^{(t_0)}$. Finally, note that a nonedge entry in $A^{(t_0)}$ does not imply that the pair was generated as a nonedge. It can also be that it was not yet generated by the causal model. What does this mean in practice? For instance, consider a streaming platform where we do not observe an interaction between user i and movie j. In this case, either user i does not know movie j is in the platform or i has actively made the decision to not watch j. The underlying causal model explaining the lack of an interaction is hidden from us. Later we show how this notion between nonedges and pairs yet not generated by the SCM is central to identify our counterfactual task.

Generating temporal and dynamic graphs. Up until now, we have referred to our observed graph $G^{(t_0)}$ as a static graph. That is, a graph where all edges are observed together at once with no temporal information. However, note that although there is no explicit notion of time in $G^{(t_0)}$, its underlying data generating process (Definition 13) is temporal. Because of such feature, our model can actually represent not only static, but also temporal and dynamic graphs. In temporal graphs we observe the edge creation time. In this case, our model can output in $A_{ij}^{(t_0)}$ the most recent time step t° (cf. Definition 14) together with the interaction value. In dynamic graphs, an edge can be created multiple times, disappear, take new values and so on. In this setting we would then not only observe the time of an interaction, but also all past interactions. Note that although $f_{\mathcal{E}}^{(t)}$ can generate an interaction between the same pair multiple times, the observation model only outputs the most recent one at t°. On the other hand, since $f_{\mathcal{X}}^{(t)}$ takes as input all previous interactions, it can copy the history of interactions between i and j to $A_{ij}^{(t_0)}$ (together with their time steps) making $A^{(t_0)}$ a sample of a dynamic graph.

Exchangeability and expressive power. After designing our family of SCMs $\mathbb{C}$, we turn to the two central theoretical questions around it: i. Is $\mathbb{C}$ finite exchangeable? ii. How expressive is $\mathbb{C}$? In Theorem 1 we show that i. any SCM $\mathcal{C} \in \mathbb{C}$ generates any two isomorphic graphs with the same probability, hence the entire family $\mathbb{C}$ is finite exchangeable and ii. there exists an SCM $\mathcal{C} \in \mathbb{C}$ that generates every pair of non-isomorphic graphs (with countable domain) with different probabilities. Ultimately, Theorem 1(i, ii) proves that $\mathbb{C}$ is both a finite exchangeable and universal family of graph SCMs. Next, we present the proofs of items i. and ii. from Theorem 1.

C.1 Proof of Theorem 1

Theorem 1 (Universality of our graph SCM)**.** *Let $\mathbb{C}$ be the family of SCMs as defined in Definitions 13 and 14 in Appendix C and $\mathbb{A}$ be the domain of the entries of the adjacency matrices of the graphs generated by it. Then,*

i. For every SCM $\mathcal{C} \in \mathbb{C}$ at an arbitrary observation time $t_0 \geq 0$, $\mathcal{C}$ always generates observed graphs $G^{(t_0)}$ where $P(A^{(t_0)} = a) = P(A^{(t_0)} = a')$ for any two isomorphic graphs with adjacencies $a, a' \in \mathbb{A}$;

ii. For all finite (jointly) exchangeable graph distributions $P(A^{(t_0)})$, if $\mathbb{A}$ is a countable set there exists an SCM $\mathcal{C} \in \mathbb{C}$ and an observation time $t_0 \geq 0$ that induces it.

Proof.
i. Let $\mathcal{C}(u, \pi, t_0)$ be the output of an SCM $\mathcal{C} \in \mathbb{C}$ with input exogenous variables $u \in \text{supp}(\mathcal{U})$, permutation π and observation time t_0. Note that given these three variables assignments $\mathcal{C}$ is a deterministic mapping to an observed graph a. We need to prove that any $\mathcal{C}(u, \pi, t_0)$ gives isomorphic graphs the same probability. Now, the probability of any model $\mathcal{C} \in \mathbb{C}$ generating graphs $a, a' \in \mathbb{A}$ is

$$P(A^{(t_0)} = a \mid \mathcal{C}) = \sum_{t_0=1}^{\infty} \frac{1}{|\mathbb{S}_n|} \sum_{\pi \in \mathbb{S}_n} \int_{u \in \text{supp}(\mathcal{U})} \mathbb{1}(\mathcal{C}(u, \pi, t_0) = a) P(\mathcal{U} = u) du, \tag{18}$$

and

$$P(A^{(t_0)} = a' \mid \mathcal{C}) = \sum_{t_0=1}^{\infty} \frac{1}{|\mathbb{S}_n|} \sum_{\pi \in \mathbb{S}_n} \int_{u \in \text{supp}(\mathcal{U})} \mathbb{1}(\mathcal{C}(u, \pi, t_0) = a') P(\mathcal{U} = u) du \tag{19}$$

respectively. Note that since a and a' are isomorphic, we can rewrite a' as $\pi' \cdot a$ for some $\pi' \in \mathbb{S}_n$. Now, since $\mathbb{S}_n$ is a group, for every $\pi \in \mathbb{S}_n$ there exists another $\pi^\dagger$ such that $\pi = \pi^\dagger \circ \pi'$. Thus, for every $\pi \in \mathbb{S}_n$ we can define another permutation $\pi^\star := \pi^\dagger \circ \pi'$ giving us $\pi \cdot a = \pi^\star \cdot a'$. Thus, whenever $\mathcal{C}(u, \pi, t_0) = a$ there exists a $\pi^\star$ such that $\mathcal{C}(u, \pi^\star, t_0) = a'$. As a result of such bijection, since the other terms match in Equations (18) and (19), we have that $P(A^{(t_0)} = a \mid \mathcal{C}) = P(A^{(t_0)} = a' \mid \mathcal{C})$.

ii. We now show that for any finite graph (finite jointly exchangeable random array) A there exists an SCM $\mathcal{C} \in \mathbb{C}$ and a time $t_0 \in \mathbb{R}$ such that $A \overset{d}{=} A^{(t_0)}$, where $A^{(t_0)}$ is the graph generated by $\mathcal{C}$ at time t_0. Since $\mathbb{A}$ is countable, let's enumerate all graphs $A_1, A_2, \ldots \in \mathbb{A}$ such that $\mathbb{A} = \cup_{i=1}^{\infty}\{A_i\}$, noting that graph with distinct indices may be isomorphic. Now, we partition the interval $[0, 1)$ into intervals $U_j = \left[\sum_{i=1}^{j} P(A_i), \sum_{i=1}^{j+1} P(A_i)\right)$, where we define $P(A_0) = 0$, noting that by the total law of probability $[0, 1) = \cup_{j=1}^{\infty} U_j$. Following Definition 13 we can construct arbitrary functions $f_{\mathcal{X}}^{(t)}, f_{\mathcal{E}}^{(t)}, t = 1, \ldots, t_0$, for SCM $\mathcal{C} \in \mathbb{C}$ such that with probability U_j we generate deterministic sequences $\mathcal{U}_{\mathcal{X}}^{(1)}, \ldots, \mathcal{U}_{\mathcal{X}}^{(t_0)}$ and $\mathcal{U}_{\mathcal{E}}^{(1)}, \ldots, \mathcal{U}_{\mathcal{E}}^{(t_0)}$ that will exactly generate the edges of A_j (in the same order). The exchangeability of $A^{(t_0)}$ comes from Definition 13, which permutes the node ids at the observation time t_0.

□

From the above we highlight the result in (ii). Our family of SCMs is not trivial nor attached to a restricted set of distributions. There exists a causal model in it that is able to generate all non-isomorphic graphs with different probabilities, *i.e.*, it can distinguish them and thus we call it an universal class of causal models for graphs. This result comes at the cost of a (possibly[7]) prohibitively large amount of causal dependencies between variables (see Figure 2 for an illustration). Since in practice it is hard to have assumptions removing them, we maintain such structural dependencies and still identify and estimate Equations (2) and (5) by posing invariance restrictions to the causal mechanisms and some experimental (probe) conditions.

D Interventional Lifting for Link Prediction

Now that we have formalized or class of SCMs, we can precisely describe Assumptions 1 to 4 and Assumption 5.

Assumption 1 (Time gap ignorability)**.** *We say that an SCM $\mathcal{C} \in \mathbb{C}$ satisfies time gap ignorability for observation time t_0 and first experiment time t_1 if*

$$f_{\mathcal{X}}^{(t_1)}\left((\mathcal{E}^{(t)})_{t=1}^{t_1}, (\mathcal{X}^{(t)})_{t=1}^{t_1-1}, \mathcal{U}_{\mathcal{X}}^{(t_1)}\right) = f_{\mathcal{X}}^{(t_0+1)}\left((\mathcal{E}^{(t)})_{t=1}^{t_0+1}, (\mathcal{X}^{(t)})_{t=1}^{t_0}, \mathcal{U}_{\mathcal{X}}^{(t_0+1)}\right).$$

Assumption 2 (Time exchangeability)**.** *We say that an SCM $\mathcal{C} \in \mathbb{C}$ satisfies time exchangeability for observation time if*

$$f_{\mathcal{X}}^{(t_0+1)}\left((\mathcal{E}^{(t)})_{t=1}^{t_0+1}, (\mathcal{X}^{(t)})_{t=1}^{t_0}, \mathcal{U}_{\mathcal{X}}^{(t_0+1)}\right) = f_{\mathcal{X}}^{(t_0+1)}\left((\mathcal{E}^{(\pi_t)})_{t=1}^{t_0+1}, (\mathcal{X}^{(\pi_t)})_{t=1}^{t_0}, \mathcal{U}_{\mathcal{X}}^{(t_0+1)}\right),$$
$$\forall \pi \in \mathbb{S}_{t_0+1} \colon \pi_{t_0+1} = t_0 + 1.$$

Assumption 3 (Non-link ignorability)**.** *Let $\mathcal{N}^{(t_0)} := \left\{t \in [t_0] \colon \mathcal{X}^{(t)} = \mathbf{0}\right\}$ be the set of time steps (until t_0) where non-links were created. Then, we say that an SCM $\mathcal{C} \in \mathbb{C}$ satisfies non-link ignorability for observation time t_0 if its mechanism $f_{\mathcal{X}}^{(t_0+1)}$ is invariant to the removal of non-links from the input sequence, i.e.,*

$$f_{\mathcal{X}}^{(t_0+1)}\left((\mathcal{E}^{(t)})_{t=1}^{t_0+1}, (\mathcal{X}^{(t)})_{t=1}^{t_0}, \mathcal{U}_{\mathcal{X}}^{(t_0+1)}\right)$$
$$= f_{\mathcal{X}}^{(t_0+1)}\left(((\mathcal{E}^{(t)})_{t=1}^{t_0+1})_{-\mathcal{N}^{(t_0)}}, ((\mathcal{X}^{(t)})_{t=1}^{t_0})_{-\mathcal{N}^{(t_0)}}, \mathcal{U}_{\mathcal{X}}^{(t_0+1)}\right).$$

[7] We say possibly since in practice the mechanisms of the true and unknown underlying SCM might be ignoring parts of the input and thus removing such dependencies.

Assumption 4 (Identifier exchangeability)**.** *We say that an SCM $\mathcal{C} \in \mathbb{C}$ satisfies identifier exchangeability for observation time t_0 and first experiment time t_0+1 if*

$$f_{\mathcal{X}}^{(t_0+1)}\Big((\mathcal{E}^{(t)})_{t=1}^{t_0+1}, (\mathcal{X}^{(t)})_{t=1}^{t_0}, \mathcal{U}_{\mathcal{X}}^{(t_0+1)}\Big) = f_{\mathcal{X}}^{(t_0+1)}\Big(((\pi_i,\pi_j)\colon (i,j) := \mathcal{E}^{(t)})_{t=1}^{t_0+1}, (\mathcal{X}^{(t)})_{t=1}^{t_0}, \mathcal{U}_{\mathcal{X}}^{(t_0+1)}\Big), \quad \forall \pi \in \mathbb{S}_n.$$

Definition 4 (Non-interfering probes)**.** *We say that a sequence of probes in M pairs $\big((I^{(t_m)}, J^{(t_m)})\big)_{m=1}^{M}$ is non-interfering if the following invariance holds*

$$f_{\mathcal{X}}^{(t_m)}\Big((\mathcal{E}^{(t)})_{t=1}^{t_0}, ((I^{(t_{m'})}, J^{(t_{m'})}))_{m'=1}^{m}\big), \big((\mathcal{X}^{(t)})_{t=1}^{t_0}, ((I^{(t_{m'})}, J^{(t_{m'})}))_{m'=1}^{m-1}\big), \mathcal{U}_{\mathcal{X}}^{(t_m)}\Big) = f_{\mathcal{X}}^{(t_0+1)}\Big(((\mathcal{E}^{(t)})_{t=1}^{t_0}, (I^{(t_1)}, J^{(t_1)})), (\mathcal{X}^{(t)})_{t=1}^{t_0}, \mathcal{U}_{\mathcal{X}}^{(t_0+1)}\Big).$$

D.1 Proof of Theorem 2

Theorem 2 (Invariances for interventional lifting in link prediction)**.** *Let $\mathbb{C}$ be the family of SCMs as defined in Definitions 13 and 14 in Appendix C and $\mathbb{A}$ be the domain of the entries of the adjacency matrices of the graphs generated by it. Then,*

i. if $\mathcal{C} \in \mathbb{C}$ has the mechanism invariances described in Assumptions 1 to 4 and exogenous variable independence at time t_0+1, i.e., $\mathcal{U}_{\mathcal{X}}^{(t_0+1)} \perp\!\!\!\perp (\mathcal{U}_{\mathcal{X}}^{(t)})_{t=1}^{t_0} \mid (I,J)$ in Definition 13. Then, the effect of an intervention in $\mathcal{C}$ can be equivalently described by Figure 3's causal DAG, where $(I,J) \sim \mu(G^{(t_0)})$ is the node pair we intervene in, $\mathcal{O}_{IJ}^{(t_0)} := \{\pi \cdot (I,J)\colon \pi \in \mathrm{Aut}(G^{(t_0)})\}$ is the set of all structurally indistinguishable pairs to (I,J) in $G^{(t_0)}$, and $\mathbf{W}_{\mathcal{O}_{IJ}^{(t_0)}}$ is a latent variable tied to $\mathcal{O}_{IJ}^{(t_0)}$ (the orbit of (I,J) in $G^{(t_0)}$).

ii. Under the conditions in (i) and assuming the extra symmetry $P(\mathcal{U}_{(I,J)}) = P(\mathcal{U}_{\pi\cdot(I,J)})$ in Figure 3's DAG, we have the following interventional lifting result (Definition 2), where $\forall \pi \in \mathrm{Aut}(G^{(t_0)})$

$$P\left(Y_{IJ}^{(t_1)}\right) = P\Big(A_{\mathcal{E}^{(t_1)}}^{(t_1)}\Big(\mathcal{E}^{(t_1)} = (I,J)\Big) \mid G^{(t_0)}\Big) = P\Big(A_{\mathcal{E}^{(t_1)}}^{(t_1)}\Big(\mathcal{E}^{(t_1)} = \pi\cdot(I,J)\Big) \mid G^{(t_0)}\Big) = P\left(Y_{\pi\cdot(I,J)}^{(t_1)}\right).$$

Proof.
ii. We start by noting that from Definition 13, we have that

$$Y_{IJ}^{(t_1)} = f_{\mathcal{X}}^{(t_1)}\Big((\mathcal{X}^{(t)})_{t=1}^{t_1-1}, (\mathcal{E}^{(t)})_{t=1}^{t_1}, \mathcal{U}_{\mathcal{X}}^{(t_1)}\Big),$$

where $\mathcal{E}^{(t_1)} = (I,J)$ is the intervened random variable while $(\mathcal{X}^{(t)})_{t=1}^{t_1-1} \sim P((\mathcal{X}^{(t)})_{t=1}^{t_1-1} \mid G^{(t_0)})$ and $(\mathcal{E}^{(t)})_{t=1}^{t_1-1} \sim P((\mathcal{E}^{(t)})_{t=1}^{t_1-1} \mid G^{(t_0)})$ are (random) sequences that must have generated $G^{(t_0)}$ until t_0. We start by considering Assumption 1, which directly reduces the above problem to an intervention at the next time step t_0+1

$$Y_{IJ}^{(t_1)} = f_{\mathcal{X}}^{(t_1)}\Big((\mathcal{X}^{(t)})_{t=1}^{t_0}, (\mathcal{E}^{(t)})_{t=1}^{t_0+1}, \mathcal{U}_{\mathcal{X}}^{(t_0+1)}\Big),$$

where $\mathcal{E}^{(t_0+1)} = (I,J)$ is the intervened random variable while $(\mathcal{X}^{(t)})_{t=1}^{t_0} \sim P((\mathcal{X}^{(t)})_{t=1}^{t_0} \mid G^{(t_0)})$ and $(\mathcal{E}^{(t)})_{t=1}^{t_0} \sim P((\mathcal{E}^{(t)})_{t=1}^{t_0} \mid G^{(t_0)})$ are (random) sequences that must have generated $G^{(t_0)}$.

Further, from Theorem 2's statement, we have that the exogenous variables sampled before time t_0+1 are independent from $\mathcal{U}_{\mathcal{X}}^{(t_0+1)}$ given (I,J), *i.e.*, $P(\mathcal{U}_{\mathcal{X}}^{(t_0+1)} \mid (\mathcal{U}_{\mathcal{X}}^{(t)})_{t=1}^{t_0}, (I,J)) = P(\mathcal{U}_{\mathcal{X}}^{(t_0+1)} \mid (I,J))$. Thus, we will be assuming $\mathcal{U}_{(I,J)} = \mathcal{U}_{\mathcal{X}}^{(t_0+1)}$ to prove (iii). from Theorem 2 it suffices to show that there exists a mechanism $h\colon \mathrm{supp}(\mathbf{W}_{\mathcal{O}_{IJ}^{(t_0)}}) \times \mathrm{supp}(\mathcal{U}_{\mathcal{X}}^{(t_0+1)}) \to \mathbb{A}$ such that

$$\begin{aligned} &h(\mathbf{W}_{\mathcal{O}_{ij}^{(t_0)}}, u_{\mathcal{X}}^{(t_0+1)}) = f_{\mathcal{X}}^{(t_0+1)}\Big((x^{(t)})_{t=1}^{t_0}, (e^{(t)})_{t=1}^{t_0+1}, u_{\mathcal{X}}^{(t_0+1)}\Big), \\ &\forall x^{(t)} \in \mathrm{supp}(\mathcal{X}^{(t)}), \forall e^{(t)} \in \mathrm{supp}(\mathcal{E}^{(t)}), t = 1,\dots,t_0, \forall u_{\mathcal{X}}^{(t_0+1)} \in \mathrm{supp}(\mathcal{U}_{\mathcal{X}}^{(t_0+1)}), \\ &\forall 1 \le n^{(t_0)} \le t_0, \forall a^{(t_0)} \in \mathbb{A}^{n^{(t_0)}\times n^{(t_0)}}, \forall (i,j) \in V^{(t_0)} \times V^{(t_0)}, \text{ with } e^{(t_0+1)} = (i,j). \end{aligned} \tag{20}$$

Let us now define the equivalence relation $\sim$ on the set of sequences $\Xi^{(t_0)} := \text{supp}\Big((\mathcal{X}^{(t)})_{t=1}^{t_0}, (\mathcal{E}^{(t)})_{t=1}^{t_0+1}\Big)$:

$$\Big((x^{(t)})_{t=1}^{t_0},(e^{(t)})_{t=1}^{t_0+1}\Big) \sim \Big((x^{(t)'})_{t=1}^{t_0}, (e^{(t)'})_{t=1}^{t_0+1}\Big) \text{ if } \exists\, \pi \in \mathbb{S}_{n^{(t_0)}}, \exists\, \pi^\dagger \in \mathbb{S}_{t_0+1} \colon \pi^\dagger \cdot (t_0+1) = t_0+1$$
$$\text{such that } \Big(\big((x^{(t)})_{t=1}^{t_0}\big)_{-\mathcal{N}^{(t_0)}}, \big((e^{(t)})_{t=1}^{t_0+1}\big)_{-\mathcal{N}^{(t_0)}}\Big)$$
$$= \Big(\big((x^{(\pi^\dagger\cdot t)'})_{t=1}^{t_0}\big)_{-\mathcal{N}^{(t_0)}}, \big((\pi \cdot e^{(\pi^\dagger\cdot t)'})_{t=1}^{t_0+1}\big)_{-\mathcal{N}^{(t_0)}}\Big),$$

where $(\cdot)_{-\mathcal{N}^{(t_0)}}$ removes the non-links from the sequence as in Assumption 3. Let $[\cdot]$ denote the equivalence class of a sequence and $[\Xi^{(t_0)}] := \Big\{\Big[\big((x^{(t)})_{t=1}^{t_0}, (e^{(t)})_{t=1}^{t_0+1}\big)\Big] \colon \big((x^{(t)})_{t=1}^{t_0}, (e^{(t)})_{t=1}^{t_0+1}\big) \in \Xi^{(t_0)}\Big\}$ be the set of all equivalence classes in $\Xi^{(t_0)}$. Then, it directly follows from the invariances in Assumptions 2 to 4 that there exists a function $g \colon [\Xi^{(t_0)}] \to \mathbb{A}$ such that, $\forall\, \big((x^{(t)})_{t=1}^{t_0}, (e^{(t)})_{t=1}^{t_0+1}\big) \in \Xi^{(t_0)}$,

$$g\Big(\Big[\big((x^{(t)})_{t=1}^{t_0}, (e^{(t)})_{t=1}^{t_0+1}\big)\Big], \mathcal{U}_{\mathcal{X}}^{(t_0+1)}\Big) = f_{\mathcal{X}}^{(t_0+1)}\Big((x^{(t)})_{t=1}^{t_0}, (e^{(t)})_{t=1}^{t_0+1}, \mathcal{U}_{\mathcal{X}}^{(t_0+1)}\Big), \tag{21}$$

where, as defined earlier, $[\cdot]$ denotes the equivalence class of the sequences.

Now, we are ready to prove that there exists a bijective mapping from $[\Xi^{(t_0)}]$ to the set of all input orbits $\mathbb{O}^{(t_0)} = \{\mathcal{O}_{ij}^{(t_0)} \colon n^{(t_0)} = 1, \cdots, t_0\, a^{(t_0)} \in \mathbb{A}^{n^{(t_0)}\times n^{(t_0)}}, (i,j) \in V^{(t_0)} \times V^{(t_0)}\}$. More specifically, consider the bijection as taking an arbitrary temporal sequence $\big((x^{(t)})_{t=1}^{t_0}, (e^{(t)})_{t=1}^{t_0}\big) \in \big[\big((x^{(t)})_{t=1}^{t_0}, (e^{(t)})_{t=1}^{t_0}\big)\big]$, applying Definition 14 and computing the orbit of $e^{(t_0+1)} = (i,j)$ in the generated graph $G^{(t_0)}$. It is straightforward from Definition 14 that any other temporal sequence in $\big[\big((x^{(t)})_{t=1}^{t_0}, (e^{(t)})_{t=1}^{t_0+1}\big)\big]$ also generates $\mathcal{O}_{ij}^{(t_0)}$. Also, for an arbitrary graph $G^{(t_0)}$ and $(i,j) \in V^{(t_0)} \times V^{(t_0)}$, consider the temporal sequence $\big((x^{(t)})_{t=1}^{t_0}, (e^{(t)})_{t=1}^{t_0}\big)$ where for any $t^\dagger \in \{1, \ldots, t_0\}$, s.t. either (a) $\exists e^{(t^\dagger)} \in E^{(t_0)}, x^{(t^\dagger)} = a^{(t_0)}_{e^{(t^\dagger)}}$, or (b) $\nexists e^{(t^\dagger)} \in E^{(t_0)}$, and $x^{(t')} = \mathbf{0}$. It is clear from Definition 14 that this procedure would generate any graph isomorphic to $G^{(t_0)}$ simply by performing a permutation $\pi \in \mathbb{S}_{n^{(t_0)}}$. If π is the identity permutation we generate the orbit $\mathcal{O}_{ij}^{(t_0)}$ with $\big((x^{(t)})_{t=1}^{t_0}, (e^{(t)})_{t=1}^{t_0+1}\big)$ where $e^{(t_0+1)} = (i,j)$. Thus, Definition 14 defines a surjection from $[\Xi^{(t_0)}]$ to $\mathbb{O}^{(t_0)}$. Next, we will show that it is also an injection.

Now, consider applying Definition 14 to an arbitrary temporal sequence $\big((x^{(t)'})_{t=1}^{t_0}, (e^{(t)'})_{t=1}^{t_0}\big)$, with $e^{(t_0+1)'} = (i',j') \in V^{(t_0)} \times V^{(t_0)}$. This temporal sequence defines a (static) graph G'. Let $\mathcal{O}'_{i'j'}$ be the orbit of (i',j') in G'. To prove Definition 14 is also an injection, we need to show that if $\mathcal{O}'_{i'j'} = \mathcal{O}_{ij}^{(t_0)}$, then $\big((x^{(t)'})_{t=1}^{t_0}, (e^{(t)'})_{t=1}^{t_0}\big) \in \big[\big((x^{(t)})_{t=1}^{t_0}, (e^{(t)})_{t=1}^{t_0}\big)\big]$. If $\mathcal{O}'_{i'j'} = \mathcal{O}_{ij}^{(t_0)}$, then the graphs are isomorphic $G' \cong G^{(t_0)}$. Then, apart from having the same number of nodes and edges, there exists a permutation $\pi^+ \in \mathbb{S}_{n^{(t_0)}}$ such that $a' = \pi^+ \cdot a^{(t_0)}$. Now, let $\bar{\pi} \in \mathbb{S}_{n^{(t_0)}}$ be the permutation sampled in Definition 14 to generate $a^{(t_0)}$ from an arbitrary sequence $\big((x^{(t)})_{t=1}^{t_0}, (e^{(t)})_{t=1}^{t_0+1}\big) \in \big[\big((x^{(t)})_{t=1}^{t_0}, (e^{(t)})_{t=1}^{t_0+1}\big)\big]$ and $\pi' \in \mathbb{S}_{n^{(t_0)}}$ the one used to generate G'. Then, we can define the permutation $\pi^\star := \bar{\pi}^{-1} \circ (\pi^+)^{-1} \circ \pi'$ and use it to match the two edge sets $\big\{(x^{(t)}, e^{(t)}) \colon t = 1, \cdots, t_0, x^{(t)} \neq \mathbf{0}\big\} = \big\{(x^{(t)'}, \pi^\star \cdot e^{(t)'}) \colon t = 1, \cdots, t_0, x^{(t)} \neq \mathbf{0}\big\}$. Finally, we can define a permutation $\pi^\dagger \in \mathbb{S}_{t_0+1}$ where $\pi \cdot t' = t$ for $\pi^\star \cdot e^{(t')'} = e^{(t)}$. We then have

$$G' \cong G^{(t_0)} \implies \big((x^{(t)})_{t=1}^{t_0}, (e^{(t)})_{t=1}^{t_0+1}\big)_{-\mathcal{N}^{(t_0)}} = \big((x^{(t)'})_{t=1}^{t_0}, (\pi^\star \cdot e^{(\pi^\dagger\cdot t)'})_{t=1}^{t_0+1}\big)_{-\mathcal{N}^{(t_0)}}$$
$$\implies \big((x^{(t)'})_{t=1}^{t_0}, (e^{(t)'})_{t=1}^{t_0+1}\big) \in \big[\big((x^{(t)})_{t=1}^{t_0}, (e^{(t)})_{t=1}^{t_0+1}\big)\big],$$

which implies that there exists a bijective mapping $q \colon [\Xi^{(t_0)}] \to \mathbb{O}^{(t_0)}$.

Finally, we can now define $\mathbf{W}_{\mathcal{O}_{ij}^{(t_0)}}$ as a bijective mapping $\mathbf{W}_{\mathcal{O}_{ij}^{(t_0)}} \colon \mathbb{O}^{(t_0)} \to \mathbb{R}^d$. Since q and $\mathbf{W}_{\mathcal{O}_{ij}^{(t_0)}}$ are bijective, they are invertible and thus we can define $h(a,b) = g(q^{-1} \circ \mathbf{W}^{-1}_{\mathcal{O}_{ij}^{(t_0)}}(a), b)$ and Equation (20) follows from Equation (21).

ii. From i. we have that the input to $Y_{IJ}^{(t_1)}$ and $Y_{\pi(I,J)}^{(t_1)}$ only differ on their input exogenous variables, which are sampled from their marginal distributions. From the theorem statement, their marginal distributions are the same and thus the random variables are equal everywhere.

□

D.2 Proof of Corollary 1

Corollary 1 follows directly from the DAG in Theorem 2 and the backdoor adjustment, see 3.2 in [53].

D.3 Proof of Corollary 2

Note that the definition of non-interfering probes (cf. Assumption 5) and the fact that exogenous variables are i.i.d. make the entire set of random variables $\{Y^{(t_m)}_{I^{(t_m)}J^{(t_m)}} : 1 \leq m \leq M\}$ also i.i.d.. Then, Corollary 2 follows directly from Corollary 1.

D.4 Proof of Proposition 1

Proof. Here we are left to show that under the stated conditions the causal DAG from Figure 3 can be equivalently represented by the one in Figure 4. Note that in Figure 3 $A^{(t_1)}_{IJ}$ is generated by a mechanism f that takes as input $\mathbf{W}_{\mathcal{O}^{(t_0)}_{IJ}}$ and $\mathcal{U}_{(I,J)}$. Now, from Definition 4 we know that there exists a set of weights $\mathbb{W}'_{\Gamma^\star}$ that makes Γ assign a unique representation to each orbit of nodes. Therefore, there exists a surjection $s\colon \{\Gamma(i,j,a^{(t_0)};\mathbf{W}'_{\Gamma^\star})\colon (i,j) \in V^{(t_0)} \times V^{(t_0)}\} \to \{\mathbf{W}_{\mathcal{O}^{(t_0)}_{ij}} \colon (i,j) \in V^{(t_0)} \times V^{(t_0)}\}$. Hence, we can define $A^{(t_1)}_{IJ}$ in Figure 4 as $A^{(t_1)}_{IJ} = f(s(\Gamma(I,J,a^{(t_0)};\mathbf{W}'_{\Gamma^\star})), \mathcal{U}_{(I,J)})$ and by the definition of s and f have the same SCM as Figure 3.

□

E Graph Embeddings for Causal Link Prediction

E.1 Proof of Theorem 3

Proof. From the definition of Definition 5, we can see that most-expressive pairwise representations are under structural (joint) pairwise representations. Therefore, it follows from [12] that with an expressive enough link function, *e.g.*, multi-layer perceptron with a sufficient number of neurons, we can achieve zero-bias in the task. Now, we are left to show that representations using structural node representations cannot achieve zero-bias for in some outcome distributions.

From Definition 6 we know that if the node pairs are node-wise isomorphic, *i.e.*, $i \cong u, j \cong v$, we have that

$$Z(i, a^{(t_0)}; \mathbf{W}_Z) = Z(u, a^{(t_0)}; \mathbf{W}_Z),$$
$$Z(j, a^{(t_0)}; \mathbf{W}_Z) = Z(v, a^{(t_0)}; \mathbf{W}_Z).$$

Note that from the definition of isomorphic nodes there exists a permutation $\pi \in \mathrm{Aut}(G^{(t_0)})$ and a (possibly other) permutation π' such that $u = \pi_i, v = \pi'_j$. Then, by defining Γ as the concatenation ($[\cdot,\cdot]$) of the two structural node embeddings it follows that if $u \cong i, j \cong v$ we have that

$$\rho\left(\left[Z(i, a^{(t_0)}; \mathbf{W}_Z), Z(j, a^{(t_0)}; \mathbf{W}_Z)\right]; \mathbf{W}_\rho\right) = \rho\left(\left[Z(u, a^{(t_0)}; \mathbf{W}_Z), Z(v, a^{(t_0)}; \mathbf{W}_Z)\right]; \mathbf{W}_\rho\right). \tag{22}$$

Now, in a pairwise symmetric graph there are at least two node pairs that are node-wise isomorphic, *i.e.*, $u \cong i, j \cong v$, but not isomorphic, *i.e.*, $(i,j) \ncong (u,v)$. That is, in such graphs the above equality holds for (i,j) and (u,v) and any choice of parameters $\mathbf{W}_Z, \mathbf{W}_\rho$ and link function ρ. However, since $(i,j) \ncong (u,v)$, there exist distributions such that $P(Y^{(t_1)}_{ij} = e) \neq P(Y^{(t_1)}_{uv} = e)$ for some $e \in \mathbb{A}$. Finally, from Equation (22) we know that structural embedding models cannot output different distributions for $Y^{(t_1)}_{ij}$ and $Y^{(t_1)}_{uv}$ and thus cannot achieve zero-bias for such an outcome distribution.

Now, to show that the bound is relevant to our task it is left for us to prove that there exists at least one pairwise symmetric graph $G^{(t_0)}$ generated by a model $\mathcal{C} \in \mathbb{C}$. Note that the model restrictions are about the generation process after time t_0, so we can leverage Theorem 2 directly. It follows from Theorem 2's proof that the results i. and ii. also hold for graphs $G^{(t_0)}$ observed at a fixed time t_0 when $n \leq \sqrt{t_0}$. Consider a graph $G^{(t_0)}, n \leq \sqrt{t_0}$ that has zero probability in every $\mathcal{C} \in \mathbb{C}$. Now, if $G'^{(t_0)}, n \leq \sqrt{t_0}$ is a graph non-isomorphic to

it, it follows from Theorem 2 that there exists a model $\mathcal{C}' \in \mathbb{C}$ that generates them with different probabilities, that is, $G'^{(t_0)}$ can be generated by $\mathcal{C}'$. Therefore, Theorem 2 guarantees that at most one graph with $n \leq \sqrt{t_0}$ (and the others isomorphic to it) cannot be generated by any model $\mathcal{C} \in \mathbb{C}$ at time t_0. In [85][Theorem 4.2] the authors show that any graph with two isomorphic components has nodes i, j, u, v satisfying Equation (22). Thus, since for a number of nodes $n \geq 6$ this class has more than one graph, there exists a $G^{(t_0)}$ with nodes i, j, u, v generated by a model $\mathcal{C} \in \mathbb{C}$ with $t_0 \geq 36$ where Equation (22) is satisfied.

□

E.2 Proof of Theorem 4

Proof. Let P_π be the permutation matrix of $\pi \in \mathbb{S}_n$ and $\theta_i^{(\text{SVD},r)}$ the SVD embedding of node i with respect to the right eigenvectors and $\theta_i^{(\text{SVD},\ell)}$ with respect to the left. Remember that the embedding of a node θ_i is considered as the concatenation of both $\theta_i^{(\text{SVD},r)}$ and $\theta_i^{(\text{SVD},\ell)}$.

- **Same embeddings** $\implies$ $P_\pi aa^T = aa^T P_\pi = aa^T$ and $P_\pi a^T a = a^T a P_\pi = a^T a$.

  Let $\Omega(G) := \{\pi \in \mathbb{S}_n \colon \pi_i = j, \theta_i^{(\text{SVD},r)} = \theta_j^{(\text{SVD},r)}, \theta_i^{(\text{SVD},\ell)} = \theta_j^{(\text{SVD},\ell)}\}$ be the set of permutations that map nodes to other nodes with the same SVD embeddings. Then, for every eigenvector x with corresponding eigenvalue λ of aa^T and $\pi \in \Omega(G)$,

  $$aa^T x = \lambda x,$$

  and since $P_\pi x = x$ we have

  $$aa^T P_\pi x = \lambda x,$$

  and

  $$P_\pi aa^T x = \lambda x.$$

  That is, $P_\pi aa^T$, $aa^T P_\pi$ and aa^T all have the same set of right eigenvectors and corresponding eigenvalues, thus

  $$P_\pi aa^T = aa^T P_\pi = aa^T.$$

  Note that the same procedure can be applied to the eigenvectors of $a^T a$ and thus

  $$P_\pi aa^T = P_\pi aa^T = aa^T, P_\pi a^T a = a^T a P_\pi = a^T a, \ \forall \pi \in \Omega.$$

- **Same embeddings** $\impliedby$ $P_\pi aa^T = P_\pi aa^T = aa^T$ and $P_\pi a^T a = a^T a P_\pi = a^T a$.

  Then, for every eigenvector x with corresponding eigenvalue λ of aa^T and $\pi \in \omega(G)$,

  $$aa^T x = \lambda x,$$

  multiply by P_π on both sides

  $$P_\pi aa^T x = \lambda P_\pi x.$$

  Now, since $P_\pi aa^T = aa^T$ we have that

  $$aa^T x = \lambda P_\pi x,$$

  which implies that $P_\pi x = x$ for every eigenvector x of aa^T and $\pi \in \omega(G)$. Note that again the exact same procedure can be applied to $a^T a$ and thus $\omega(G) = \Omega(G)$.

- $P_\pi aa^T = aa^T, P_\pi a^T a = a^T a \iff a_{kv} = a_{\ell v}, a_{vk} = a_{v\ell} \quad \forall v \in V, k, \ell \in V \colon k \cong \ell.$

  – First, if $aa^T = aa^T P_\pi = P_\pi aa^T$ for $\pi \in \Omega(G)$,

  $$(aa^T)_{ii} = \sum_{v \in V} a_{iv} a_{iv},$$

  and

  $$(aa^T)_{\pi_i i} = \sum_{v \in V} a_{\pi_i v} a_{iv},$$

  which implies that $a_{iv} = a_{\pi_i v}$ $\forall v \in V, \pi \in \Omega(G)$.

– Now, we apply the same procedure leveraging $P_\pi a^T a$

$$(a^T a)_{ii} = \sum_{v \in V} a_{vi} a_{vi},$$

and

$$(a^T a)_{\pi i} = \sum_{v \in V} a_{v\pi_i} a_{vi},$$

which implies that $a_{vi} = a_{v\pi_i}\ \forall v \in V, \pi \in \Omega(G)$.

– The other direction is satisfied straightforwardly.

Note from the last item that $\Omega(G) \subseteq \mathbb{S}_n$ is the set of permutations that swaps only nodes with identical neighborhoods. From that it follows that $\Omega(G) \subseteq \mathrm{Aut}(G)$ is a subset of the automorphisms of G as well, thus only isomorphic nodes can get the same SVD embeddings. Overall, two nodes get the same SVD embedding if they have the exact same neighborhood.

□

E.3 Proof of Corollary 3

Proof. As mentioned in the main text, Corollary 3 follows directly from Theorem 4 and the definition of strictly positional node embeddings (cf. Definition 10). □

F Can structure capture the link formation process of real-world networks?

Here we investigate **Q4** from Section 6, *i.e.*, the extent to which a shared set of parameters representing node pairs' structure can encapsulate the link formation process of real-world graphs. More specifically, we are interested in testing whether $Y_{IJ}^{(t_1)} \overset{d}{=} Y_{UV}^{(t_1)}$ when (I, J) and (U, V) are structurally similar. In order to select node pairs with such a property, we consider an arbitrary pair (I, J) in the test set and the pair (U, V) such that $\Gamma(I, J, A^{(t_0)}; \mathbf{W}_\Gamma^\star); \mathbf{W}_\rho) \approx \Gamma(U, V, A^{(t_0)}; \mathbf{W}_\Gamma^\star); \mathbf{W}_\rho)$, where $\Gamma(a, b, A^{(t_0)}; \mathbf{W}_\Gamma^\star)$ is the trained Label GCN pairwise embedding space of pair $(a, b) \in V^{(t_0)} \times V^{(t_0)}$ on graph $A^{(t_0)}$. As an alternate hypothesis, we also consider the case where (U, V) is a node pair selected uniformly at random (from the test set as well). Note, however, that we only have a sample of $Y_{IJ}^{(t_1)}$ and $Y_{UV}^{(t_1)}$. Thus, to construct the probe outcome distribution of each node pair we consider the outcome of its 10 closest neighbors in the Label GCN pairwise embedding space. Since the two tested node pairs might share neighbors and directly induce the same distribution, we only consider the non-intersecting sets of 10 neighbors —if a node pair is a neighbor of both tested pairs, we assign it to the closest tested pair. To test if the distributions are the same ($Y_{IJ}^{(t_1)} \overset{d}{=} Y_{UV}^{(t_1)}$), we use Fisher's exact test [23] with a significance level of 0.05. Results are shown for all node pairs in the test set of both AE and LFM datasets from Section 6.3 in Figures 12 and 13. We can see how indeed structurally similar node pairs tend to have the same probe outcome distribution, in contrast to pairs selected at random, giving credence to the central assumption in our work that in some real-world tasks structural similarity before probing also implies similarity of probe outcomes.

G Datasets and Models

We conducted our experiments on four datasets: (1) Family tree, (2) Covariance matrix, (3) Amazon Electronics (AE) and (4) Last FM (LFM) datasets. These datasets allowed us to evaluate our findings in different settings, since they are diverse in terms of number of nodes and sparsity of their graphs. We compared (strictly) positional node embeddings, structural node embeddings and structural pairwise node embeddings, whose representative architectures are described in the main text. For the family tree dataset,

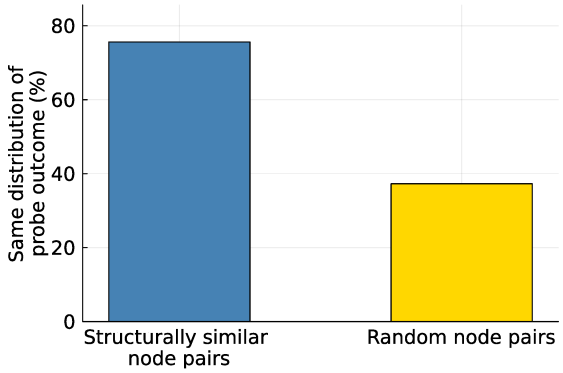

Figure 12: Results for the AE dataset. We show the percentage of pairs of node pairs that are structurally similar and have the same distribution vs. the ones that are paired uniformly at random.

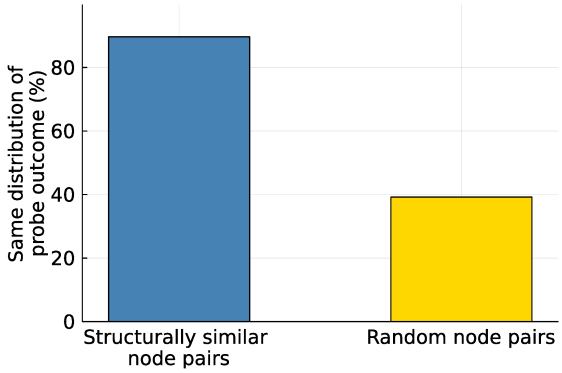

Figure 13: Results for the LFM dataset. We show the percentage of pairs of node pairs that are structurally similar and have the same distribution vs. the ones that are paired uniformly at random.

we also considered knowledge graph embeddings. All models are implemented in Pytorch [52] and Pytorch-Geometric [22]. Training was done on NVIDIA GeForce RTX 2080 Ti, GeForce GTX 1080 Ti, TITAN V, and TITAN Xp GPUs. Details on the datasets, models and hyperparameter grid can be found in what follows.

G.1 Datasets

Family tree. We generated the dataset by first creating family trees using the codebase and the ontology provided in [32], and then by combining them to ensure a percentage of the family trees contain two isomorphic subtrees. Following [32], each family tree is generated incrementally, starting from a single person, and it is grown by randomly selecting a person to whom a child or a parent is added. This process is repeated until the family tree reaches the maximum size of 26 nodes or the generation is randomly stopped, which happens with probability 0.002. Furthermore, each tree is constrained to a maximum depth of 5 and a maximum branching factor of 5, and the generated family trees are ensured to be non-isomorphic. After generating the family trees, we randomly selected 100 of the generated trees. In each of those, we randomly selected a leaf node and randomly chose from the previously not selected trees, two trees to be connected to the leaf or one tree to be connected twice, with probabilities 0.7 and 0.3 respectively. After this procedure, our dataset contains 100 family trees, with 30% of them containing two isomorphic subtrees. Finally, we use the ontology in [32] to compute all possible inferences for each tree in the dataset via symbolic reasoning. Only the `parentOf` relation is used as the observed graph, and the goal is to predict the other inferred relations (27 in total). We split the isomorphic subtrees between train and test such that relations within one subtree form the probes' (train) links, and the relations in the other subtree the counterfactual (test) links. In training nonedges are sampled uniformly at random from the training subtree. As for test, the nonedges are built taking one endpoint in the training subtree and another in the test subtree —to get nonedges that are node-wise isomormophic to edges in the test set.

Covariance matrix. We constructed the dataset making use of the data collected in [29], available in the UCI repository [15]. The data contains measurements values for 279 attributes from 452 patients. We considered only the patients that do not have missing values in any measurement, and we removed nominal and zero variance attributes. After this pre-processing step, our data comprises 68 patients and 182 attributes. We used a subset of 40 patients (selected at random) and computed the covariance matrix between the attributes, which is used as the observed adjacency matrix. Then, we recompute the covariance matrix with all the 68 patients and we split the attributes into two disjoint sets of size 75% and 25% of the total which are used as probes' (train) links and counterfactual (test) links respectively.

Amazon Electronics and Last FM. We considered the Amazon Electronics (AE) [74] and the Last FM (LFM) [49] datasets. The observed graph is obtained by considering user-item interactions occurring from

11/24/2015 to 12/24/2015 in AE and from 2007 until 2013 in LFM. In train and test we use interactions happening between 12/24/2015 and 12/31/2015 for AE and in 2014 for LFM. In both datasets we experiment with the subgroup of male users, while, at test time, our counterfactual queries are about female users. Note that, as mentioned in the main text, nonedges are obtained by sampling nodes uniformly at random.

G.2 Trained models

For all node embedding models we use as link function a Hadamard product followed by a multi-layer perceptron with one hidden layer, of the same size as the embedding, and ELU activations. In the following, we discuss details of embedding architectures and their hyperparameters.

G.2.1 Positional Node Embeddings

Below we discuss the architecture's choices for each dataset.

Family tree. We evaluated Nonnegative Matrix Factorization (NMF) [43] using the default implementation provided in Scikit-Learn [54]. For SVD [30] we make use of the low-rank implementation available in Pytorch [52]. Both models generated embeddings of size 256. We implemented positional GCN on top of a GCN architecture by using an additional embedding layer with the same size of the GNN layers. We used 3 GCN layers with dimension 256. Our models are trained with batches of sizes 32 (NMF) and 1024 (positional GCN) and an Adam optimizer with learning rate 0.005.

Covariance matrix. Same as *Family tree* for SVD, but with an embedding of size 64 and a learning rate of 0.01.

Amazon Electronics. Same as *Family tree* for NMF and SVD but with an embedding of size 8. Positional GCN uses 2 GNN layers with dimension 8. Our models are trained with batches of size 32 and an Adam optimizer with learning rate 0.005.

Last FM. Same as *Family tree* for NMF and SVD and positional GCN. Our models are trained with batches of size 32 and an Adam optimizer with learning rate 0.001.

G.2.2 Structural Node Embeddings

Below we discuss the architecture's choices for each dataset.

Family tree. We considered a GCN architecture with 3 layers with dimension 256. Our models are trained with batches of size 32 and an Adam optimizer with learning rate 0.005.

Covariance matrix. We considered a GCN architecture with 3 layers with dimension 64. We additionally make use of batch norm between each convolutional layer. Our models are trained with batches of size 16 and an Adam optimizer with learning rate 0.01.

Amazon Electronics. We considered a GCN architecture with 2 layers with dimension 8. Our models are trained with batches of size 32 and an Adam optimizer with learning rate 0.005.

Last FM. We considered a GCN architecture with 3 layers with dimension 256. Our models are trained with batches of size 32 and an Adam optimizer with learning rate 0.001.

G.2.3 Knowledge Graph Embeddings

For the *Family tree* dataset we used the Torch KGE [8] implementation of ComplEx, TransE and DistMult. Note that like SVD and NMF, the KGE embeddings are obtained through a pre-training procedure —the embeddings are then used to train a link function with Equation (13) as our task's objective. The pre-training

procedure uses Torch KGE implementation with a learning rate of 0.005, 0.00001 as the $L2$ regularization parameter and 2000 as the batch size.

G.2.4 Structural Pairwise Node Embeddings

Below we discuss the architecture's choices for each dataset.

Family tree. We use the same training hyperparameters as the GCN model, except for LabelGCN where we reduce the batch size to 8. For SEAL we use a dimension of 32 and k-hops subgraphs around the node pairs with $k = 3$ —other parameters follow the original work default implementation. For Neo-GNN, we used 3 GCN layers of dimension 32, learning rate of 0.005, batch size 32, paths of length 2, node dimension 128 and edge dimension 8 — all other parameters follow the original work default implementation.

Covariance matrix. We use LabelGCN as representative of structural pairwise embeddings. We use 3 GCN layers of size 64 with batch norm. We train with batches of size 16 and an Adam optimizer with learning rate 0.01.

Amazon Electronics. SEAL uses k-hops subgraphs around the node pairs with $k = 1$ and a model composed of 2 GCN layers with dimension 32. The model is trained with batches of size 32 and an Adam optimizer with learning rate 0.005 and weight decay of 0.00005. For LabelGCN we instead use a model with 2 GCN layers with dimension 8. The model is trained with batches of size 8 and an Adam optimizer with learning rate 0.005. As for Neo-GNN, we used an $L2$ regularization of 0.00005, a learning rate of 0.005, batch size 32, 2 GNN layers with dimension 32, paths of length 1, node dimension 128 and edge dimension 8 — all other parameters follow the original work verbatim.

Last FM. For LabelGCN we use a model with 3 GCN layers with dimension 256. The model is trained with batches of size 10 and an Adam optimizer with learning rate 0.001. For SEAL we use a dimension of 16, 3 GCN layers, and k-hops subgraphs around the node pairs with $k = 1$ —other parameters follow the original work default implementation. For Neo-GNN, we used 2 GCN layers with dimension 16, a learning rate of 0.001, batch size 10, paths of length 1, node dimension 128 and edge dimension 8 — all other parameters follow the original work default implementation.

H Limitations and Possible Extensions

Finally, we would like to highlight two limitations of our work that could be explored in future research. First, our family of SCMs is designed to model systems with causal link prediction tasks in mind, that is, they are universal models with respect to the observational distribution of the graph, designed to answer causal link prediction queries. For tasks focused on intervening, for instance, in higher-order structures, the query cannot be trivially formulated through our SCMs. Possible extensions could include higher-order generalizations of Definition 12. Finally, we also highlight how Assumption 1, a central assumption in our work, is not satisfied in some specific scenarios. In particular, in very large, highly dynamical, graphs nodes may significantly evolve in the time between a probe and its observed effect. In these scenarios, Assumption 1 may be violated and further work is needed in order to address these cases.