
Compositional Fairness Constraints for Graph Embeddings

Avishek Joey Bose^{1 2} William L. Hamilton^{1 2 3}

Abstract

Learning high-quality node embeddings is a key building block for machine learning models that operate on graph data, such as social networks and recommender systems. However, existing graph embedding techniques are unable to cope with fairness constraints, e.g., ensuring that the learned representations do not correlate with certain attributes, such as age or gender. Here, we introduce an adversarial framework to enforce fairness constraints on graph embeddings. Our approach is *compositional*—meaning that it can flexibly accommodate different combinations of fairness constraints during inference. For instance, in the context of social recommendations, our framework would allow one user to request that their recommendations are invariant to both their age and gender, while also allowing another user to request invariance to just their age. Experiments on standard knowledge graph and recommender system benchmarks highlight the utility of our proposed framework.

1. Introduction

Learning low-dimensional embeddings of the nodes in a graph is a fundamental technique underlying state-of-the-art approaches to link prediction and recommender systems (Hamilton et al., 2017b). However, in many applications—especially those involving social graphs—it is desirable to exercise control over the information contained within learned node embeddings. For instance, we may want to ensure that recommendations are fair or balanced with respect to certain attributes (e.g., that they do not depend on a user’s race or gender) or we may want to ensure privacy by not exposing certain attributes through learned node representations. In this work we investigate the feasibility of enforcing such *invariance constraints* on (social) graph embeddings.

¹McGill University ²Mila ³Facebook AI Research. Correspondence to: Avishek Joey Bose <joey.bose@mail.mcgill.ca>.

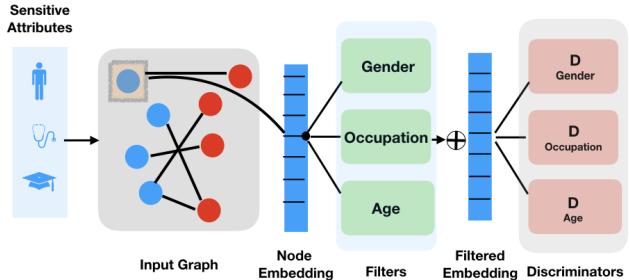


Figure 1. Overview of our approach: Our goal is to generate graph embeddings that are invariant to particular sensitive attributes (e.g., age or gender). We train a set of “filters” to prevent adversarial discriminators from classifying the sensitive information from the filtered embeddings. After training, these filters can be composed together in different combinations, allowing the flexible generation of embeddings that are invariant w.r.t. any subset of the sensitive attributes.

While enforcing invariance constraints on general classification models (Chouldechova, 2017; Gajane & Pechenizkiy, 2017; Kamishima et al., 2012) and collaborative filtering algorithms (Yao & Huang, 2017) has received considerable attention in recent years, these techniques have yet to be considered within the context of graph embeddings—a setting that introduces particular challenges due to the non-i.i.d. and non-Euclidean nature of relational, graph data.

Moreover, in the case of social graphs and large-scale recommender systems, it is often the case that there are many possible sensitive attributes that we *may* want to enforce invariance constraints over. Previous work on enforcing invariance (or “fairness”) in social applications has generally focused on situations that involve one sensitive attribute (e.g., age in the context of credit or loan decisions; Zemel et al. (2013)), but in the context of social graph embeddings there can be an extremely large number of possible sensitive attributes. In fact, in extreme settings we may even want to be fair with respect to the existence of individual edges. For instance, a user on a social networking platform might want that platform’s recommender system to ignore the fact that they are friends with a certain other user, or that they engaged with a particular piece of content.

Our contributions. We introduce an adversarial framework to enforce *compositional* fairness constraints on graph embeddings for multiple sensitive attributes. The key idea be-

hind our approach is that we learn a set of *adversarial filters* that remove information about particular sensitive attributes. Crucially, each of these learned filters can be *optionally* applied after training, so the model can flexibly generate embeddings that are invariant with respect to different combinations of sensitive attributes. As the space of possible combinations of sensitive attributes can be combinatorially large, we demonstrate that our compositional strategy can generate invariant embeddings even on unseen combinations at test time. Our contribution is at the intersection of research on (social) graph embedding and algorithmic fairness. We build upon the success of recent adversarial approaches to fairness (Edwards & Storkey, 2015), disentanglement (Mathieu et al., 2016), and transfer learning (Madras et al., 2018)—extending these approaches to the domain of graph representation learning and introducing new algorithmic techniques to accommodate compositional constraints during inference.

2. Related Work

We now briefly highlight core related work on (social) graph embeddings and algorithmic fairness, which our research builds upon.

2.1. Graph Embedding

At the core of our proposed methodology is the notion of learning low-dimensional embeddings of graph-structured data, especially social data. Graph embedding techniques have a long history in the social sciences, with connections to early research on “sociograms” (small hand-constructed social networks) and latent variable models of social interactions (Faust, 1988; Majone, 1972). In more recent years, the task of embedding graph-structured data has received increasing attention from the machine learning and data mining communities (Cai et al., 2018; Hamilton et al., 2017b). Generally, the goal of these works is to map graph nodes to low-dimensional vector embeddings, such that the original graph can be reconstructed from these embeddings. Traditional approaches to this problem include Laplacian eigenmaps (Belkin & Niyogi, 2002) and matrix factorization techniques (Ng et al., 2001), with recent years witnessing a surge in methods that rely on random-walk based objectives (Grover & Leskovec, 2016; Perozzi et al., 2014), deep autoencoders (Wang et al., 2016), and graph neural networks (Hamilton et al., 2017a; Kipf & Welling, 2016).

Learned graph embeddings can be used for a wide variety of tasks, including node classification, relation prediction, and clustering (Hamilton et al., 2017b). Here, we focus on the relation prediction task, i.e., using the learned representations to predict previously unobserved relationships between the input nodes. The relation prediction task is exceptionally general—for example, it generalizes basic rec-

ommender systems, knowledge base completion, and even node classification (*ibid.*).

2.2. Algorithmic Fairness

Unlike previous research on graph embedding, in this work we focus on the challenge of enforcing fairness or invariance constraints on the learned representations. Recent work on fairness in machine learning, including work on fairness in collaborative filtering, involves making predictions that are balanced or invariant with respect to certain sensitive variables (e.g., age or gender) (Chouldechova, 2017; Gajane & Pechenizkiy, 2017; Kamishima et al., 2012; Madras et al., 2018; Zemel et al., 2013; Yao & Huang, 2017). Formally, in the standard “fair classification” setting we consider a data point $\mathbf{x} \in \mathbb{R}^n$, its class label $y \in \mathcal{Y}$, and a binary sensitive attribute $a \in \{0, 1\}$ (e.g., indicating gender). The high-level goal is then to train a model to predict y from \mathbf{x} , while making this prediction invariant or fair with respect to a (Madras et al., 2018). There are many specific definitions of fairness, such as whether fairness refers to parity or satisfying certain preferences (see (Gajane & Pechenizkiy, 2017) for a detailed discussion). In the context of fair machine learning, our core contribution is motivating, implementing, and evaluating an approach to enforce fairness within the context of graph embeddings. There are a number of complications introduced by this setting—for instance, rather than one classification task, we instead have thousands or even millions of interdependent edge relationships.

3. Preliminaries

We consider the general case of embedding a heterogeneous or multi-relational (social) graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which consists of a set of directed edge triples $e = \langle u, r, v \rangle \in \mathcal{E}$, where $u, v \in \mathcal{V}$ are nodes and $r \in \mathcal{R}$ is a relation type. We further assume that each node is of a particular type, $\mathcal{T} \subseteq \mathcal{V}$, and that relations may have constraints regarding the types of nodes that they can connect.

Relation Prediction. The general *relation prediction* task on such a graph is as follows. Let $\mathcal{E}_{\text{train}} \subset \mathcal{E}$ denote a set of observed *training* edges and let $\bar{\mathcal{E}} = \{\langle v_i, r, v_j \rangle : v_i, v_j \in \mathcal{V}, r \in \mathcal{R}\} \setminus \mathcal{E}$ denote the set of *negative* edges that are not present in the true graph \mathcal{G} . Given $\mathcal{E}_{\text{train}}$, we aim to learn a scoring function s such that

$$s(e) > s(e'), \forall e \in \mathcal{E}, e' \in \bar{\mathcal{E}}. \quad (1)$$

In other words, the learned scoring function should ideally score any true edge higher than any negative edge.

Embedding-based Models. In the context of graph embeddings, we aim to solve this relation prediction task by learning a function $\text{ENC} : \mathcal{V} \mapsto \mathbb{R}^d$ that maps each node $v \in \mathcal{V}$ to an embedding $\mathbf{z}_v = \text{ENC}(v)$. In this case, the signature

of the score function becomes $s : \mathbb{R}^d \times \mathcal{R} \times \mathbb{R}^d \mapsto \mathbb{R}$, i.e., it takes two node embeddings $\mathbf{z}_u, \mathbf{z}_v \in \mathbb{R}^d$ and a relation $r \in \mathcal{R}$ and scores the likelihood that the edge $e = \langle u, r, v \rangle$ exists in the graph. Generally, the intuition in embedding-based approaches is that the distance between two node embeddings should encode the likelihood that there is an edge between the nodes. Following standard practice, we consider the optimization of these scoring functions using contrastive learning methods that make use of a corruption distribution such as *noise contrastive estimation* (Dyer, 2014; Mnih & Teh, 2012) and similar variants (Bose et al., 2018), where the loss over a batch of edges $\mathcal{E}_{\text{batch}} \subseteq \mathcal{E}_{\text{train}}$ is given by:

$$\sum_{e \in \mathcal{E}_{\text{edge}}} L_{\text{edge}}(s(e), s(e_1^-), \dots, s(e_m^-)), \quad (2)$$

where L_{edge} is a per-edge loss function and $e_1^-, \dots, e_m^- \in \bar{\mathcal{E}}$ are “negative samples”, i.e., randomly sampled edges that do not exist in the graph. Loss functions of this form generally attempt to maximize the likelihood of true edges compared to the negative samples.

Fairness. In order to incorporate the notion of fairness into the graph embedding setup, we assume that for exactly one node type \mathcal{T}^* , all nodes of this type, i.e. all $u \in \mathcal{T}^*$, have K categorical sensitive attributes, $a_u^k \in \mathcal{A}_k, k = 1, \dots, K$, and for simplicity, we assume that there are no other features or attributes associated with the nodes and edges in the graph.¹ The challenge in enforcing fairness is thus to ensure that the learned node embeddings, \mathbf{z}_u , are not biased or unfair with respect to these sensitive attributes—a point which we formalize in the next section.

4. Invariant Graph Embeddings

We first motivate and argue in favor of a particular form of “fairness” (or rather *invariance*) within the context of graph embeddings. Following this, we outline our compositional and adversarial approach for enforcing these invariance constraints on graph embeddings.

4.1. Pragmatic Fairness as Invariance

In this paper we consider a simple, user-centric formulation of fairness within the context of social graph embeddings. Using gender as an example of a sensitive attribute and movie recommendation as an example relation prediction task, our approach is guided by the following question: *If one gives a user a button that says “Please ignore my gender when recommending movies”, what does a user expect from the system after this button is pressed?* Here, we accept it as non-controversial that the expectation from the user is that recommendation does not depend in any way on their

¹Though this assumption can easily be relaxed.

gender, i.e., that the recommendation would be the same regardless of their gender. Formally, given a user u , this expectation amounts to an assumption of independence,

$$s(e) \perp a_u \quad \forall v \in \mathcal{V}, r \in \mathcal{R} \quad (3)$$

between the recommendation—i.e., the score of the edge, $s(e) = s(\langle \mathbf{z}_u, r, \mathbf{z}_v \rangle)$ —and the sensitive attribute a_u .

One issue in directly enforcing Equation (3) is that there are many (potentially millions) of possible edges that we might want to score for every node $u \in \mathcal{T}^*$, making it intractable to enforce independence on each of these decisions individually. However, if we assume that the score function $s(\langle \mathbf{z}_u, r, \mathbf{z}_v \rangle)$ depends on u only through u ’s embedding, \mathbf{z}_u , then we can guarantee the independence in Equation (3) for all edge predictions by enforcing what we call *representational invariance*:

$$\mathbf{z}_u \perp a_u, \quad \forall u \in \mathcal{V}. \quad (4)$$

In other words, we require that the mutual information $I(\mathbf{z}_u, a_u)$ is 0.

Generalizing to the setting of multiple sensitive attributes, for a given set of sensitive attributes $S \subseteq \{1, \dots, K\}$, we would require that

$$I(\mathbf{z}_u, a_u^k) = 0, k \in S, \forall u \in \mathcal{V}, \quad (5)$$

which amounts to the assumption of S independent invariance constraints on the S distinct sensitive attributes.² Importantly, we assume that the set S is not fixed (e.g., different users might request different invariance constraints). In the language of algorithmic fairness, the representational invariance of Equation (5) implies that traditional demographic parity constraints are satisfied on the sensitive attributes and recommendations.

4.2. Model Definition

In this work, we enforce representational invariance constraints on the node embeddings (Equation 5) by introducing an adversarial loss and a technique to “filter” the embeddings generated by the ENC function. Note, again, that a unique challenge here is that S —the set of sensitive attributes we want to be invariant with respect to—is not fixed across nodes; i.e., we may want to enforce invariance on different sets of sensitive attributes for different nodes.

Note also that the framework presented in this section is quite general and can function with arbitrary combinations of base node embedding functions ENC and edge-prediction losses L_{edge} (see Equation 2). We discuss three concrete instantiations of this framework in Section 5.

²Note that this does not necessarily imply “subgroup fairness” on the joint distribution (Kearns et al., 2017).

Compositional Encoder. The first key component of our model is generalizing the ENC embedding function to optionally “filter” out the information about certain sensitive attributes. In particular, for every sensitive attribute $k \in \{1, \dots, K\}$ we define a filter function $f_k : \mathcal{R}^d \mapsto \mathcal{R}^d$ that is trained to remove the information about the k th sensitive attribute. If we want the node embedding to be invariant w.r.t. some set of sensitive attributes $S \subseteq \{1, \dots, K\}$, we then generate its embedding by “composing” the output of the $|S|$ filtered embeddings using a *compositional encoder*:

$$\text{C-ENC}(u, S) = \frac{1}{|S|} \sum_{k \in S} f_k(\text{ENC}(u)) \quad (6)$$

To train C-ENC(u, S) we sample a binary mask to determine the set S at every iteration. In this work, we sample the binary mask as a sequence of k independent Bernoulli draws with a common fixed probability $p = 0.5$; however, other application-specific distributions (e.g., incorporating dependencies between the attributes) could be employed. Sampling random binary masks forces the model to produce invariant embeddings for different combinations of sensitive attributes during training with the hope of generalizing to unseen combinations during inference time — a phenomena that we empirically validate in Section 5.2.

Adversarial Loss. To train the compositional encoder, we employ an adversarial regularizer. For each sensitive attribute $k \in K$, we define a discriminator $D_k : \mathbb{R}^d \times \mathcal{A}_k \mapsto [0, 1]$, which attempts to predict the k th sensitive attribute from the node embeddings. Assuming we are given an edge-prediction loss function L_{edge} (as in Equation 2), we can then define our new adversarially regularized per-edge loss as

$$L(e) = L_{\text{edge}}(s(e), s(e_1^-), \dots, s(e_m^-)) + \lambda \sum_{k \in S} \sum_{a^k \in \mathcal{A}_k} \log(D_k(\text{C-ENC}(u, S), a^k)), \quad (7)$$

where λ is a hyperparameter controlling the strength of the adversarial regularization. To optimize this loss in a mini-batch setting, we alternate between two types of stochastic gradient descent updates: (1) T minibatch updates minimizing $L(e)$ with respect to C-ENC (with all the D_k fixed), and (2) T' minibatch updates minimizing $-L(e)$ with respect to D_k , $k = 1, \dots, K$ (with C-ENC fixed).

Theoretical Considerations. For clarity and simplicity, we consider the case of a single binary sensitive attribute, with the theoretical intuitions naturally generalizing to the multi-attribute and multi-class settings. Assuming a single binary sensitive attribute a_k , by simple application of Proposition 2 in Goodfellow et al. (2014), we have:³

³Theorem 1 holds as a consequence of Proposition 2 in Goodfellow et al. (2014) if we simply replace the task of distinguishing real/fake data by classifying a binary sensitive attribute.

Theorem 1. *If c-enc and D_k have enough capacity, T' is large enough so that D_k is allowed to reach its optimum on $-L(e)$ (with c-enc fixed), and c-enc is optimized according to $L(e)$ (with D fixed), then $I(\mathbf{z}_u, a_u) \rightarrow 0, \forall u \in \mathcal{T}^*$ as $\lambda \rightarrow \infty$.*

That is, if we increase the weight of the adversarial regularizer to infinity, the equilibrium of the minimax game in Equation (7) occurs when there is zero mutual information between the sensitive attribute and the embeddings. Of course, as $\lambda \rightarrow \infty$ trivial solutions to this game exist (e.g., C-ENC simply outputting a constant value) and in practice setting $\lambda < \infty$ leads to a tradeoff between performance on edge prediction and representational invariance.

5. Experiments

We investigated the impact of enforcing invariance on graph embeddings using three datasets: Freebase15k-237⁴, MovieLens-1M⁵, and an edge-prediction dataset derived from Reddit.⁶ The dataset statistics are given in Table 1. Our experimental setup closely mirrors that of (Madras et al., 2018) where we jointly train the main model with adversaries, but when testing invariance, we train a *new* classifier (with the same capacity as the discriminator) to predict the sensitive attributes from the learned embeddings.

The goal of our experiments was to answer three questions:

- (Q1) **The cost of invariance.** What is the tradeoff between enforcing invariance on the representations and accuracy on the main edge prediction task?
- (Q2) **The impact of compositionality.** How does the performance of a compositional approach, which jointly enforces fairness over a set of sensitive attributes, compare to a more traditional model that only enforces fairness on a single attribute?
- (Q3) **Invariance on unseen combinations.** In settings with many sensitive attributes, is our approach able to enforce invariance even on combinations of sensitive attributes that it never saw during training?

Throughout these experiments, we rely on two baselines: First, we compare against baselines that do not include any invariance constraints, i.e., models with $\lambda = 0$. Second, we compare against a non-compositional adversarial approach where we separately train K distinct encoders and K distinct adversaries for each of the K sensitive attributes in the data. This non-compositional adversary is essentially an extension of Edwards & Storkey (2015)’s approach to the graph embedding domain.

⁴www.microsoft.com/en-us/download/details.aspx?id=52312

⁵grouplens.org/datasets/movielens/1m/

⁶reddit.com

5.1. Setup and Datasets

Before describing our experimental results, we first outline the key properties of the datasets we used, as well as the specific encoders and edge-prediction loss functions used.

In all experiments, we used multi-layer perceptrons (MLPs) with leaky ReLU activation functions (Xu et al., 2015) as the discriminators D_k and filters f_k . The Appendix contains details on the exact hyperparameters (e.g., number of layers and sizes) used for all the different experiments, as well as details on the training procedures (e.g., number of epochs and data splits). Code to reproduce our results is available at: <https://github.com/joeybose/Flexible-Fairness-Constraints>.

FREEBASE15K-237

Freebase 15k-237 is a standard benchmark used for knowledge base completion (Toutanova et al., 2015). In this work, we use Freebase 15k-237 as a semi-synthetic testbed to evaluate the impact of adversarial regularization. Taking the entity attribute labels from Moon et al. (2017), we used the 3-most common attribute labels (e.g., /award/award_nominee) as “sensitive” attributes. The goal in this dataset is to perform the standard knowledge base completion task, while having the entity embeddings be invariant with respect to these “sensitive” attribute labels. While synthetic, this dataset provides a useful reference point due to its popularity in the graph embedding literature.

For our encoder and edge-prediction loss function, we follow Ji et al. (2015)’s TransD approach, since we found this approach gave significant performance boosts compared to simpler models (e.g., TransE). In this model, the encoding of a node/entity depends on the edge relation being predicted, as well as on whether the entity is the head or tail in a relation (i.e., the edge direction matters). In particular, the embedding of the head node (i.e., the source node) in an edge relation is given by:

$$\text{ENC}(u, \langle u, r, v \rangle) = (\mathbf{r}_p \mathbf{u}_p^\top + \mathbf{I}^{d \times d}) \mathbf{u}, \quad (8)$$

where $\mathbf{u}, \mathbf{u}_p, \mathbf{r}_p \in \mathbb{R}^d$ are trainable embedding parameters and $\mathbf{I}^{d \times d}$ is a d -dimensional identity matrix. The encoding function for the tail node is defined analogously. The score function for this approach is given by

$$s(\langle u, r, v \rangle) = -\|\text{ENC}(u, \langle u, r, v \rangle) + \mathbf{r} - \text{ENC}(v, \langle u, r, v \rangle)\|_2,$$

where $\mathbf{r} \in \mathbb{R}^d$ is another trainable embedding parameter (one per relation). Finally, we use a standard max-margin loss with a single negative sample per positive edge:

$$L_{\text{edge}}(s(e), s(e^-)) = \max(0, 1 - s(e) + s(e^-)). \quad (9)$$

MOVIELENS-1M

Our second dataset is derived from the MovieLens-1M recommender system benchmark (Harper & Konstan, 2016). This is a standard recommender system benchmark, where the goal is to predict the rating that users assign movies. However, unlike previous work, in our experiments we treat the user features (age, gender, and occupation) as sensitive attributes (rather than as additional feature information for the recommendation task). Following Berg et al. (2017) we treat this recommendation task as an edge prediction problem between users and movies, viewing the different possible ratings as different edge relations.

For this dataset we use a simple “embedding-lookup” encoder, where each user and movie is associated with a unique embedding vector in \mathbb{R}^d . As a scoring function, we follow Berg et al. (2017) and use a log-likelihood approach:

$$s(\langle u, r, m \rangle) = \mathbf{z}_u^\top \mathbf{Q}_r \mathbf{z}_v - \log\left(\sum_{r' \in \mathcal{R}} \mathbf{z}_u^\top \mathbf{Q}_{r'} \mathbf{z}_v\right),$$

The relation matrices $\mathbf{Q}_r \in \mathbb{R}^{d \times d}$ are computed as:

$$\mathbf{Q}_r = a_{r,1} \mathbf{P}_1 + a_{r,2} \mathbf{P}_2,$$

where $a_{r,1}, a_{r,2} \in \mathcal{R}$ and $\mathbf{P}_1, \mathbf{P}_2 \in \mathcal{R}^{d \times d}$ are trainable parameters. In this case, the loss function is simply the negative of the log-likelihood score.

REDDIT

The final dataset we consider is based on the social media website Reddit—a popular, discussion-based website where users can post and comment on content in different topical communities, called “subreddits”. For this dataset, we consider a traditional edge prediction task, where the goal is to predict interactions between users and subreddit communities.

To construct the edge prediction task, we examined all comments from the month of November in 2017, and we placed an edge between a user and a community if this user commented on that community at least once within this time period. We then took the 10-core of this graph to remove low-degree nodes, which resulted in a graph with approximately 366K users, 18K communities, and 7M edges. Given this graph, the main task is to train an edge-prediction model on 90% of the user-subreddit edges and then predict missing edges in a held-out test set of the remaining edges.

Reddit is a pseudonymous website with no public user attributes. Thus, to define sensitive attributes, we treat certain subreddit nodes as *sensitive nodes*, and the sensitive attributes for users are whether or not they have an edge connecting to these sensitive nodes. In other words, the fairness objective in this setting is to force the model to be invariant to whether or not a user commented on a particular community. To select the “sensitive” subreddit communities,

Table 1. Statistics for the three datasets, including the total number of nodes ($|\mathcal{V}|$) and number of nodes with sensitive attributes $|\mathcal{T}^*|$, the number of sensitive attributes and their types and the total number of edges in the graph.

DATASET	$ \mathcal{V} $	$ \mathcal{T}^* $	#SENSITIVE ATTRIBUTES	EDGES	BINARY ATTRIBUTES?	MULTICLASS ATTRIBUTES?
FB 15K-237	14,940	14,940	3	168,618	✓	✗
MOVIELENS1M	9,940	6,040	3	1,000,209	✓	✓
REDDIT COMMENTS	385,735	366,797	10	7,255,096	✓	✗

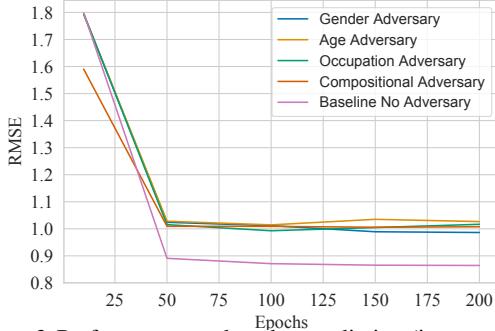


Figure 2. Performance on the edge prediction (i.e., recommendation) task on MovieLens, using RMSE as in Berg et al. (2017).

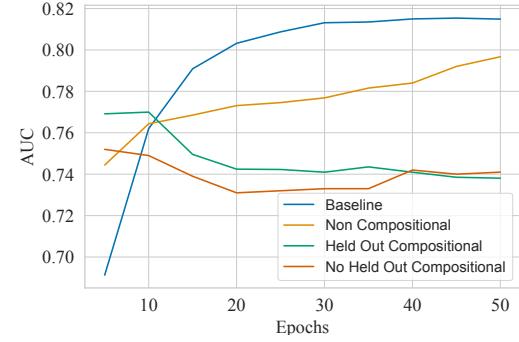


Figure 3. Performance on the edge prediction (i.e., recommendation) task on the Reddit data. Evaluation is using the AUC score, since there is only one edge/relationship type.

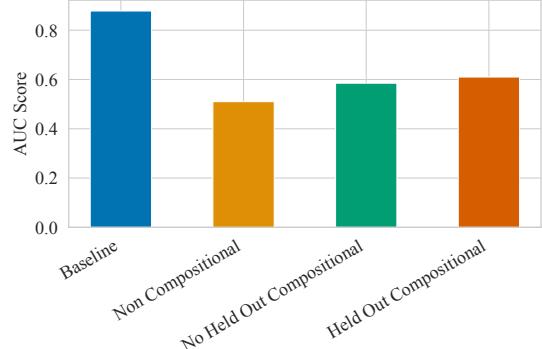


Figure 4. Ability to predict sensitive attributes on the Reddit data when using various embedding approaches. Bar plots correspond to the average AUC across the 10 binary sensitive attributes.

we randomly sampled 10 from the top-100 communities by degree.⁷ Note that this setting represents the extreme case where we want the model to be invariant with respect to the existence of particular edges in the input graph.

As with MovieLens-1M, we use a simple “embedding-lookup” encoder. In this case, there is only a single relation type—indicating whether a Reddit user has commented on a “ subreddit” community. Thus, we employ a simple dot-product based scoring function, $s(\langle u, r, v \rangle) = \mathbf{z}_u^\top \mathbf{z}_v$, and we use a max-margin loss as in Equation (9).

5.2. Results

We now address the core experimental questions (Q1-Q3).

Q1: THE COST OF INVARIANCE

In order to quantify the extent to which the learned embeddings are invariant to the sensitive attributes (e.g., after adversarial training), we freeze the trained compositional encoder C-ENC and train a new MLP classifier to predict each sensitive attribute from the filtered embeddings (i.e., we train one new classifier per sensitive attribute). We also evaluate the performance of these filtered embeddings on the original prediction tasks. In the best case, a newly trained MLP classifier should have random accuracy when attempting to predict the sensitive attributes from the filtered embeddings, but these embeddings should still provide strong

⁷We excluded the top-5 highest-degree outlying communities.

performance on the main edge prediction task. Thus, for binary sensitive attributes, an ideal result is an AUC score of 0.5 when attempting to predict the sensitive attributes from the learned embeddings.

Overall, we found that on the more realistic social recommendation datasets—i.e., the MovieLens-1M and Reddit datasets—our approach was able to achieve a reasonable tradeoff, with the near-complete removal of the sensitive information leading to a roughly 10% relative error increase on the edge prediction tasks. In other words, on these two datasets the sensitive attributes were nearly impossible to predict from the filtered embeddings, while the accuracy on the main edge prediction task was roughly 10% worse than

Table 2. Ability to predict sensitive attributes on the MovieLens data when using various embedding approaches. For gender attribute the score is AUC while for age and occupation attributes the score is micro averaged F1. The columns represent the different embedding approaches (e.g., with or without adversarial regularizatin) while the rows are the attribute being classified.

MOVIELENS1M	BASELINE NO AD- VERSARY	GENDER ADVERSARY	AGE ADVERSARY	OCCUPATION ADVERSARY	COMP. ADVERSARY	MAJORITY CLASSIFIER	RANDOM CLASSIFIER
GENDER	0.712	0.532	0.541	0.551	0.511	0.5	0.5
AGE	0.412	0.341	0.333	0.321	0.313	0.367	0.141
OCCUPATION	0.146	0.141	0.108	0.131	0.121	0.126	0.05

Table 3. Ability to predict sensitive attributes on the Freebase15k-237 data when using various embedding approaches. AUC scores are reported, since all the sensitive attributes are binary. The mean rank on the main edge-prediction task is also reported.

FB15K-237	BASELINE NO AD- VERSARY	NON COMP. AD- VERSARY	COMP. ADVERSARY
ATTRIBUTE 0	0.97	0.82	0.77
ATTRIBUTE 1	0.99	0.81	0.79
ATTRIBUTE 2	0.98	0.81	0.81
MEAN RANK	285	320	542

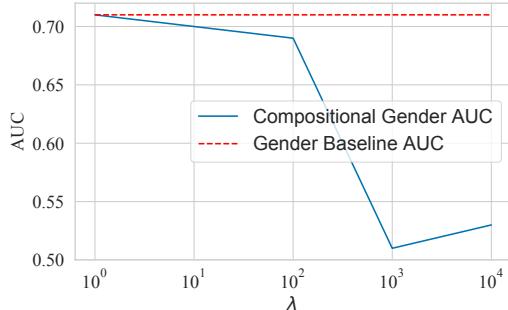


Figure 5. Tradeoff of Gender AUC score for a compositional adversary versus different λ

a baseline approach that does not include the invariance constraints. Table 2 and Figure 2 summarize these results for the MovieLens data, where we can see that the accuracy of classifying the sensitive attributes is on-par with a majority-vote classifier (Table 2) while the RMSE degrades from 0.865 to 1.01 with the compositional adversary. Figures 5 and 6 illustrate this tradeoff and show how the RMSE for the edge prediction task and ability to predict the sensitive attributes change as we vary the regularization strength, λ . As expected, increasing λ does indeed produce more invariant embeddings but at the cost of higher RMSE values. Figures 3 and 4 similiarily summarize these results on Reddit.

Interestingly, we found that on the Freebase15k-237 dataset it was not possible to completely remove the sensitive infor-

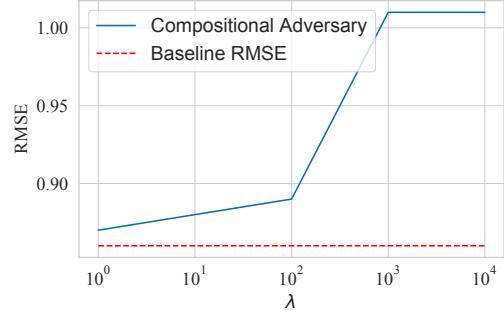


Figure 6. Tradeoff of RMSE for a compositional adversary versus different λ

mation without incurring a significant cost on the original edge prediction task. This result is not entirely surprising, since for this dataset the “sensitive” attributes were synthetically constructed from entity type annotations, which are presumably very relevant to the main edge/relation prediction task. However, it is an interesting point of reference that demonstrates the potential limitations of removing sensitive information from learned graph embeddings.

Q2: THE IMPACT OF COMPOSITIONALITY

In all our experiments, we observed that our compositional approach performed favorably compared to an approach that individually enforced fairness on each individual attribute. In fact, on the MovieLens-1M data (and the synthetic Freebase15k-237 data), the compositionally trained adversary *outperformed* the individually trained adversaries in terms of removing information about the sensitive attributes (Table 2). In other words, training a model to jointly remove information about the sensitive attributes using the compositional encoder (Equation 6) removed more information about the sensitive attributes than training separate adversarially regularized embedding models for each sensitive attribute. This result is not entirely surprising, as it essentially indicates that the different sensitive attributes (age, gender, and occupation) are correlated in this dataset. Nonetheless, it is a positive result indicating that the extra flexibility afforded by the compositional approach does

not necessarily lead to a decrease in performance. That said, on the Reddit data we observed the opposite trend and found that the compositional approach performed worse in terms of its ability to remove information about the sensitive attributes (Figure 4) as well as a small drop on the performance of the main edge prediction task (Figure 3).

Q3: INVARIANCE ON UNSEEN COMBINATIONS

One of the key benefits of the compositional encoder is that it can flexibly generate embeddings that are invariant to any subset of $S \subseteq \{1, \dots, K\}$ of the sensitive attributes in a domain. In other words, at inference time, it is possible to generate 2^K distinct embeddings for an individual node, depending on the exact set of invariance constraints. However, given this combinatorially large output space, a natural question is whether this approach performs well when generalizing to unseen combinations of sensitive attributes.

We tested this phenomenon on the Reddit dataset, since it has the largest number of sensitive attributes (10, compared to 3 sensitive attributes for the other two datasets). During training we held out 10% of the combinations of sensitive attributes, and we then evaluated the model’s ability to enforce invariance on this held-out set. As we can see in Figure 4, the performance drop for the held-out combinations is very small (0.025), indicating that our compositional approach is capable of effectively generalizing to unseen combinations. The Appendix contains further results demonstrating how this trends scales gracefully when we increase the number of sensitive attributes from 10 to 50.

QUANTIFYING BIAS

In all of the above results, we used the ability to classify the sensitive attributes as a proxy for bias being contained within the embeddings. While this is a standard approach, e.g., see Edwards & Storkey (2015), and an intuitive method for evaluating representational invariance—a natural question is whether the adversarial regularization also decreases bias in the edge prediction tasks. Ideally, after filtering the embeddings, we would have that the edge predictions themselves are not biased according to the sensitive attributes.

To quantify this issue, we computed a “prediction bias” score for the MovieLens1M dataset: For each movie, we computed the absolute difference between the average rating predicted for each possible value of a sensitive attribute and we then averaged these scores over all movies. Thus, for example, the bias score for gender corresponds to the average absolute difference in predicted ratings for male vs. female users, across all movies. From the perspective of fairness our adversary imposes a soft demographic parity constraint on the main task. A reduction in prediction bias across the different subgroups represents an empirical measure of achieving demographic parity. Figure 7 highlights these

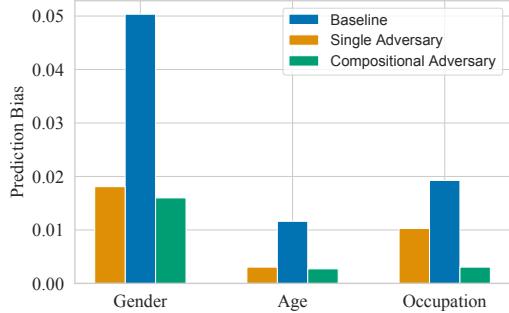


Figure 7. Prediction Bias for different Sensitive Attributes under three settings in MovieLens1M.

results, which show that adversarial regularization does indeed drastically reduce prediction bias. Interestingly, using a compositional adversary works better than a single adversary for a specific sensitive attribute which we hypothesize is due to correlation between sensitive attributes.

6. Discussion and Conclusion

Our work sheds light on how fairness can be enforced in graph representation learning—a setting that is highly relevant to large-scale social recommendation and networking platforms. We found that using our proposed compositional adversary allows us to flexibly accommodate unseen combinations of fairness constraints without explicitly training on them. Crucially, this highlights how fairness could be deployed in a real-word, user-driven setting, where it is necessary to optionally enforce a large number of possible invariance constraints over learned graph representations.

In terms of limitations and directions for future work, one important limitation is that we only consider one type of adversarial loss to enforce fairness. While this adversarial loss is theoretically motivated and known to perform well, there are other recent variations in the literature (e.g., Madras et al. (2018)—as well as related non-adversarial regularizers (e.g., Zemel et al. (2013)). Also, while we considered imposing fairness over sets of attributes, we did not explicitly model subgroup-level fairness (Kearns et al., 2017). Extending and testing our framework with these alternatives is a natural direction for future work.

There are also important questions about how our framework translates to real-world production systems. For instance, in this work we enforced fairness with respect to randomly sampled sets of attributes, but in real-world environments, these sets of attributes would come from user preferences, which may themselves be biased; e.g., it might be more common for female users to request fairness than male users potentially leading to new kinds of demographic inequalities. Understanding how these preference biases could impact our framework is an important direction for future inquiry.

Acknowledgements

The authors would like to thank the anonymous ICML reviewers for their helpful comments. In addition we would like to thank Koustuv Sinha, Riashat Islam and Andre Cianflone for helpful feedback on earlier drafts of this work. This research was funded in part by an academic grant from Microsoft Research, as well as a Canada CIFAR Chair in AI, held by Prof. Hamilton.

References

- Belkin, M. and Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, 2002.
- Berg, R. v. d., Kipf, T. N., and Welling, M. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*, 2017.
- Bose, A. J., Ling, H., and Cao, Y. Adversarial contrastive estimation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, 2018. URL <https://arxiv.org/abs/1805.03642>.
- Cai, H., Zheng, V. W., and Chang, K. A comprehensive survey of graph embedding: problems, techniques and applications. *IEEE Transactions on Knowledge and Data Engineering*, 2018.
- Chouldechova, A. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017.
- Dyer, C. Notes on noise contrastive estimation and negative sampling. *arXiv preprint arXiv:1410.8251*, 2014.
- Edwards, H. and Storkey, A. Censoring representations with an adversary. In *ICLR*, 2015.
- Faust, K. Comparison of methods for positional analysis: Structural and general equivalences. *Soc. Net.*, 10(4):313–341, 1988.
- Gajane, P. and Pechenizkiy, M. On formalizing fairness in prediction with machine learning. *arXiv preprint arXiv:1710.03184*, 2017.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *NIPS*, pp. 2672–2680, 2014.
- Grover, A. and Leskovec, J. node2vec: Scalable feature learning for networks. In *KDD*, 2016.
- Hamilton, W., Ying, R., and Leskovec, J. Inductive representation learning on large graphs. In *NIPS*, 2017a.
- Hamilton, W., Ying, R., and Leskovec, J. Representation learning on graphs: Methods and applications. *IEEE Data Engineering Bulletin*, 2017b.
- Harper, F. M. and Konstan, J. A. The movielens datasets: History and context. *AcM transactions on interactive intelligent systems (tiis)*, 5(4):19, 2016.
- Ji, G., He, S., Xu, L., Liu, K., and Zhao, J. Knowledge graph embedding via dynamic mapping matrix. In *ACL*, 2015.
- Kamishima, T., Akaho, S., Asoh, H., and Sakuma, J. Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 35–50. Springer, 2012.
- Kearns, M., Neel, S., Roth, A., and Wu, Z. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. *arXiv:1711.05144*, 2017.
- Kipf, T. and Welling, M. Variational graph auto-encoders. In *NIPS Workshop on Bayesian Deep Learning*, 2016.
- Madras, D., Creager, E., Pitassi, T., and Zemel, R. Learning adversarially fair and transferable representations. 2018.
- Majone, G. Social space and social distance: some remarks on metric methods in data analysis. *Quality and Quantity*, 6(2):239–252, 1972.
- Mathieu, M. F., Zhao, J. J., Zhao, J., Ramesh, A., Sprechmann, P., and LeCun, Y. Disentangling factors of variation in deep representation using adversarial training. In *Advances in Neural Information Processing Systems*, pp. 5040–5048, 2016.
- Mnih, A. and Teh, Y. W. A fast and simple algorithm for training neural probabilistic language models. 2012.
- Moon, C., Jones, P., and Samatova, N. F. Learning entity type embeddings for knowledge graph completion. In *CIKM*, 2017.
- Ng, A., Jordan, M., Weiss, Y., et al. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001.
- Perozzi, B., Al-Rfou, R., and Skiena, S. Deepwalk: Online learning of social representations. In *KDD*, 2014.
- Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P., and Gamon, M. Representing text for joint embedding of text and knowledge bases. In *EMNLP*, 2015.
- Wang, D., Cui, P., and Zhu, W. Structural deep network embedding. In *KDD*, 2016.

Xu, B., Wang, N., Chen, T., and Li, M. Empirical evaluation of rectified activations in convolutional network. *Deep Learning Workshop, ICML 2015*, 2015.

Yao, S. and Huang, B. New fairness metrics for recommendation that embrace differences. In *Workshop on Fairness, Accountability, and Transparency in Machine Learning (FATML)*, June 2017.

Zemel, R., Wu, Y., Swersky, K., Pitassi, T., and Dwork, C. Learning fair representations. In *International Conference on Machine Learning*, pp. 325–333, 2013.

A. Implementation Details

We implement each discriminator and adversarial filter as multi-layer perceptrons (MLPs) with a leaky ReLU non-linearity between layers, and we use the Adam optimizer with default parameters. Unless otherwise specified we use $\lambda = 1000$ for all experiments and datasets. For fair comparison our discriminator during training time and subsequent sensitive attribute classifier share the same architecture and capacity. Finally, for every step performed by the main encoding model the Discriminator is updated 5 times. We found that this was necessary to provide a sufficient supervisory signal to the main encoding model.

B. FB15k-237 Details

To generate negative triplets we randomly sample either a head or tail entity during training, with a ratio of 20 negatives for each positive triplet. The TransD model is trained for 100 epochs with an embedding dimension of 20, selected using cross-validation, while the sensitive attribute classifiers are trained for 50 epochs. The discriminators, sensitive attribute classifier and adversarial filters are modelled as MLP's with 4,4 and 2 layers respectively. Lastly, we use the training, validation and testing splits provided in the datasets.

C. MovieLens1M

As with FB15k-237 we use model the discriminators and sensitive attribute classifiers are modelled as MLP's but 9 layers with dropout with $p = 0.3$ between layers while the adversarial filter remains unchanged from FB15k-237. We found that regularization was crucial to the performance of main model and we use BatchNorm after the embedding lookup in the main model which has an embedding dimensionality of 30. As only user nodes contain sensitive attributes our discriminators do not compute losses using movie nodes. Finally, to train our sensitive attribute classifier we construct a 90% split of all users while the remaining user nodes are used for test. The same ratio of train/test is used for the actual dataset which contains users,movies and corresponding ratings for said movies. Finally, we train the main model and sensitive attribute classifiers for 200 epochs.

D. Reddit

Like FB15k-237 we generate negative triplets by either sampling head or tail entities which are either users or subreddits but unlike FB15k-237 we keep the ratio of negatives and positives the same. We also inherit the same architectures for discriminator, sensitive attribute classifier and attribute filters used in MovieLens1M. The main model however

Table 4. Average AUC values across top-k sensitive attributes for Reddit. The results are reported on a Held Out test of different combinations of attributes.

REDDIT	HELD OUT AUC
20 SENSITIVE ATTRIBUTES	0.569
30 SENSITIVE ATTRIBUTES	0.569
40 SENSITIVE ATTRIBUTES	0.556
50 SENSITIVE ATTRIBUTES	0.519

uses an embedding dimensionality of 50. Similar to MovieLens1M only user nodes contain sensitive attributes and as such the discriminator and sensitive attribute classifier does not compute losses with respect to subreddit nodes. Also, our training set comprises of a 90% split of all edges while the remaining 10% is used as a test set. To test compositional generalizability we held out 10% of user nodes. Lastly, we train the main model for 50 epochs and the sensitive attribute classifier for 100 epochs.

E. Additional Results on Reddit

To the test degree of which invariance is affected by the number of sensitive attributes we report additional results on the Reddit dataset. Specifically, we report results for the Held out set with 20, 30, 40, and 50 sensitive attributes. Overall, these results show no statistically significant degradation in terms of invariance performance or task accuracy.