A. Proofs

In this section we provide the detailed proofs of both theorems in the main text. We first rigorously show the relationship between the adversarial advantage and the inference error made by a worst-case adversarial:

Claim 4. $1 - \mathrm{Adv}_{\mathcal{D}}(\mathcal{F}_A) = \inf_{f\in\mathcal{F}_A} \left(\Pr_{\mathcal{D}}(f(Z) = 1 \mid A = 0) + \Pr_{\mathcal{D}}(f(Z) = 0 \mid A = 1)\right)$.

Proof. Recall that $\mathcal{F}_A$ is symmetric, hence $\forall f \in \mathcal{F}_A$, $1 - f \in \mathcal{F}_A$ as well:

$$\begin{aligned} 1 - \mathrm{Adv}_{\mathcal{D}}(\mathcal{F}_A) &= 1 - \sup_{f\in\mathcal{F}_A} \left|\Pr_{\mathcal{D}}(f(Z) = 0 \mid A = 0) - \Pr_{\mathcal{D}}(f(Z) = 0 \mid A = 1)\right| \\ &= 1 - \sup_{f\in\mathcal{F}_A} \left(\Pr_{\mathcal{D}}(f(Z) = 0 \mid A = 0) - \Pr_{\mathcal{D}}(f(Z) = 0 \mid A = 1)\right) \\ &= \inf_{f\in\mathcal{F}_A} \left(\Pr_{\mathcal{D}}(f(Z) = 1 \mid A = 0) + \Pr_{\mathcal{D}}(f(Z) = 0 \mid A = 1)\right), \end{aligned}$$

where the second equality above is because

$$\sup_{f\in\mathcal{F}_A} \left(\Pr_{\mathcal{D}}(f(Z) = 0 \mid A = 0) - \Pr_{\mathcal{D}}(f(Z) = 0 \mid A = 1)\right)$$

is always non-negative due to the symmetric assumption of $\mathcal{F}_A$. ■

Before we prove the lower bound in Theorem 1, we first need to introduce the following lemma, which is known as the data-processing inequality of the TV distance.

Lemma 5 (Data-processing of the TV distance)**.** Let $\mathcal{D}$ and $\mathcal{D}'$ be two distributions over the same sample space and g be a Markov kernel of the same space, then $d_{\mathrm{TV}}(g_\sharp\mathcal{D}, g_\sharp\mathcal{D}') \leq d_{\mathrm{TV}}(\mathcal{D}, \mathcal{D}')$, where $g_\sharp\mathcal{D}(g_\sharp\mathcal{D}')$ is the pushforward of $\mathcal{D}(\mathcal{D}')$.

Lemma 6 (Contraction of the Wasserstein distance)**.** Let $f : \mathcal{Z} \to \mathcal{Y}$ and $C > 0$ be a constant such that $\|f\|_L \leq C$. For any distributions $\mathcal{D}, \mathcal{D}'$ over $\mathcal{Z}$, $W_1(f_\sharp\mathcal{D}, f_\sharp\mathcal{D}') \leq C \cdot W_1(\mathcal{D}, \mathcal{D}')$.

Proof. We use the dual representation of the Wasserstein distance to prove this lemma:

$$\begin{aligned} W_1(f_\sharp\mathcal{D}, f_\sharp\mathcal{D}') &= \sup_{\|f'\|_L\leq 1} \left|\int f'\, d(f_\sharp\mathcal{D}) - \int f'\, d(f_\sharp\mathcal{D}')\right| \\ &= \sup_{\|f'\|_L\leq 1} \left|\int f'\circ f\, d\mathcal{D} - \int f'\circ f\, d\mathcal{D}'\right| \\ &\leq \sup_{\|h\|_L\leq C} \left|\int h\, d\mathcal{D} - \int h\, d\mathcal{D}'\right| \\ &= CW_1(\mathcal{D}, \mathcal{D}'), \end{aligned}$$

where the inequality is due to the fact that for $\|f'\|_L \leq 1$, $\|f' \circ f\|_L \leq \|f'\|_L \cdot \|f\|_L = C$. ■

The following fact will also be used in the proof of Theorem 1.

Proposition 7. Let Y and Y' be two Bernoulli random variables with distributions $\mathcal{D}$ and $\mathcal{D}'$. Then $W_1(\mathcal{D}, \mathcal{D}') = |\Pr(Y = 1) - \Pr(Y' = 1)|$.

Proof. Since both Y and Y' are Bernoulli random variables taking values in $\{0, 1\}$, we solve the following linear program to compute $W_1(\mathcal{D}, \mathcal{D}')$ according to the primal definition of the Wasserstein distance. Define $D \in \mathbb{R}^{2\times 2}$ as $D_{ij} = |i - j|$ to be the distance matrix between Y and Y'. Then the solution of the following linear program (LP) gives $W_1(\mathcal{D}, \mathcal{D}')$:

$$\begin{aligned} \min_{\gamma\in\mathbb{R}^{2\times 2}} \quad & \mathrm{Tr}(\gamma D) = \sum_{i,j=0}^{1} \gamma_{ij} D_{ij} \\ \text{subject to} \quad & \sum_{i,j=0}^{1} \gamma_{ij} = 1, \gamma_{ij} \geq 0, \sum_{j=0}^{1} \gamma_{ij} = \Pr(Y = i), \sum_{i=0}^{1} \gamma_{ij} = \Pr(Y' = j). \end{aligned} \tag{5}$$

The objective function $\mathrm{Tr}(\gamma D)$ is the transportation cost of a specific coupling γ, hence the optimal γ^* corresponds to the optimal transport between Y and Y'. For this simple LP, we have

$$\sum_{i,j=0}^{1} \gamma_{ij} D_{ij} = \gamma_{01} + \gamma_{10}.$$

On the other hand, the constraint set gives

$$\begin{aligned} \gamma_{00} + \gamma_{01} &= \Pr(Y = 0); & \gamma_{00} + \gamma_{10} &= \Pr(Y' = 0); \\ \gamma_{10} + \gamma_{11} &= \Pr(Y = 1); & \gamma_{01} + \gamma_{11} &= \Pr(Y' = 1); \end{aligned}$$

From which we observe

$$|\gamma_{01} - \gamma_{10}| = |\Pr(Y = 1) - \Pr(Y' = 1)| = |\Pr(Y = 0) - \Pr(Y' = 0)|,$$

hence,

$$\begin{aligned} (\gamma_{01} + \gamma_{10}) + |\gamma_{01} - \gamma_{10}| &= 2\max\{\gamma_{01}, \gamma_{10}\} \\ &\geq 2|\gamma_{01} - \gamma_{10}|, \end{aligned}$$

which implies $\forall\gamma$ that is feasible,

$$\mathrm{Tr}(\gamma D) = \gamma_{01} + \gamma_{10} \geq 2|\gamma_{01} - \gamma_{10}| - |\gamma_{01} - \gamma_{10}| = |\Pr(Y = 1) - \Pr(Y' = 1)|.$$

To see that this lower bound is attainable, without loss of generality, assuming that $\Pr(Y = 1) \geq \Pr(Y' = 1)$, the following γ^* suffices:

$$\gamma^*_{00} = \Pr(Y = 0); \quad \gamma^*_{01} = 0; \quad \gamma^*_{10} = \Pr(Y = 1) - \Pr(Y' = 1); \quad \gamma^*_{11} = \Pr(Y' = 1).$$ ■

With the above tools, we are ready to prove Theorem 1:

Theorem 1. Let Z be the node representations produced by a GNN g and $\mathcal{F}_A$ be the set of all binary predictors. Define $\delta_{Y|A} := |\Pr_{\mathcal{D}_0}(Y = 1) - \Pr_{\mathcal{D}_1}(Y = 1)|$. Then for a classifier h such that $\|h\|_L \leq C$,

$$\begin{aligned} &\varepsilon_{Y|A=0}(h \circ g) + \varepsilon_{Y|A=1}(h \circ g) \\ &\geq \delta_{Y|A} - C \cdot W_1(g_\sharp \mathcal{D}_0, g_\sharp \mathcal{D}_1) \\ &\geq \delta_{Y|A} - 2RC \cdot \mathrm{Adv}_{\mathcal{D}}(\mathcal{F}_A). \end{aligned} \tag{3}$$

Proof. Let $g_\sharp\mathcal{D}$ be the induced (pushforward) distribution of $\mathcal{D}$ under the GNN feature encoder g. To simplify the notation, we also use $\mathcal{D}_0$ and $\mathcal{D}_1$ to denote the conditional distribution of $\mathcal{D}$ given $A = 0$ and $A = 1$, respectively. Since $h : \mathcal{Z} \to \{0, 1\}$ is the task predictor, it follows that $(h \circ g)_\sharp\mathcal{D}_0$ and $(h \circ g)_\sharp\mathcal{D}_1$ induce two distributions over $\{0, 1\}$. Recall that $W_1(\cdot, \cdot)$ is a distance metric over the space of probability distributions, by a chain of triangle inequalities, we have:

$$\begin{aligned} W_1(\mathcal{D}(Y \mid A = 0), \mathcal{D}(Y \mid A = 1)) &\leq W_1(\mathcal{D}(Y \mid A = 0), (h \circ g)_\sharp\mathcal{D}_0) \\ &+ W_1((h \circ g)_\sharp\mathcal{D}_0, (h \circ g)_\sharp\mathcal{D}_1) + W_1((h \circ g)_\sharp\mathcal{D}_1, \mathcal{D}(Y \mid A = 1)). \end{aligned}$$

Now by Lemma 6, we have

$$W_1((h \circ g)_\sharp\mathcal{D}_0, (h \circ g)_\sharp\mathcal{D}_1) \leq C \cdot W_1(g_\sharp\mathcal{D}_0, g_\sharp\mathcal{D}_1).$$

Next we bound $W_1(\mathcal{D}(Y \mid A = a), (h \circ g)_\sharp\mathcal{D}_a)$, $\forall a \in \{0, 1\}$:

$$\begin{aligned} W_1(\mathcal{D}(Y \mid A = a), (h \circ g)_\sharp\mathcal{D}_a) &= |\Pr_{\mathcal{D}}(Y = 1 \mid A = a) - \Pr_{\mathcal{D}}((h \circ g)(X) = 1 \mid A = a)| \\ &\qquad \text{(Lemma 6, Both } Y \text{ and } h(g(X)) \text{ are binary)} \\ &= |\mathbb{E}_{\mathcal{D}}[Y \mid A = a] - \mathbb{E}_{\mathcal{D}}[(h \circ g)(X) \mid A = a]| \\ &\leq \mathbb{E}_{\mathcal{D}}[|Y - (h \circ g)(X)| \mid A = a] && \text{(Triangle inequality)} \end{aligned}$$

$$
\begin{aligned}
&= \Pr_{\mathcal{D}}(Y \neq (h \circ g)(X) \mid A = a) \\
&\leq \varepsilon_{Y|A=a}(h \circ g),
\end{aligned}
$$

where the last inequality is due to the fact that the cross-entropy loss is an upper bound of the 0-1 binary loss. Again, realizing that both $\mathcal{D}(Y \mid A = 0)$ and $\mathcal{D}(Y \mid A = 1)$ are Bernoulli distributions, applying Lemma 6, we have

$$W_1(\mathcal{D}(Y \mid A = 0), \mathcal{D}(Y \mid A = 1)) = \delta_{Y|A}.$$

Combining all the inequalities above, we establish the following inequality:

$$\varepsilon_{Y|A=0}(h \circ g) + \varepsilon_{Y|A=1}(h \circ g) \geq \delta_{Y|A} - C \cdot W_1(g_\sharp \mathcal{D}_0, g_\sharp \mathcal{D}_1).$$

For the second part of the inequality, since $\sup_{z \in \mathcal{Z}} \|z\| \leq R$, the diameter of $\mathcal{Z}$ is bounded by $2R$. Now using the classic result between the TV distance and the Wasserstein distance over a metric space (Gibbs & Su, 2002), we have

$$W_1(g_\sharp \mathcal{D}_0, g_\sharp \mathcal{D}_1) \leq 2R \cdot d_{\text{TV}}(g_\sharp \mathcal{D}_0, g_\sharp \mathcal{D}_1),$$

To complete the proof, we show that $d_{\text{TV}}(g_\sharp \mathcal{D}_0, g_\sharp \mathcal{D}_1) = \text{Adv}_{\mathcal{D}}(\mathcal{F}_A)$: since $\mathcal{F}_A$ contains all the binary predictors,

$$
\begin{aligned}
d_{\text{TV}}(g_\sharp \mathcal{D}_0, g_\sharp \mathcal{D}_1) &= \sup_{E \text{ is measurable}} \left| \Pr_{g_\sharp \mathcal{D}_0}(E) - \Pr_{g_\sharp \mathcal{D}_1}(E) \right| \\
&= \sup_{f_E \in \mathcal{F}_A} \left| \Pr_{g_\sharp \mathcal{D}_0}(f_E(Z) = 1) - \Pr_{g_\sharp \mathcal{D}_1}(f_E(Z) = 1) \right| \\
&= \sup_{f_E \in \mathcal{F}_A} \left| \Pr_{g_\sharp \mathcal{D}}(f_E(Z) = 1 \mid A = 0) - \Pr_{g_\sharp \mathcal{D}}(f_E(Z) = 1 \mid A = 1) \right| \\
&= \text{Adv}_{\mathcal{D}}(\mathcal{F}_A),
\end{aligned}
$$

where in the second equation above $f_E(\cdot)$ is the characteristic function of the event E. Now combining the above two inequalities together, we have:

$$
\begin{aligned}
\varepsilon_{Y|A=0}(h \circ g) + \varepsilon_{Y|A=1}(h \circ g) &\geq \delta_{Y|A} - C \cdot W_1(g_\sharp \mathcal{D}_0, g_\sharp \mathcal{D}_1) \\
&\geq \delta_{Y|A} - 2RC \cdot \text{Adv}_{\mathcal{D}}(\mathcal{F}_A).
\end{aligned}
$$

■

Corollary 2 then follows directly from Theorem 1:

Corollary 2. Assume the conditions in Theorem 1 hold. Let $\alpha := \Pr_{\mathcal{D}}(A = 0)$, then

$$
\begin{aligned}
\varepsilon_Y(h \circ g) &\geq \min\{\alpha, 1 - \alpha\}\big(\delta_{Y|A} - C \cdot W_1(g_\sharp \mathcal{D}_0, g_\sharp \mathcal{D}_1)\big) \\
&\geq \min\{\alpha, 1 - \alpha\}\big(\delta_{Y|A} - 2RC \cdot \text{Adv}_{\mathcal{D}}(\mathcal{F}_A)\big).
\end{aligned}
$$

Proof. Realize that

$$
\begin{aligned}
\varepsilon_Y(h \circ g) &= \Pr_{\mathcal{D}}(A = 0) \cdot \varepsilon_{Y|A=0}(h \circ g) + \Pr_{\mathcal{D}}(A = 1) \cdot \varepsilon_{Y|A=1}(h \circ g) \\
&\geq \min\{\Pr_{\mathcal{D}}(A = 0), \Pr_{\mathcal{D}}(A = 1)\} \cdot \big(\varepsilon_{Y|A=0}(h \circ g) + \varepsilon_{Y|A=1}(h \circ g)\big).
\end{aligned}
$$

Applying the lower bound in Theorem 1 then completes the proof. ■

The following lemma about the inverse binary entropy will be useful in the proof of Theorem 3:

Lemma 8 (Calabro (2009))**.** Let $H_2^{-1}(s)$ be the inverse binary entropy function for $s \in [0, 1]$, then $H_2^{-1}(s) \geq s/2 \lg(6/s)$.

With the above lemma, we are ready to prove Theorem 3.

Theorem 3. Let Z^* be the optimal GNN node embedding of (4). Define $\alpha := \Pr_{\mathcal{D}}(A = 0)$, $H^* := H(A \mid Z^*)$ and $W_1^* := W_1(Z^* \mid A = 0, Z^* \mid A = 1)$. Then 1). For any adversary $f : \mathcal{Z} \to \{0, 1\}$, $\Pr(f(Z) \neq A) \geq H^*/2\lg(6/H^*)$, 2). For any Lipschitz adversary f such that $\|f\|_L \leq C$, $\Pr(f(Z) \neq A) \geq \min\{\alpha, 1-\alpha\}(1 - CW_1^*)$.

Proof. To ease the presentation we define $Z = Z^*$. To prove this theorem, let E be the binary random variable that takes value 1 iff $A \neq f(Z)$, i.e., $E = \mathbb{I}(A \neq f(Z))$. Now consider the joint entropy of $A, f(Z)$ and E. On one hand, we have:

$$H(A, f(Z), E) = H(A, f(Z)) + H(E \mid A, f(Z)) = H(A, f(Z)) + 0 = H(A \mid f(Z)) + H(f(Z)).$$

Note that the second equation holds because E is a deterministic function of A and $f(Z)$, that is, once A and $f(Z)$ are known, E is also known, hence $H(E \mid A, f(Z)) = 0$. On the other hand, we can also decompose $H(A, f(Z), E)$ as follows:

$$H(A, f(Z), E) = H(E) + H(A \mid E) + H(f(Z) \mid A, E).$$

Combining the above two equalities yields

$$H(E, A \mid f(Z)) = H(A \mid f(Z)).$$

On the other hand, we can also decompose $H(E, A \mid f(Z))$ as

$$H(E, A \mid f(Z)) = H(E \mid f(Z)) + H(A \mid E, f(Z)).$$

Furthermore, since conditioning cannot increase entropy, we have $H(E \mid f(Z)) \leq H(E)$, which further implies

$$H(A \mid f(Z)) \leq H(E) + H(A \mid E, f(Z)).$$

Now consider $H(A \mid E, f(Z))$. Since $A \in \{0, 1\}$, by definition of the conditional entropy, we have:

$$H(A \mid E, f(Z)) = \Pr(E = 1)H(A \mid E = 1, f(Z)) + \Pr(E = 0)H(A \mid E = 0, f(Z)) = 0 + 0 = 0.$$

To lower bound $H(A \mid f(Z))$, realize that

$$I(A; f(Z)) + H(A \mid f(Z)) = H(A) = I(A; Z) + H(A \mid Z).$$

Since $f(Z)$ is a randomized function of Z such that $A \perp f(Z) \mid Z$, due to the celebrated data-processing inequality, we have $I(A; f(Z)) \leq I(A; Z)$, which implies

$$H(A \mid f(Z)) \geq H(A \mid Z).$$

Combine everything above, we have the following chain of inequalities hold:

$$H(A \mid Z) \leq H(A \mid f(Z)) \leq H(E) + H(A \mid E, f(Z)) = H(E),$$

which implies

$$\Pr(A \neq f(Z)) = \Pr(E = 1) \geq H_2^{-1}(H(A \mid Z)),$$

where $H_2^{-1}(\cdot)$ denotes the inverse function of the binary entropy $H(t) := -t \log t - (1-t)\log(1-t)$ when $t \in [0, 1]$. We then apply Lemma 8 to further lower bound the inverse binary entropy function by

$$\Pr(A \neq f(Z)) \geq H_2^{-1}(H(A \mid Z)) \geq H(A \mid Z)/2\lg(6/H(A \mid Z)),$$

completing the proof of the first lower bound. For the second part, realize that

$$\begin{aligned}\Pr(f(Z) \neq A) &= \Pr(A = 0)\Pr(f(Z) = 1 \mid A = 0) + \Pr(A = 1)\Pr(f(Z) = 0 \mid A = 1)\\ &\geq \min\{\alpha, 1-\alpha\}\left(\Pr_{\mathcal{D}_0}(f(Z) = 1) + \Pr_{\mathcal{D}_1}(f(Z) = 0)\right).\end{aligned}$$

Now to lower bound $\Pr_{\mathcal{D}_a}(f(Z) = 1 - a)$, we apply the same argument in the proof of Theorem 1, which gives us

$$\Pr_{\mathcal{D}_a}(f(Z) = 1 - a) = \Pr_{\mathcal{D}}(f(Z) \neq A \mid A = a)$$

$$\begin{aligned}
&= \mathbb{E}_{\mathcal{D}}[|f(Z) - A| \mid A = a] \\
&\geq |\mathbb{E}_{\mathcal{D}}[f(Z) \mid A = a] - \mathbb{E}_{\mathcal{D}}[A \mid A = a]| \\
&= |\Pr_{\mathcal{D}}(f(Z) = 1 \mid A = a) - \Pr_{\mathcal{D}}(A = 1 \mid A = a)| \\
&= W_1(\mathcal{D}_a(f(Z)), A \mid A = a) \qquad \text{(Lemma 6, Both } A \text{ and } f(Z) \text{ are binary).}
\end{aligned}$$

As a last step, using the triangle inequality of $W_1(\cdot, \cdot)$ and Lemma 6, we have

$$W_1(\mathcal{D}_0(f(Z)), A \mid A = 0) + W_1(\mathcal{D}_1(f(Z)), A \mid A = 1) \geq \delta_{A|A} - CW_1^* = 1 - CW_1^*.$$

Combining all the steps above yields

$$\begin{aligned}
\Pr(f(Z) \neq A) &\geq \min\{\alpha, 1-\alpha\} \left(\Pr_{\mathcal{D}_0}(f(Z) = 1) + \Pr_{\mathcal{D}_1}(f(Z) = 0) \right) \\
&\geq \min\{\alpha, 1-\alpha\} \left(W_1(\mathcal{D}_0(f(Z)), A \mid A = 0) + W_1(\mathcal{D}_1(f(Z)), A \mid A = 1) \right) \\
&\geq \min\{\alpha, 1-\alpha\}(1 - CW_1^*),
\end{aligned}$$

which completes the second part of the proof. ■

B. Experimental Setup Details

Optimization For the objective function, we selected block gradient descent-ascent to optimize our models. In particular, we took advantage of the optim module in PyTorch (Paszke et al., 2019) by designing a custom gradient-reversal layer, first introduced by (Ganin et al., 2016), to be placed between the attacker and the GNN layer we seek to defend. The implementation of the graident-reversal layer can be found in the Appendix. During training, we would designate two Optimizer instances, one having access to only task-related parameters, and the other having access to attack-related parameters and parameters associated with GNN defense. We could then call the .step() method on the optimizers in an alternating fashion to train the entire network, where the gradient-reversal layer would carry out both gradient descent (of the attacker) and ascent (of protected layers) as expected. Tradeoff control via λ is achieved through multiplying the initial learning rate of the adversarial learner by the desired factor. For graphs that are harder to optimize, we introduce pre-training as the first step in the pipeline, where we train the encoder and the task decoder for a few epochs before introducing the adversarial learner.

Movielens 1M The main dataset of interest for this work is Movielens-1M [3], a benchmarking dataset in evaluating recommender systems, developed by (Harper & Konstan, 2015). In this dataset, nodes are either users or movies, and the type of edge represents the rating the user assigns to a movie. Adapting the formulation of (Bose & Hamilton, 2019b), we designate the main task as edge prediction and designate the adversarial task as extracting user-related information from the GNN embedding using multi-layer perceptrons with LeakyReLU functions (Maas, 2013) as nonlinearities. Training/test splits are creating using a random 90/10 shuffle. The network encoder consists of a trainable embedding layer followed by neighborhood aggregation layers. Node-level embeddings have a dimension of 20, and the decoder is a naive bilinear decoder, introduced in (Berg et al., 2017). Both the adversarial trainers and the main task predictors are trained with separate Adam optimizers with learning rate set to 0.01. Worst-case attackers are trained for 30 epochs with a batch-size 256 nodes before the original model is trained for 25 epochs with a batch-size of 8,192 edges.

Planetoid Planetoid [4] is the common name for three datasets (Cora, CiteSeer, Pubmed) used in benchmarks of graph neural networks in the literature, introduced by (Yang et al., 2016). Nodes in these datasets represent academic publications, and edges represent citation links. Since the Cora dataset is considered to be small to have any practical implications in the performance of our algorithm, we report only the results of CiteSeer and Pubmed. Similar to Movielens, the main task is edge prediction, and the attacker will seek to predict node attributes from GNN-processed embeddings. The network architecture is message-passing layers connected with ReLU nonlinearities, and both the decoder and attacker are also

[3] https://grouplens.org/datasets/movielens/1m/

[4] Raw data available at https://github.com/kimiyoung/planetoid/tree/master/data. For this work, we used the wrapper provided by https://pytorch-geometric.readthedocs.io/en/latest/_modules/torch_geometric/datasets/planetoid.html.

single-layer message-passing modules. Regarding training/valid/test splits, we adopt the default split used in the original paper, maintained by (Fey & Lenssen, 2019). The network encoder consists of a trainable embedding layer followed by neighborhood aggregation layers. Node-level embeddings have a dimension of 64, and both the adversarial trainers and the main task predictors are trained with separate Adam optimizers with learning rate set to 0.01. Worst-case attackers are trained for 80 epochs with before the original model is trained for 150 epochs, and the entire graph is fed into the network at once during each epoch.

QM9 QM9 [5] is a dataset used to benchmark machine learning algorithms in quantum chemistry (Wu et al., 2017), consisting of around 130,000 molecules (represented in their spatial information of all component atoms) and 19 regression targets. The main task would be to predict the dipole moment μ for a molecule graph, while the attacker will seek to extract its isotropic polarizability α from the embeddings. The encoder is a recurrent architecture consisting of a NNConv (Gilmer et al., 2017) unit, a GRU (Cho et al., 2014) unit and a Set2Set (Vinyals et al., 2015) unit, with both the decoder and the attacker (as regressors) 2-layer multi-layer perceptrons with ReLU nonlinearities. The training/valid/test is selected in the following manner: the order of samples is randomly shuffled at first, then the first 10,000 and 10,000 - 20,000 samples are selected for testing and validation respectively, and the remaining samples are used for training. Preprocessing is done with scripts provided by (Fey & Lenssen, 2019) [6] , using functions from (Landrum). Node-level embeddings have a dimension of 64, and both the adversarial trainers and the main task predictors are trained with separate Adam optimizers with learning rate set to 0.001. Worst-case attackers are trained for 30 epochs with before the original model is trained for 40 epochs with a batch-size of 128 molecular graphs.

FB15k-237/WN18RR These two datasets are benchmarks for knowledge base completion: while FB15k-237 [7] is semi-synthetic with nodes as common entities, WN18RR [8] is made by words found in the thesaurus. Our formulation is as follows: while the main task from both datasets is edge prediction, the attackers' goals are different:

- For FB15k-237, we took node-level attributes from (Moon et al., 2017) [9], and task the attacker with predicting the 50-most frequent labels. Since a node in FB15k-237 may have multiple labels associated with it, adversarial defense on this may be seen as protecting sets of node-level attributes, in contrast to single-attribute defense in other experimental settings.
- For WN18RR, we consider two attributes for a node (as a word): its word sense (sense greater than 20 are considered as the same heterogeneous class), and part-of-speech tag. The labels are obtained from (Bordes et al., 2013) [10].

As for the architecture, we used a modified version of the CompGCN paper (Vashishth et al., 2019), where the attacker has access to the output of the CompGCN layer (of dimension 200), and the original task utilizes the ConvE model for the decoder. The training/valid/test split also aligns with the one used in the CompGCN paper. On both datasets, the adversarial trainers and main task predictors are trained with separate Adam optimizers with learning rate set to 0.001. Worst-case attackers are trained for 30 epochs with a batch-size of 128 nodes before the original model is trained for 120 epochs after 35 epochs of pre-training, with a batch-size of 128 nodes.

Computing Infrastructure All models are trained with NVIDIA GeForce® RTX 2080 Ti graphics processing units (GPU) with 11.0 GB GDDR6 memory on each card, and non-training-related operations are performed using Intel® Xeon® Processor E5-2670 (20M Cache, 2.60 GHz).

[5] Raw data available at https://s3-us-west-1.amazonaws.com/deepchem.io/datasets/molnet_publish/qm9.zip and https://ndownloader.figshare.com/files/3195404

[6] Available at https://pytorch-geometric.readthedocs.io/en/latest/_modules/torch_geometric/datasets/qm9.html

[7] https://www.microsoft.com/en-us/download/details.aspx?id=52312

[8] https://github.com/TimDettmers/ConvE

[9] https://github.com/cmoon2/knowledge_graph

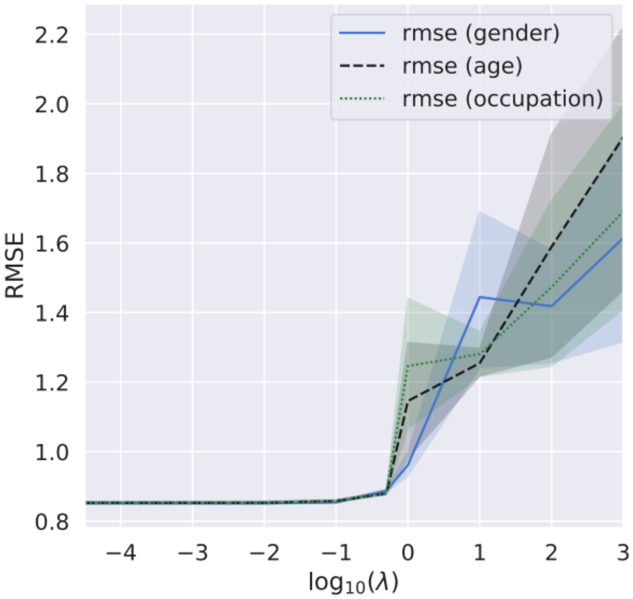
[10] https://everest.hds.utc.fr/doku.php?id=en:smemlj12

Estimated Average Runtime Below are the averge training time per epoch for each models used in the main text, when the training is performed on the computing infrastructure mentioned above:

DATASET	Encoder	t
CITESEER	ChebNet	0.0232s
	GCN	0.0149s
	GAT	0.0282s
PUBMED	ChebNet	0.0920s
	GCN	0.0824s
	GAT	0.129s
QM9	MPNN	199.25s
MOVIELENS-1M	GCN	12.05s
	GAT	45.86s
FB15K-237	CompGCN	463.39s
WN18RR	CompGCN	181.55s

C. Degredation of RMSE on Movielens-1M dataset Regarding Neighborhood Attack

This is a supplementary figure for the neighborhood attack experiments introduced in the main section. Band represents 95% confidence interval over five runs.



D. N-Hop Algorithm for Neighborhood Defense

Intuitively, this algorithm greedily constructs a path of length n by uniformly picking a neighbor from the current end of the path and checking if the node has existed previously in the path, avoiding formation of cycles. Worst-case running time of this algorithm is $O(n^2)$, because in each step of the main loop, the algorithm performs $O(n)$ checks in the worst case scenario.

Algorithm 2 Monte-Carlo Probabilistic N-Hop

```
**Input:** $G = (V,E)$: undirected graph (via adjacency list); $v \in V$: starting node; $n \geq 1$: hop
**Output:** On success: $v' \in V$ such that $d(v, v') = n$ or NO if such vertex doesn't exist; On failure: $v' \in V$ such that $1 \leq d(v, v') \leq n$ or NO if such vertex doesn't exist
$V = \emptyset$ {Initial path is empty}
$t = 0$
$v' = v$
repeat
  $S = [\mathcal{N}(v')]$ {$O(1)$ time by adjacency list}
  $i = \text{RandInt}(0, |S|)$ {$O(1)$ uniform random sample (without replacement)}
  $e = S.\text{pop}(i)$
  repeat
    $i = \text{RandInt}(0, |S|)$
    $e = S.\text{pop}(i)$
  until $\neg(e \in V$ and $S \neq [])$ {Loop runs at most $O(n)$ times}
  if $e \notin V$ then
    $V = V \cap \{e\}$
    $v' = e$
  else
    reject with NO {Current path not satisfiable, reject}
  end if
  $t = t + 1$
until $t >= n$
accept with $v'$
```