Recent Advances in Graph Neural Network Robustness

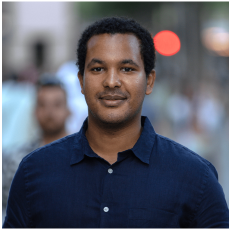
Johannes Lutzeyer

Data Science and Mining Team, Laboratoire d'Informatique (LIX),
École Polytechnique, Institut Polytechnique de Paris

May 15, 2024

Today I present work that was done in collaboration with

Sofiane Ennadir
PhD Student KTH



Yassine Abbahaddou
PhD Student LIX

Prof. Henrik Boström
Professor KTH

Prof. Michalis Vazirgiannis
Distinguished Professor LIX

Graph Representation Learning

Overall Goal: Learn "informative" representations of graph structured data

Graph Representation Learning

Overall Goal: Learn "informative" representations of graph structured data

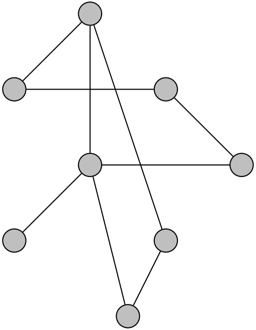
What is graph structured data?

Graph Representation Learning

Overall Goal: Learn "informative" representations of graph structured data

What is graph structured data?
It's the combination of

- a graph $G = (V, E)$;

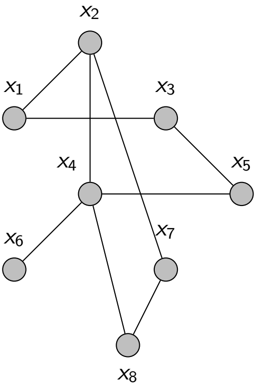


Graph Representation Learning

Overall Goal: Learn "informative" representations of graph structured data

What is graph structured data?
It's the combination of

- a graph $G = (V, E)$;
- node-features $X = [x_1, \ldots, x_n]^T$.

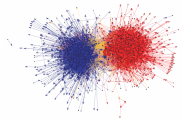


Graph Representation Learning

Overall Goal: Learn “informative” representations of graph structured data

What is graph structured data?
It’s the combination of

- a graph $G = (V, E)$;
- node-features $X = [x_1, \dots, x_n]^T$.



US political weblogs
(Adamic & Glance, 2005)

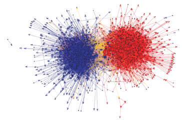
Where does it arise?

Graph Representation Learning

Overall Goal: Learn "informative" representations of graph structured data

What is graph structured data?
It's the combination of

- a graph $G = (V, E)$;
- node-features $X = [x_1, \ldots, x_n]^T$.



US political weblogs
(Adamic & Glance, 2005)

Where does it arise?

Caffeine molecule
(Bronstein, 2021)

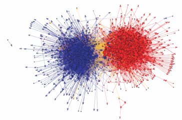
Graph Representation Learning

Overall Goal: Learn "informative" representations of graph structured data

What is graph structured data?
It's the combination of

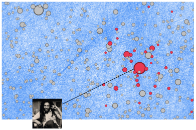
- a graph $G = (V, E)$;
- node-features $X = [x_1, \ldots, x_n]^T$.

Where does it arise?



US political weblogs
(Adamic & Glance, 2005)

Caffeine molecule
(Bronstein, 2021)



Deezer artists
(Salha-Galvan, 2022)

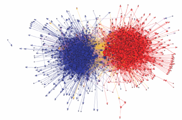
Graph Representation Learning

Overall Goal: Learn "informative" representations of graph structured data

What is graph structured data?
It's the combination of

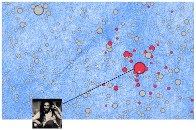
- a graph $G = (V, E)$;
- node-features $X = [x_1, \ldots, x_n]^T$.

Where does it arise?
It's ubiquitous!



US political weblogs
(Adamic & Glance, 2005)

Caffeine molecule
(Bronstein, 2021)



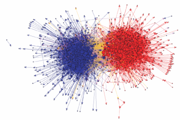
Deezer artists
(Salha-Galvan, 2022)

Graph Representation Learning

Overall Goal: Learn "informative" representations of graph structured data

What is graph structured data?
It's the combination of

- a graph $G = (V, E)$;
- node-features $X = [x_1, \dots, x_n]^T$.

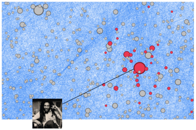


US political weblogs
(Adamic & Glance, 2005)

Where does it arise?
It's ubiquitous!

Caffeine molecule
(Bronstein, 2021)

What can we learn from it?



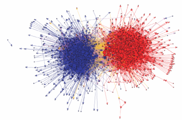
Deezer artists
(Salha-Galvan, 2022)

Graph Representation Learning

Overall Goal: Learn "informative" representations of graph structured data

What is graph structured data?
It's the combination of

- a graph $G = (V, E)$;
- node-features $X = [x_1, \ldots, x_n]^T$.



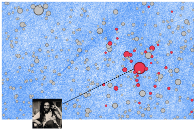
US political weblogs
(Adamic & Glance, 2005)

Where does it arise?
It's ubiquitous!

Caffeine molecule
(Bronstein, 2021)

What can we learn from it?

- Node and Graph Classification



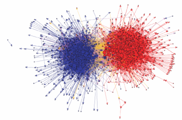
Deezer artists
(Salha-Galvan, 2022)

Graph Representation Learning

Overall Goal: Learn "informative" representations of graph structured data

What is graph structured data?
It's the combination of

- a graph $G = (V, E)$;
- node-features $X = [x_1, \ldots, x_n]^T$.



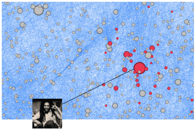
US political weblogs
(Adamic & Glance, 2005)

Where does it arise?
It's ubiquitous!

Caffeine molecule
(Bronstein, 2021)

What can we learn from it?

- Node and Graph Classification
- Node and Graph Regression



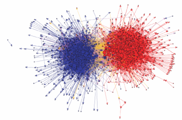
Deezer artists
(Salha-Galvan, 2022)

Graph Representation Learning

Overall Goal: Learn "informative" representations of graph structured data

What is graph structured data?
It's the combination of

- a graph $G = (V, E)$;
- node-features $X = [x_1, \dots, x_n]^T$.



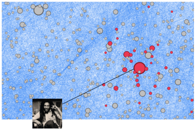
US political weblogs
(Adamic & Glance, 2005)

Where does it arise?
It's ubiquitous!

Caffeine molecule
(Bronstein, 2021)

What can we learn from it?

- Node and Graph Classification
- Node and Graph Regression
- Link Prediction



Deezer artists
(Salha-Galvan, 2022)

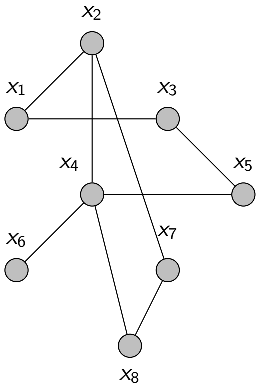
Graph Neural Networks

Graph Neural Networks (GNNs) are neural networks that take graph-structured data as input.

Graph Neural Networks

Graph Neural Networks (GNNs) are neural networks that take graph-structured data as input.

In this talk we will only see a specific type of GNN, the Message Passing Neural Networks.



Graph Neural Networks

Graph Neural Networks (GNNs) are neural networks that take graph-structured data as input.

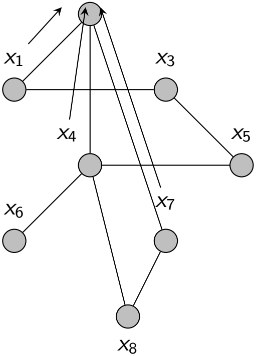
In this talk we will only see a specific type of GNN, the Message Passing Neural Networks.

$$m_v^{(k)} = M^{(k)} \left(\left\{ h_w^{(k-1)} : w \in \mathcal{N}(v) \right\} \right),$$

E.g., the Graph Convolutional Network (GCN, Kipf and Welling, 2017)

$$\tilde{A}X.$$

$$m_2^{(1)} = \frac{1}{\sqrt{d_2}} \sum_{i \in \{1,4,7\}} \frac{x_i}{\sqrt{d_i}}$$



Graph Neural Networks

Graph Neural Networks (GNNs) are neural networks that take graph-structured data as input.

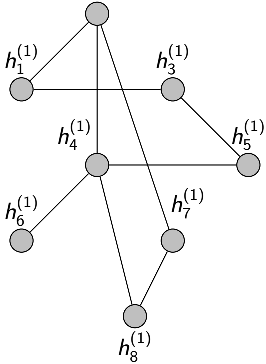
In this talk we will only see a specific type of GNN, the Message Passing Neural Networks.

$$m_v^{(k)} = M^{(k)}\left(\left\{h_w^{(k-1)} : w \in \mathcal{N}(v)\right\}\right),$$
$$h_v^{(k)} = U^{(k)}\left(h_v^{(k-1)}, m_v^{(k)}\right).$$

E.g., the Graph Convolutional Network (GCN, Kipf and Welling, 2017)

$$H^{(1)} = \mathrm{ReLU}\left(\tilde{A}XW^{(1)}\right).$$

$$h_2^{(1)} = \sigma\left(\left(\frac{x_2}{d_2} + m_2^{(1)}\right)W\right)$$



Graph Neural Networks

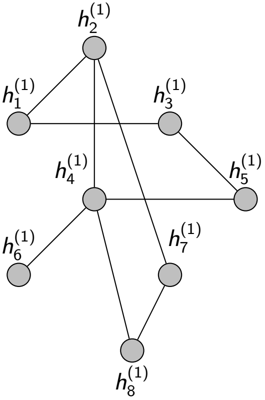
Graph Neural Networks (GNNs) are neural networks that take graph-structured data as input.

In this talk we will only see a specific type of GNN, the Message Passing Neural Networks.

$$
\begin{aligned}
m_v^{(k)} &= M^{(k)}\left(\left\{h_w^{(k-1)} : w \in \mathcal{N}(v)\right\}\right), \\
h_v^{(k)} &= U^{(k)}\left(h_v^{(k-1)}, m_v^{(k)}\right).
\end{aligned}
$$

E.g., the Graph Convolutional Network (GCN, Kipf and Welling, 2017)

$$H^{(1)} = \mathrm{ReLU}\left(\tilde{A}XW^{(1)}\right).$$



Iteratively performing the message-passing and update computations allows us to build 'deep' learning models, e.g., a 3-layer GCN

$$\hat{y} = \sigma\left(\tilde{A}\,\mathrm{ReLU}\left(\tilde{A}\,\mathrm{ReLU}\left(\tilde{A}XW^{(1)}\right)W^{(2)}\right)W^{(3)}\right).$$

Academic and Industrial Success of GNNs

Empirical and Theoretical **Research**:

- expressivity analysis of GNNs (Xu et al., 2019; Geerts and Reutter, 2022);

Academic and Industrial Success of GNNs

Empirical and Theoretical **Research**:

- expressivity analysis of GNNs (Xu et al., 2019; Geerts and Reutter, 2022);
- bottlenecks, e.g., oversmoothing and oversquashing (Alon and Yahav, 2020; Deac et al., 2022)

Academic and Industrial Success of GNNs

Empirical and Theoretical **Research**:

- expressivity analysis of GNNs (Xu et al., 2019; Geerts and Reutter, 2022);
- bottlenecks, e.g., oversmoothing and oversquashing (Alon and Yahav, 2020; Deac et al., 2022)
- robustness to adversarial attacks and noise (Günnemann, 2022; Zhou et al., 2020; Seddik et al., 2022, AISTATS).

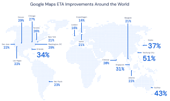
Academic and Industrial Success of GNNs

Empirical and Theoretical **Research**:

- expressivity analysis of GNNs (Xu et al., 2019; Geerts and Reutter, 2022);
- bottlenecks, e.g., oversmoothing and oversquashing (Alon and Yahav, 2020; Deac et al., 2022)
- robustness to adversarial attacks and noise (Günnemann, 2022; Zhou et al., 2020; Seddik et al., 2022, AISTATS).

Successful **Applications** of GNNs:

- Google Maps (Lange and Perez, 2020);



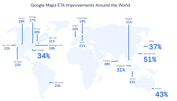
Academic and Industrial Success of GNNs

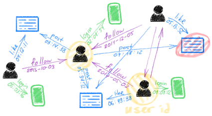
Empirical and Theoretical **Research**:

- expressivity analysis of GNNs (Xu et al., 2019; Geerts and Reutter, 2022);
- bottlenecks, e.g., oversmoothing and oversquashing (Alon and Yahav, 2020; Deac et al., 2022)
- robustness to adversarial attacks and noise (Günnemann, 2022; Zhou et al., 2020; Seddik et al., 2022, AISTATS).

Successful **Applications** of GNNs:

- Google Maps (Lange and Perez, 2020);
- Twitter (Bronstein, 2020);

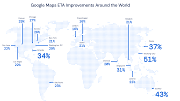


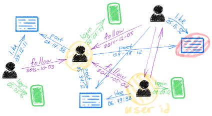


Academic and Industrial Success of GNNs

Empirical and Theoretical **Research**:

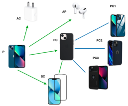
- expressivity analysis of GNNs (Xu et al., 2019; Geerts and Reutter, 2022);
- bottlenecks, e.g., oversmoothing and oversquashing (Alon and Yahav, 2020; Deac et al., 2022)
- robustness to adversarial attacks and noise (Günnemann, 2022; Zhou et al., 2020; Seddik et al., 2022, AISTATS).





Successful **Applications** of GNNs:

- Google Maps (Lange and Perez, 2020);
- Twitter (Bronstein, 2020);
- Amazon, Alibaba, Pinterest & Uber Eats (Virinchi et al., 2022; Wang et al., 2018; Ying et al., 2018; Jain et al., 2019);



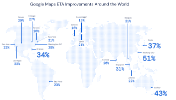
Academic and Industrial Success of GNNs

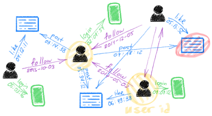
Empirical and Theoretical **Research**:

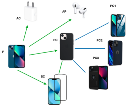
- expressivity analysis of GNNs (Xu et al., 2019; Geerts and Reutter, 2022);
- bottlenecks, e.g., oversmoothing and oversquashing (Alon and Yahav, 2020; Deac et al., 2022)
- robustness to adversarial attacks and noise (Günnemann, 2022; Zhou et al., 2020; Seddik et al., 2022, AISTATS).

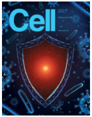
Successful **Applications** of GNNs:

- Google Maps (Lange and Perez, 2020);
- Twitter (Bronstein, 2020);
- Amazon, Alibaba, Pinterest & Uber Eats (Virinchi et al., 2022; Wang et al., 2018; Ying et al., 2018; Jain et al., 2019);
- Discovery of two *new antibiotics* (Stokes et al., 2020; Liu et al., 2023);









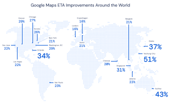
Academic and Industrial Success of GNNs

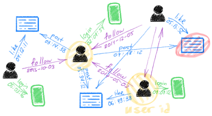
Empirical and Theoretical **Research**:

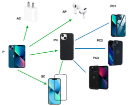
- expressivity analysis of GNNs (Xu et al., 2019; Geerts and Reutter, 2022);
- bottlenecks, e.g., oversmoothing and oversquashing (Alon and Yahav, 2020; Deac et al., 2022)
- robustness to adversarial attacks and noise (Günnemann, 2022; Zhou et al., 2020; Seddik et al., 2022, AISTATS).

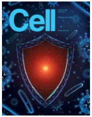
Successful **Applications** of GNNs:

- Google Maps (Lange and Perez, 2020);
- Twitter (Bronstein, 2020);
- Amazon, Alibaba, Pinterest & Uber Eats (Virinchi et al., 2022; Wang et al., 2018; Ying et al., 2018; Jain et al., 2019);
- Discovery of two *new antibiotics* (Stokes et al., 2020; Liu et al., 2023);
- LinkedIn (Borisyuk et al., 2024).







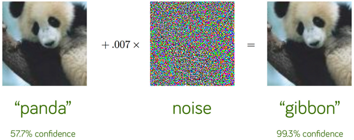


Bounding the Expected Robustness of Graph Neural Networks Subject to Node Feature Attacks

Abbahaddou*, Ennadir*, Lutzeyer, Vazirgiannis & Boström (2024, ICLR)

(Graph) Adversarial Attacks

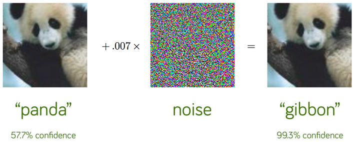
Goal: Adversarial attacks apply a *small* change to the input to achieve a *large* change in the output of our model.



(Goodfellow et al., 2015)

(Graph) Adversarial Attacks

Goal: Adversarial attacks apply a *small* change to the input to achieve a *large* change in the output of our model.

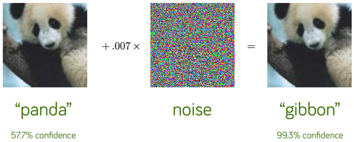


(Goodfellow et al., 2015)

To quantify the robustness of a function processing graph structured data, i.e., $f : (\mathcal{G}, \mathcal{X}) \to \mathcal{Y}$ we need:

(Graph) Adversarial Attacks

Goal: Adversarial attacks apply a *small* change to the input to achieve a *large* change in the output of our model.



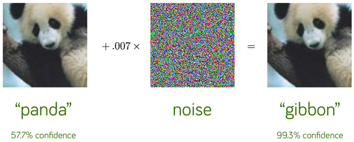
(Goodfellow et al., 2015)

To quantify the robustness of a function processing graph structured data, i.e., $f : (\mathcal{G}, \mathcal{X}) \rightarrow \mathcal{Y}$ we need:

- a distance on the input space $d_2^{\alpha,\beta}([G, X], [\tilde{G}, \tilde{X}]) = \min_{P \in \Pi} \left(\alpha \|A - P\tilde{A}P^T\|_2 + \beta \|X - P\tilde{X}\|_2 \right),$

(Graph) Adversarial Attacks

Goal: Adversarial attacks apply a *small* change to the input to achieve a *large* change in the output of our model.



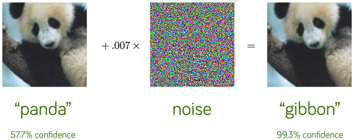
(Goodfellow et al., 2015)

To quantify the robustness of a function processing graph structured data, i.e., $f : (\mathcal{G}, \mathcal{X}) \to \mathcal{Y}$ we need:

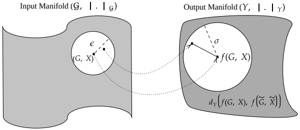
- a distance on the input space $d_2^{\alpha,\beta}([G, X], [\tilde{G}, \tilde{X}]) = \min_{P \in \Pi} \left(\alpha \|A - P\tilde{A}P^T\|_2 + \beta \|X - P\tilde{X}\|_2 \right)$,
- and a distance on the output space $d_1(f(\tilde{G}, \tilde{X}), f(G, X)) = \|f(\tilde{G}, \tilde{X}) - f(G, X)\|_1$.

(Graph) Adversarial Attacks

Goal: Adversarial attacks apply a *small* change to the input to achieve a *large* change in the output of our model.



(Goodfellow et al., 2015)



To quantify the robustness of a function processing graph structured data, i.e., $f : (\mathcal{G}, \mathcal{X}) \to \mathcal{Y}$ we need:

- a distance on the input space
  $d_2^{\alpha,\beta}([G, X], [\tilde{G}, \tilde{X}]) = \min_{P \in \Pi} \left(\alpha \|A - P\tilde{A}P^T\|_2 + \beta \|X - P\tilde{X}\|_2 \right),$
- and a distance on the output space
  $d_1(f(\tilde{G}, \tilde{X}), f(G, X)) = \|f(\tilde{G}, \tilde{X}) - f(G, X)\|_1.$

(Graph) Adversarial Attacks

Goal: Adversarial attacks apply a *small* change to the input to achieve a *large* change in the output of our model.

To quantify the robustness of a function processing graph structured data, i.e., $f : (\mathcal{G}, \mathcal{X}) \to \mathcal{Y}$ we need:

- a distance on the input space $d_2^{\alpha,\beta}([G, X], [\tilde{G}, \tilde{X}]) = \min_{P \in \Pi} \left(\alpha \|A - P\tilde{A}P^T\|_2 + \beta \|X - P\tilde{X}\|_2 \right),$
- and a distance on the output space $d_1(f(\tilde{G}, \tilde{X}), f(G, X)) = \|f(\tilde{G}, \tilde{X}) - f(G, X)\|_1.$

Expected Adversarial Robustness

Let the *expected vulnerability* of a graph function f be defined as $\mathrm{Adv}_\epsilon^{\alpha,\beta}[f] = \mathbb{P}_{(G,X)\sim\mathcal{D}_{\mathcal{G},\mathcal{X}}}[(\tilde{G}, \tilde{X}) \in B^{\alpha,\beta}(G, X, \epsilon) : d_{\mathcal{Y}}(f(\tilde{G}, \tilde{X}), f(G, X)) > \sigma],$ with $B^{\alpha,\beta}(G, X, \epsilon) = \{(\tilde{G}, \tilde{X}) : d^{\alpha,\beta}([G, X], [\tilde{G}, \tilde{X}]) < \epsilon\}$ for any budget $\epsilon \geq 0$.

(Graph) Adversarial Attacks

Goal: Adversarial attacks apply a *small* change to the input to achieve a *large* change in the output of our model.

To quantify the robustness of a function processing graph structured data, i.e., $f : (\mathcal{G}, \mathcal{X}) \to \mathcal{Y}$ we need:

- a distance on the input space $d_2^{\alpha,\beta}([G, X], [\tilde{G}, \tilde{X}]) = \min_{P \in \Pi} \left(\alpha \|A - P\tilde{A}P^T\|_2 + \beta \|X - P\tilde{X}\|_2 \right)$,
- and a distance on the output space $d_1(f(\tilde{G}, \tilde{X}), f(G, X)) = \|f(\tilde{G}, \tilde{X}) - f(G, X)\|_1$.

Expected Adversarial Robustness

Let the *expected vulnerability* of a graph function f be defined as $\mathrm{Adv}_\epsilon^{\alpha,\beta}[f] = \mathbb{P}_{(G,X)\sim\mathcal{D}_{\mathcal{G},\mathcal{X}}}[(\tilde{G}, \tilde{X}) \in B^{\alpha,\beta}(G, X, \epsilon) : d_{\mathcal{Y}}(f(\tilde{G}, \tilde{X}), f(G, X)) > \sigma]$, with $B^{\alpha,\beta}(G, X, \epsilon) = \{(\tilde{G}, \tilde{X}) : d^{\alpha,\beta}([G, X], [\tilde{G}, \tilde{X}]) < \epsilon\}$ for any budget $\epsilon \geq 0$.

Then, a graph function $f : (\mathcal{G}, \mathcal{X}) \to \mathcal{Y}$ is $((d^{\alpha,\beta}, \epsilon), (d_{\mathcal{Y}}, \gamma))$–*robust* if its vulnerability $Adv_\epsilon^{\alpha,\beta}[f]$ can be upper-bounded by γ, i.e., $Adv_\epsilon^{\alpha,\beta}[f] \leq \gamma$.

Problem Set-Up & Theoretical Results

Recall, Graph Neural Networks (GNNs) take both a graph A and node features X as input.

Problem Set-Up & Theoretical Results

Recall, Graph Neural Networks (GNNs) take both a graph A and node features X as input.

Problem: Most defense approaches for GNNs defend structural attacks altering A. There exists very little work on how to defend against attacks on the node features X.

Problem Set-Up & Theoretical Results

Recall, Graph Neural Networks (GNNs) take both a graph A and node features X as input.

Problem: Most defense approaches for GNNs defend structural attacks altering A. There exists very little work on how to defend against attacks on the node features X.

Upper Bound on GCN Vulnerability

We consider node-feature attacks on the input graph (A, X), with a budget ϵ and L-layer GCNs with weight matrices $W^{(i)}$ $i \in \{1, \dots, L\}$.
Then, the vulnerability of GCNs is upper bounded by

$$\gamma = \prod_{i=1}^{L} \|W^{(i)}\|_1 \frac{\epsilon \sum_{u \in \mathcal{V}} \hat{w}_u}{\sigma},$$

with $\hat{w}_u$ denoting the sum of normalized walks of length $(L-1)$ starting from node u.

Problem Set-Up & Theoretical Results

Recall, Graph Neural Networks (GNNs) take both a graph A and node features X as input.

Problem: Most defense approaches for GNNs defend structural attacks altering A. There exists very little work on how to defend against attacks on the node features X.

Upper Bound on GCN Vulnerability

We consider node-feature attacks on the input graph (A, X), with a budget ϵ and L-layer GCNs with weight matrices $W^{(i)}$ $i \in \{1, \ldots, L\}$.
Then, the vulnerability of GCNs is upper bounded by

$$\gamma = \prod_{i=1}^{L} \|W^{(i)}\|_1 \frac{\epsilon \sum_{u \in \mathcal{V}} \hat{w}_u}{\sigma},$$

with $\hat{w}_u$ denoting the sum of normalized walks of length $(L-1)$ starting from node u.

Insight: Our upper bound on the vulnerability of a GCN is **smaller for small** $\prod_{i=1}^{L} \|W^{(i)}\|_1$ yielding a **more robust GCN.**

Methodology

Fact: Orthonormal matrices have norm 1.

$\Rightarrow$ According to our bound a GNN with orthonormal weight matrices should be more robust.

Methodology

Fact: Orthonormal matrices have norm 1.

$\Rightarrow$ According to our bound a GNN with orthonormal weight matrices should be more robust.

Björk Orthonormalisation Algorithm

Given a weight matrix W we iteratively alter it to approximate the closest orthonormal matrix $\hat{W}$. When $\hat{W}_0 = W$, we recursively compute

$$\hat{W}_{k+1} = \hat{W}_k \left(I + \frac{1}{2}\left(I - \hat{W}_k^T \hat{W}_k \right) + \ldots + (-1)^p \binom{-1/2}{p} \left(I - \hat{W}_k^T \hat{W}_k \right)^p \right).$$

Methodology

Fact: Orthonormal matrices have norm 1.

$\Rightarrow$ According to our bound a GNN with orthonormal weight matrices should be more robust.

Björk Orthonormalisation Algorithm

Given a weight matrix W we iteratively alter it to approximate the closest orthonormal matrix $\hat{W}$. When $\hat{W}_0 = W$, we recursively compute

$$\hat{W}_{k+1} = \hat{W}_k \left(I + \frac{1}{2}\left(I - \hat{W}_k^T \hat{W}_k \right) + \ldots + (-1)^p \binom{-1/2}{p} \left(I - \hat{W}_k^T \hat{W}_k \right)^p \right).$$

Proposed Solution: In our *GCORN* model we propose the inclusion of several Björk Orthonormalisation iterations in each forward pass during the training of a GCN, **yielding weight matrices that approach orthonormality and thereby a more robust GNN**.

Results

Table: Node classification accuracy (± standard deviation) for feature-based attacks.

Attack	Dataset	GCN	GCN-k	AirGNN	RGCN	ParsevalR	GCORN
Random ($\psi = 0.5$)	Cora	68.4 ± 1.9	69.2 ± 2.6	73.5 ± 1.9	71.6 ± 0.3	72.9 ± 0.9	**77.1 ± 1.8**
	CiteSeer	57.8 ± 1.5	62.3 ± 1.2	64.6 ± 1.6	63.7 ± 0.6	65.1 ± 0.8	**67.8 ± 1.4**
	PubMed	68.3 ± 1.2	71.2 ± 1.1	70.9 ± 1.3	71.4 ± 0.5	71.8 ± 0.8	**73.1 ± 1.1**
	CS	85.3 ± 1.1	86.7 ± 1.1	87.5 ± 1.6	88.2 ± 0.9	87.6 ± 0.6	**89.8 ± 1.2**
	OGBN-Arxiv	68.2 ± 1.5	52.8 ± 0.5	66.5 ± 1.3	63.8 ± 1.9	68.3 ± 1.9	**69.1 ± 1.8**
Random ($\psi = 1.0$)	Cora	41.7 ± 2.1	46.3 ± 2.8	53.7 ± 2.2	52.8 ± 1.6	55.3 ± 1.2	**57.6 ± 1.9**
	CiteSeer	38.2 ± 1.3	45.3 ± 1.4	49.8 ± 2.1	43.7 ± 2.2	51.2 ± 1.2	**57.3 ± 1.7**
	PubMed	60.1 ± 1.7	62.3 ± 1.3	62.4 ± 1.2	61.9 ± 1.2	61.3 ± 1.7	**65.8 ± 1.4**
	CS	69.9 ± 1.3	73.2 ± 0.9	76.7 ± 2.8	76.2 ± 1.4	78.7 ± 1.2	**81.3 ± 1.6**
	OGBN-Arxiv	66.4 ± 1.9	46.6 ± 0.6	62.7 ± 1.6	63.0 ± 2.4	66.1 ± 0.7	**67.3 ± 2.1**
PGD	Cora	54.1 ± 2.4	58.3 ± 1.6	68.2 ± 1.8	62.5 ± 1.2	68.6 ± 1.7	**71.1 ± 1.4**
	CiteSeer	52.3 ± 1.1	59.6 ± 1.6	59.3 ± 2.1	61.9 ± 1.1	62.1 ± 1.5	**65.6 ± 1.4**
	PubMed	66.1 ± 2.1	67.3 ± 1.3	70.8 ± 1.7	69.5 ± 0.9	68.9 ± 2.1	**72.3 ± 1.3**
	CS	71.3 ± 1.1	74.1 ± 0.8	76.3 ± 2.1	76.6 ± 1.2	77.3 ± 0.6	**79.6 ± 1.2**
	OGBN-Arxiv	67.5 ± 0.9	49.9 ± 0.7	55.7 ± 0.9	63.6 ± 0.7	67.6 ± 1.2	**68.1 ± 1.1**
Nettack	Cora	60.9 ± 2.5	64.2 ± 5.2	66.7 ± 3.8	63.4 ± 3.8	67.5 ± 2.5	**68.3 ± 1.4**
	CiteSeer	55.8 ± 1.4	71.7 ± 1.4	67.5 ± 2.5	70.8 ± 3.8	69.2 ± 3.8	**77.5 ± 2.5**
	PubMed	60.0 ± 2.5	65.8 ± 2.9	69.2 ± 1.4	**71.7 ± 3.8**	68.3 ± 1.4	70.8 ± 1.4
	CS	55.8 ± 1.4	71.6 ± 1.4	76.7 ± 1.4	71.7 ± 2.9	75.8 ± 2.8	**78.3 ± 1.4**
	OGBN-Arxiv	49.2 ± 2.9	53.3 ± 1.4	**56.7 ± 1.4**	52.6 ± 2.5	55.8 ± 1.4	55.8 ± 1.4

- Our **GCORN model often outperforms** existing defense approaches when subject to feature based attacks.

Results

Table: Node classification accuracy (± standard deviation) for structure-based attacks.

Attack	Dataset	GCN	GCN-Jaccard	RGCN	GNN-SVD	GNN-Guard	ParsevalR	GCORN
Mettack	Cora	73.0 ± 0.7	75.4 ± 1.8	69.2 ± 0.3	73.6 ± 0.9	74.4 ± 0.8	71.9 ± 0.7	**77.3 ± 0.5**
	CiteSeer	63.2 ± 0.9	69.5 ± 1.9	68.9 ± 0.6	65.8 ± 0.6	68.8 ± 1.5	68.3 ± 0.8	**73.7 ± 0.3**
	PubMed	60.7 ± 0.7	62.9 ± 1.8	65.1 ± 0.4	82.1 ± 0.8	**84.8 ± 0.3**	69.5 ± 1.1	71.8 ± 0.4
	CoraML	73.1 ± 0.6	75.4 ± 0.4	77.1 ± 1.1	71.3 ± 1.0	76.5 ± 0.7	76.9 ± 1.3	**79.2 ± 0.6**
PGD	Cora	76.7 ± 0.9	78.3 ± 1.1	72.0 ± 0.3	71.6 ± 0.4	75.0 ± 2.0	78.4 ± 1.2	**79.9 ± 0.4**
	CiteSeer	67.8 ± 0.8	70.9 ± 1.0	62.2 ± 1.8	60.3 ± 2.4	68.9 ± 2.2	70.6 ± 1.0	**73.1 ± 0.5**
	PubMed	75.3 ± 1.6	73.8 ± 1.3	78.6 ± 0.4	81.9 ± 0.4	**84.3 ± 0.4**	77.3 ± 0.7	77.4 ± 0.4
	CoraML	76.9 ± 1.2	75.0 ± 2.4	77.5 ± 0.3	73.1 ± 0.5	75.5 ± 0.8	81.3 ± 0.4	**84.1 ± 0.2**
DICE	Cora	74.9 ± 0.8	76.9 ± 0.9	79.6 ± 0.3	72.2 ± 1.4	75.6 ± 1.1	**79.7 ± 0.8**	78.9 ± 0.4
	CiteSeer	64.1 ± 0.5	66.0 ± 0.6	68.7 ± 0.5	62.6 ± 1.2	65.5 ± 1.1	68.9 ± 0.4	**74.6 ± 0.4**
	PubMed	79.4 ± 0.4	78.3 ± 0.2	**79.8 ± 0.4**	76.6 ± 0.5	77.8 ± 0.7	79.2 ± 0.3	78.1 ± 0.6
	CoraML	78.3 ± 0.6	77.5 ± 0.3	80.1 ± 0.4	58.7 ± 0.4	77.5 ± 0.2	80.5 ± 1.3	**81.1 ± 0.8**

- Our **GCORN model often outperforms** existing defense approaches when subject to feature based attacks.
- GCORN is also effective against **structure-based, as well as combined structure and feature attacks**.

A Simple and Yet Fairly Effective Defense for Graph Neural Networks

Ennadir, Abbahaddou, Lutzeyer, Vazirgiannis & Boström (2024, AAAI)

Problem Set-Up

Problem: Available defense methods often have high computational complexity and training time (often increasing with increasing graph size).

Problem Set-Up

Problem: Available defense methods often have high computational complexity and training time (often increasing with increasing graph size).

Solution Approach: We propose a GNN, called the *NoisyGNN*, in which **hidden states are perturbed** by random noise following a normal distribution $N \sim \mathcal{N}(0, \beta I)$, i.e., our GNNs are of the form

$$\hat{y} = \sigma\left(\tilde{A}\ \mathrm{ReLU}\left(\tilde{A}XW^{(1)} + N\right) W^{(2)}\right).$$

Theoretical Results

Upper Bounds on GNN Vulnerability

We consider structural perturbations of the input graph (A, X), with a budget ϵ and 2-layer GNNs with 1-Lipschitz continuous activation functions and weight matrices $W^{(1)}, W^{(2)}$.

- Then, the vulnerability of GCNs is upper bounded by

$$\frac{2(\|W^{(2)}\|\|W^{(1)}\|\|X\|\epsilon)^2}{\beta};$$

- Then, the vulnerability of GINs is upper bounded by

$$\frac{(\|W^{(2)}\|\|W^{(1)}\|\|X\|\epsilon(2\|A\|+\epsilon))^2}{2\beta}.$$

Theoretical Results

Upper Bounds on GNN Vulnerability

We consider structural perturbations of the input graph (A, X), with a budget ϵ and 2-layer GNNs with 1-Lipschitz continuous activation functions and weight matrices $W^{(1)}, W^{(2)}$.

- Then, the vulnerability of GCNs is upper bounded by

$$\frac{2(\|W^{(2)}\|\|W^{(1)}\|\|X\|\epsilon)^2}{\beta};$$

- Then, the vulnerability of GINs is upper bounded by

$$\frac{(\|W^{(2)}\|\|W^{(1)}\|\|X\|\epsilon(2\|A\|+\epsilon))^2}{2\beta}.$$

Insight: Our upper bound on the vulnerability of a GNN is **smaller for large** β yielding a **more robust GNN.**

Experimental Results

Dataset	Attack Budget	GCNGuard	GCN-Jaccard	GCN-SVD	RGNN	NoisyGCN
Cora	Clean	77.5 ± 0.7	80.9 ± 0.7	80.6 ± 0.4	**83.5** ± **0.3**	83.2 ± 0.4
	Budget (5%)	75.8 ± 0.6	78.9 ± 0.8	78.4 ± 0.6	78.3 ± 0.6	**81.2** ± **0.7**
	Budget (10%)	74.7 ± 0.4	**76.7** ± **0.7**	71.5 ± 0.8	70.7 ± 0.8	74.5 ± 0.6
CiteSeer	Clean	70.1 ± 1.5	71.2 ± 0.7	70.7 ± 0.4	**72.3** ± **0.5**	71.9 ± 0.4
	Budget (5%)	69.9 ± 1.1	70.3 ± 2.3	68.9 ± 0.7	70.6 ± 0.7	**72.3** ± **0.6**
	Budget (10%)	70.0 ± 1.5	67.5 ± 2.1	68.8 ± 0.6	68.7 ± 1.2	**70.4** ± **0.8**
PubMed	Clean	84.5 ± 0.6	85.0 ± 0.5	82.7 ± 0.3	**85.1** ± **0.8**	85.0 ± 0.6
	Budget (5%)	**84.3** ± **0.9**	79.6 ± 0.3	81.3 ± 0.6	81.1 ± 0.7	81.8 ± 0.4
	Budget (10%)	**84.1** ± **0.3**	67.4 ± 1.1	81.1 ± 0.7	65.2 ± 0.4	73.3 ± 0.6
PolBlogs	Clean	93.1 ± 0.6	-	86.5 ± 0.8	94.9 ± 0.3	**95.2** ± **0.4**
	Budget (5%)	72.8 ± 0.8	-	**85.1** ± **1.6**	76.0 ± 0.8	79.7 ± 0.6
	Budget (10%)	68.7 ± 1.0	-	**84.8** ± **2.3**	69.2 ± 1.2	73.4 ± 0.5

Table: Node classification accuracy (± standard deviation) when subject to Mettack.

- Our NoisyGCNs **sometimes outperform** other defense methods.

Experimental Results

Table: Mean training time analysis (in s) of the NoisyGNN in comparison to other baselines for both the GCN and GIN instances.

Dataset	GCNGuard	GCN-Jaccard	RGCN	GCN-SVD	NoisyGCN
Cora	28.52	1.93	1.16	1.39	1.29
CiteSeer	36.04	1.58	1.23	1.12	1.24
PubMed	731.26	12.27	34.19	4.60	2.41
PolBlogs	18.17	5.17	0.96	0.80	0.65
Dataset	**GINGuard**	**GIN-Jaccard**	**RGCN**	**GIN-SVD**	**NoisyGIN**
Cora	48.93	3.12	1.31	1.51	1.93
CiteSeer	58.45	3.78	1.44	2.20	2.76
PubMed	963.58	16.28	41.09	6.33	7.86
PolBlogs	43.7	5.52	0.95	3.71	3.16

- Our NoisyGCNs **sometimes outperform** other defense methods.
- NoisyGNNs are **faster to train** than most other defense methods.

Experimental Results

Table: Classification accuracy ($\pm$ standard deviation) of combining defense methods with the proposed noise injection on different benchmark datasets.

Method	Cora	CiteSeer	PolBlogs
GINGuard	61.8±0.5	55.6±1.8	82.7±0.6
+ Noisy	**66.2±1.3**	**58.3±1.9**	**83.6±0.8**
GIN-Jaccard	70.4±1.1	61.2±2.3	-
+ Noisy	**72.9±0.8**	**64.9±1.8**	-
GCNGuard	69.5±0.7	66.2±0.6	64.7±0.8
+ Noisy	**72.4±1.2**	**68.9±0.9**	**65.8±1.3**
GCN-Jaccard	66.7±0.5	61.2±1.1	-
+ Noisy	**69.6±0.9**	**63.1±0.6**	-

- Our NoisyGCNs **sometimes outperform** other defense methods.
- NoisyGNNs are **faster to train** than most other defense methods.
- When **combined with other defense methods**, best performance is achieved.

Other Topics We Have Been Working On

- Analysed the Expressive Power of a GNN Operating on Paths in a Graph
  (Michel et al., 2023, ICML)
- Designed a GNN able to capture Neighbourhood Interaction Effects
  (Chatzianastasis et al., 2023, AAAI)
- Studied GNNs for Text Classification
  (Abbahaddou et al., 2023, NeurIPS Workshop)
- Graph Autoencoders for Joint Community Detection and Link Prediction
  (Salha-Galvan et al., 2022, Neural Networks Journal)
- Antibiotic Resistance Prediction Using GNNs
  (Qabel et al., 2022, NeurIPS Workshop)
- Improving GNNs at Scale: Approximate PageRank and CoreRank
  (Ramos Vela et al., 2022, NeurIPS Workshop)
- Sparsifying Weight Matrices in GNNs
  (Lutzeyer et al., 2022, ICLR Workshop)
- Analysing the Robustness of GNNs to Structural Noise
  (Seddik et al., 2022, AISTATS)
- Optimised Graph Shift Operators in GNNs for optimal graph representation
  (Dasoulas et al., 2021, ICLR)

Conclusions

- Graph Representation Learning is a highly active area of research at the moment gaining both academic and industrial interest.

Conclusions

- Graph Representation Learning is a highly active area of research at the moment gaining both academic and industrial interest.
- Graph Neural Networks are a versatile and powerful tool, that you may want to consider using.

Conclusions

- Graph Representation Learning is a highly active area of research at the moment gaining both academic and industrial interest.
- Graph Neural Networks are a versatile and powerful tool, that you may want to consider using.

Specifically, with regards to the presented project

- Both the introduction of noise and the orthonormalisation of weight matrices are viable avenues towards more robust Graph Neural Networks.

Thank you for your attention!

@JLutzeyer

References

Y. Abbahaddou, J. F. Lutzeyer & M. Vazirgiannis, "Graph Neural Networks on Discriminative Graphs of Words," *NeurIPS New Frontiers in Graph Learning Workshop,* 2023.

Y. Abbahaddou, S. Ennadir, J. F. Lutzeyer, M. Vazirgiannis & H. Boström, "Bounding the Expected Robustness of Graph Neural Networks Subject to Node Feature Attacks," *International Conference on Learning Representations(ICLR),* 2024.

L. A. Adamic & N. Glance, "The political blogosphere and the 2004 US election: divided they blog," *In Proceedings of the 3rd International Workshop on Link Discovery*, pp. 36–43, 2005.

H. Abdine, M. Chatzianastasis, C. Bouyioukos & M. Vazirgiannis, "Prot2Text: Multimodal Protein's Function Generation with GNNs and Transformers," *Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI),* 2024.

U. Alon & E. Yahav, "On the Bottleneck of Graph Neural Networks and its Practical Implications," In: *International Conference on Learning Representations (ICLR)*, 2020.

F. Borisyuk, S. He, Y. Ouyang, M. Ramezani, P. Du, X. Hou, C. Jiang, N. Pasumarthy, P. Bannur, B. Tiwana, P. Liu, "LiGNN: Graph Neural Networks at LinkedIn," *arXiv:2402.11139*, 2024.

M. Bronstein, "Graph ML at Twitter," *Twitter Engineering Blog Post*, `https://blog.twitter.com/engineering/en_us/topics/insights/2020/graph-ml-at-twitter`, 2020.

M. Bronstein, "Geometric Deep Learning: The Erlangen Programme of ML," *Keynote Talk at The International Conference on Learning Representations*, 2021.

M. Chatzianastasis, J. F. Lutzeyer, G. Dasoulas & M. Vazirgiannis, "Graph Ordering Attention Networks," *Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI),* 2023.

A. Deac, M. Lackenby & P. Veličković, "Expander Graph Propagation," *arXiv:2210.02997*, 2022.

B. Doerr, A. Dremaux, J. F. Lutzeyer & A. Stumpf, "How the move acceptance hyper-heuristic copes with local optima: drastic differences between jumps and cliffs," In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 2023.

G. Dasoulas, J. F. Lutzeyer & M. Vazirgiannis, "Learning Parametrised Graph Shift Operators," In: *International Conference on Learning Representations (ICLR)*, 2021.

S. Ennadir, Y. Abbahaddou, J. F. Lutzeyer, M. Vazirgiannis & H. Boström, "A Simple and Yet Fairly Effective Defense for Graph Neural Networks," *Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI),* 2024. 2017.

M. N. Hamid & I. Friedberg, "Transfer Learning Improves Antibiotic Resistance Class Prediction," *biorxiv:10.1101/2020.04.17.047316*, 2020.

F. Geerts & J. L. Reutter, "Expressiveness and Approximation Properties of Graph Neural Networks," *International Conference on Learning Representations (ICLR)*, 2022.

J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals & G. E. Dahl, "Neural message passing for Quantum chemistry," *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

I. J. Goodfellow, J. Shlens, & C. Szegedy, "Explaining and harnessing adversarial examples," International Conference of Learning Representations (ICLR), 2015.

S. Günnemann, "Graph Neural Networks: Adversarial Robustness," *Graph Neural Networks: Foundations, Frontiers, and Applications*, pp. 149–176, 2022.

A. Jain, I. Liu, A. Sarda & P. Molino, "Food Discovery with Uber Eats: Using Graph Learning to Power Recommendations," *Uber Engineering Blog Post*, `https://eng.uber.com/uber-eats-graph-learning/`, 2019.

J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli & D. Hassabis, "Highly accurate protein structure prediction with AlphaFold," *Nature*, pp. 583–589, 2021.

Thomas N. Kipf & M. Welling, "Semi-supervised classification with graph convolutional networks," *International Conference on Learning Representations (ICLR)*, 2017.

O. Lange & L. Perez, "Traffic prediction with advanced Graph Neural Networks," *DeepMind Research Blog Post*, `https://deepmind.com/blog/article/traffic-prediction-with-advanced-graph-neural-networks`, 2020.

Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, A. dos Santos, Costa, M. Fazel-Zarandi, R. Sercu, S. Candido & A. Rives, "Language Models of Protein Sequences at the Scale of Evolution Enable Accurate Structure Prediction," *biorxiv:10.1101/10.1101/2022.07.20.500902v1*, 2022.

G. Liu, D. B. Catacutan, K. Rathod, K. Swanson, W. Jin, J. C. Mohammed, A. Chiappino-Pepe, S. A. Syed, M. Fragis, K. Rachwalski, J. Magolan, M. G. Surette, B. K. Coombes, T. Jaakkola, R. Barzilay, J. J. Collins, J. M. Stokes, "Deep learning-guided discovery of an antibiotic targeting Acinetobacter baumannii," *Nature Chemical Biology*, pp. 1–9, 2023.

J. Lutzeyer, C. Wu & M. Vazirgiannis, "Graph Neural Network Simplification: Sparsifying the Update Step," *ICLR Workshop on Geometrical and Topological Representation Learning*, 2022.

G. Michel, G. Nikolentzos, J. Lutzeyer & M. Vazirgiannis, "Path Neural Networks: Expressive and Accurate Graph Neural Networks," *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J.E Lenssen, G. Rattan & M. Grohe, "Weisfeiler and Lehman Go Neural: Higher-order Graph Neural Networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 4602–4609, 2019.

G. Nikolentzos, M. Vazirgiannis, C. Xypolopoulos, M. Lingman & E. G. Brandt, "Synthetic Electronic Health Records Generated With Variational Graph Autoencoders," *NPJ Digital Medicine*, 2023.

A. Qabel, S. Ennadir, G. Nikolentzos, J. F. Lutzeyer, M. Chatzianastasis, H. Boström & M. Vazirgiannis, "Structure-Aware Antibiotic Resistance Classification Using Graph Neural Networks," *NeurIPS AI for Science Workshop*, 2022.

A. R. Ramos Vela, J. F. Lutzeyer, A. Giovanidis & M. Vazirgiannis, "Improving Graph Neural Networks at Scale: Combining Approximate PageRank and CoreRank," *NeurIPS New Frontiers in Graph Learning Workshop*, 2022.

G. Salha-Galvan, J. F. Lutzeyer, G. Dasoulas, R. Hennequin & M. Vazirgiannis, "Modularity-Aware Graph Autoencoders for Joint Community Detection and Link Prediction," *arxiv:2202.00961*, 2022.

G. Salha-Galvan, *Contributions to Representation Learning with Graph Autoencoders and Applications to Music Recommendation,* PhD thesis: École Polytechnique, Institut Polytechnique de Paris, 2022.

M. E. A. Seddik, C. Wu, J. F. Lutzeyer & M. Vazirgiannis, "Node Feature Kernels Increase Graph Convolutional Network Robustness," *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022.

J. M. Stokes, K. Yang, K. Swanson, W. Jin, A. Cubillos-Ruiz, N. M. Donghia, C. R. MacNair, S. French, L. A. Carfrae, Z. Bloom-Ackermann, V. M. Tran, A. Chiappino-Pepe, A. H. Badran, I. W. Andrews, E. J. Chory, G. M. Church, E. D. Brown, T. S. Jaakkola, R. Barzilay & J. J. Collins, "A Deep Learning Approach to Antibiotic Discovery," *Cell*, pp. 688–702, 2020.

L. Sun, Y. Dou, C. Yang, J. Wang, P. S. Yu & B. Li, "Adversarial attack and defense on graph data: A survey," *arXiv:1812.10528*, 2020.

S. Virinchi, A. Saladi & A. Mondal, "Recommending Related Products Using Graph Neural Networks in Directed Graphs," In: *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2022.

J. Wang, P. Huang, H. Zhao, Z. Zhang, B. Zhao & Dik Lun Lee, "Billion-scale Commodity Embedding for E-Commerce Recommendation in Alibaba," In Proceedings of the *24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 839–848, 2018.

K. Xu, W. Hu, J. Leskovec & S. Jegelka. "How powerful are graph neural networks?", *International Conference on Learning Representations (ICLR)*, 2019.

R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton & J. Leskovec, "Graph Convolutional Neural Networks for Web-Scale Recommender Systems," In Proceedings of the *24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 974–983, 2018.

Y. Zhou, H. Zheng & X. Huang, "Graph Neural Networks: Taxonomy, Advances and Trends," *arXiv:2012.08752*, 2020.