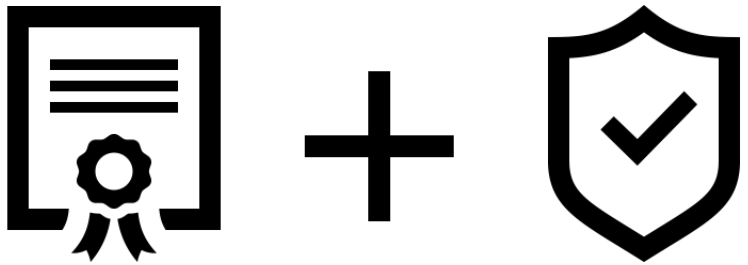
Certifiable Robustness and Robust Training for Graph Convolutional Networks

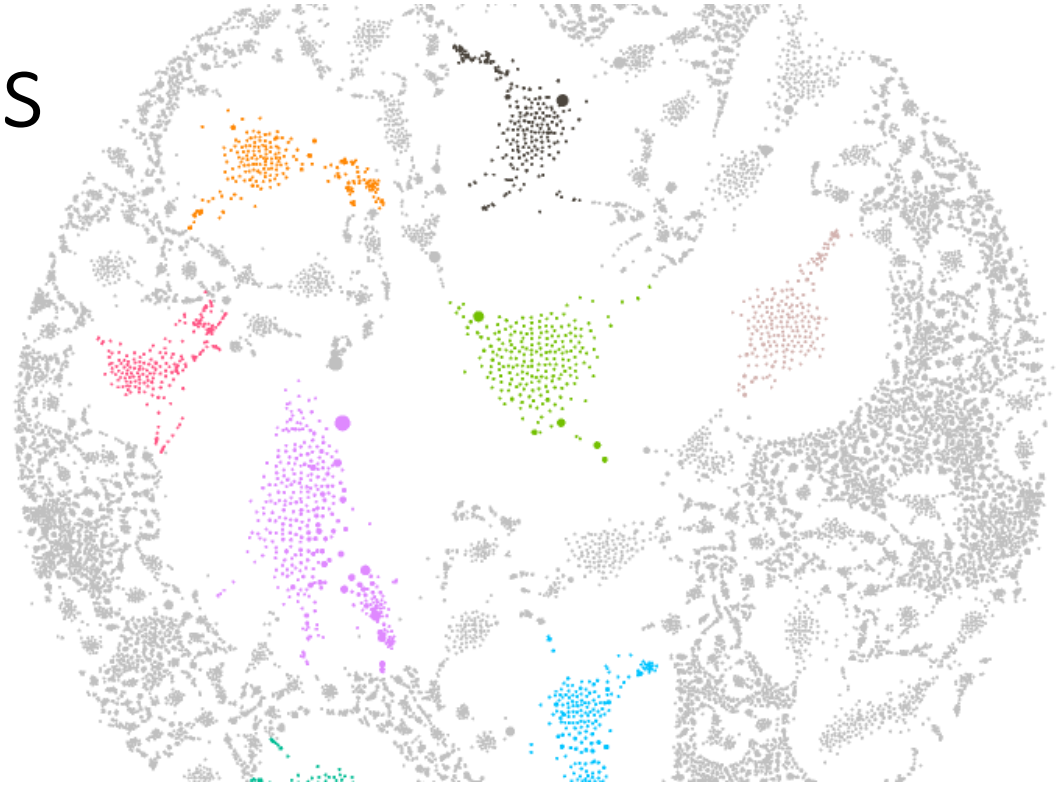


Daniel Zügner, Stephan Günnemann

Technical University of Munich, Germany

KDD 2019

Deep Learning on Graphs



[image: linkurio.us]

Graphs are ubiquitous

- Social networks
- Web graphs
- Knowledge graphs

Graph neural networks (GNNs): state of the art on tasks such as

- **Semi-supervised node classification** (our work's focus)
- Link prediction
- Unsupervised representation learning (node embeddings)

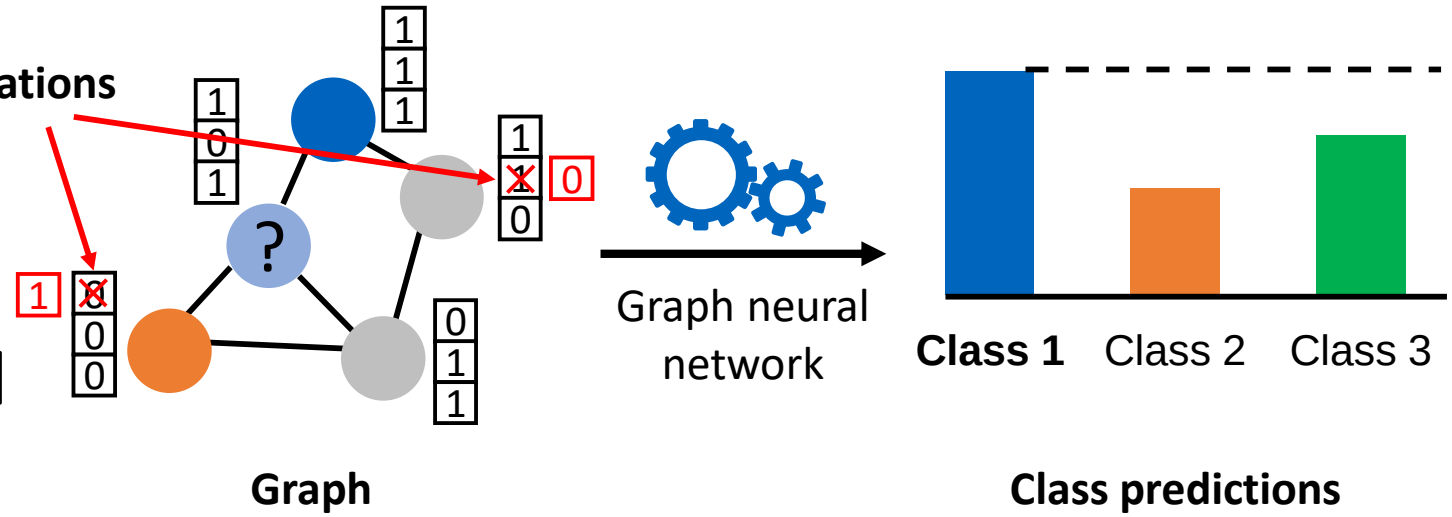
Graph Neural Networks are not robust

GNNs are not robust w.r.t. adversarial attacks.

[Dai et al. 2018]

[Wang et al. 2018]

[Zügner et al. 2018],
[Zügner and Günnemann 2019]



Adversarial attacks on GNNs:

- **Node feature perturbations** (this work's focus)
- Graph structure perturbations

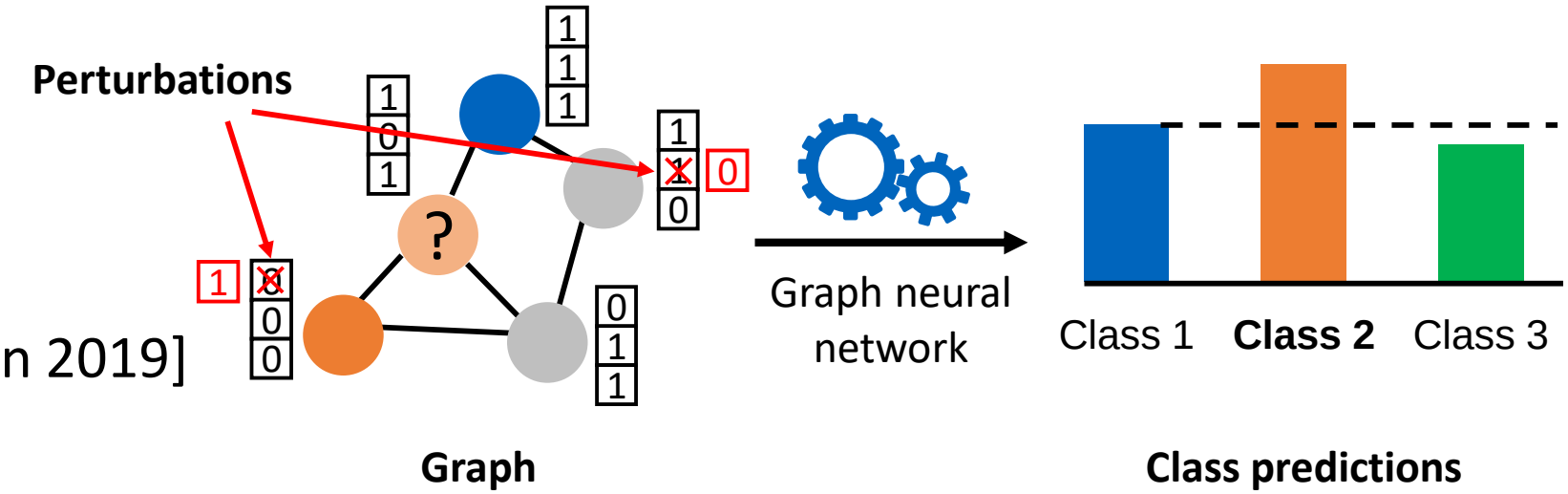
Graph Neural Networks are not robust

GNNs are not robust w.r.t. adversarial attacks.

[Dai et al. 2018]

[Wang et al. 2018]

[Zügner et al. 2018],
[Zügner and Günnemann 2019]



Adversarial attacks on GNNs:

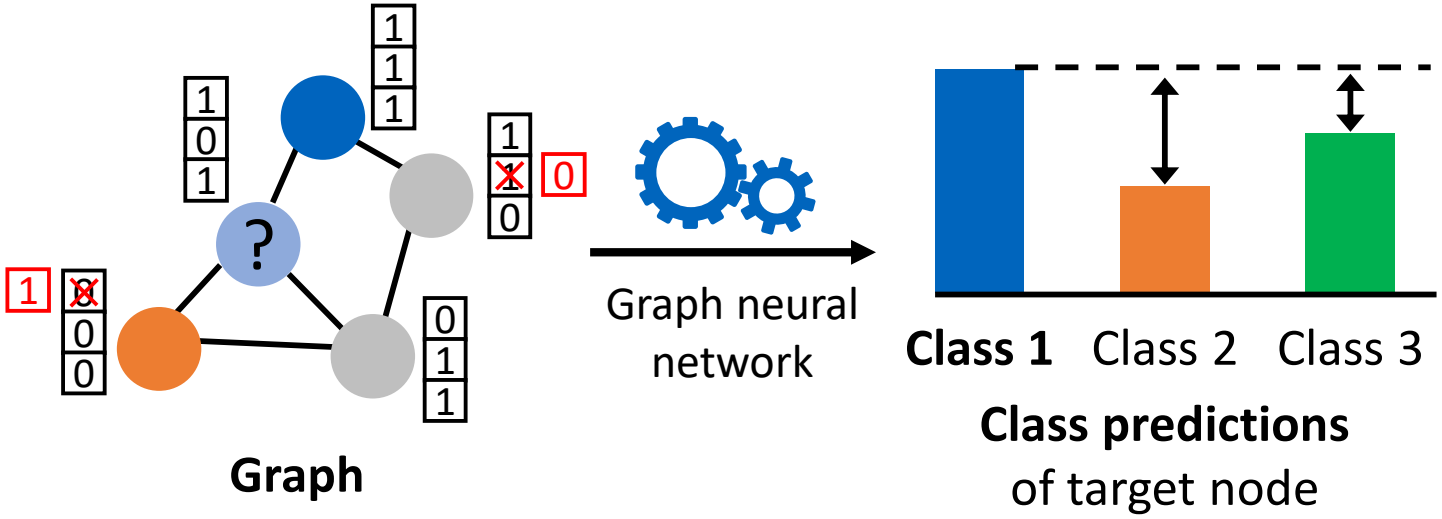
- **Node feature perturbations** (this work's focus)
- Graph structure perturbations

Contribution 1: Robustness Certification

Setting

- **Robustness certificate**
  - Mathematical guarantee that the **predicted class** of an instance does **not change** under **any** admissible perturbation
- **Binary node attributes**
  - e.g., multi-hot vector per node (paper) indicating words in its abstract
- L_0**-bounded** perturbations on the node attributes
  - At most q attributes may be changed per node
  - At most Q attributes may be changed in total (over all nodes)

Classification margin



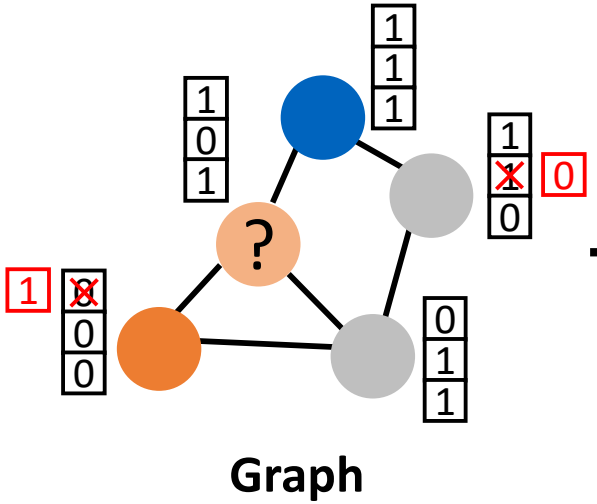
Classification margin:

$$m = \min_{c \neq c^*} \log p(c^*) - \log p(c)$$

> 0: correct classification

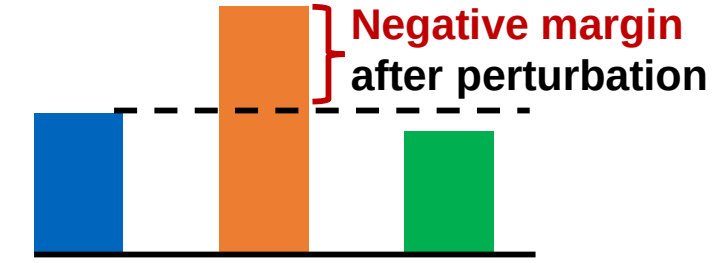
< 0: incorrect classification

Classification margin



Graph

Graph neural network



Class predictions
of target node

Classification margin:

$$m = \min_{c \neq c^*} \log p(c^*) - \log p(c)$$

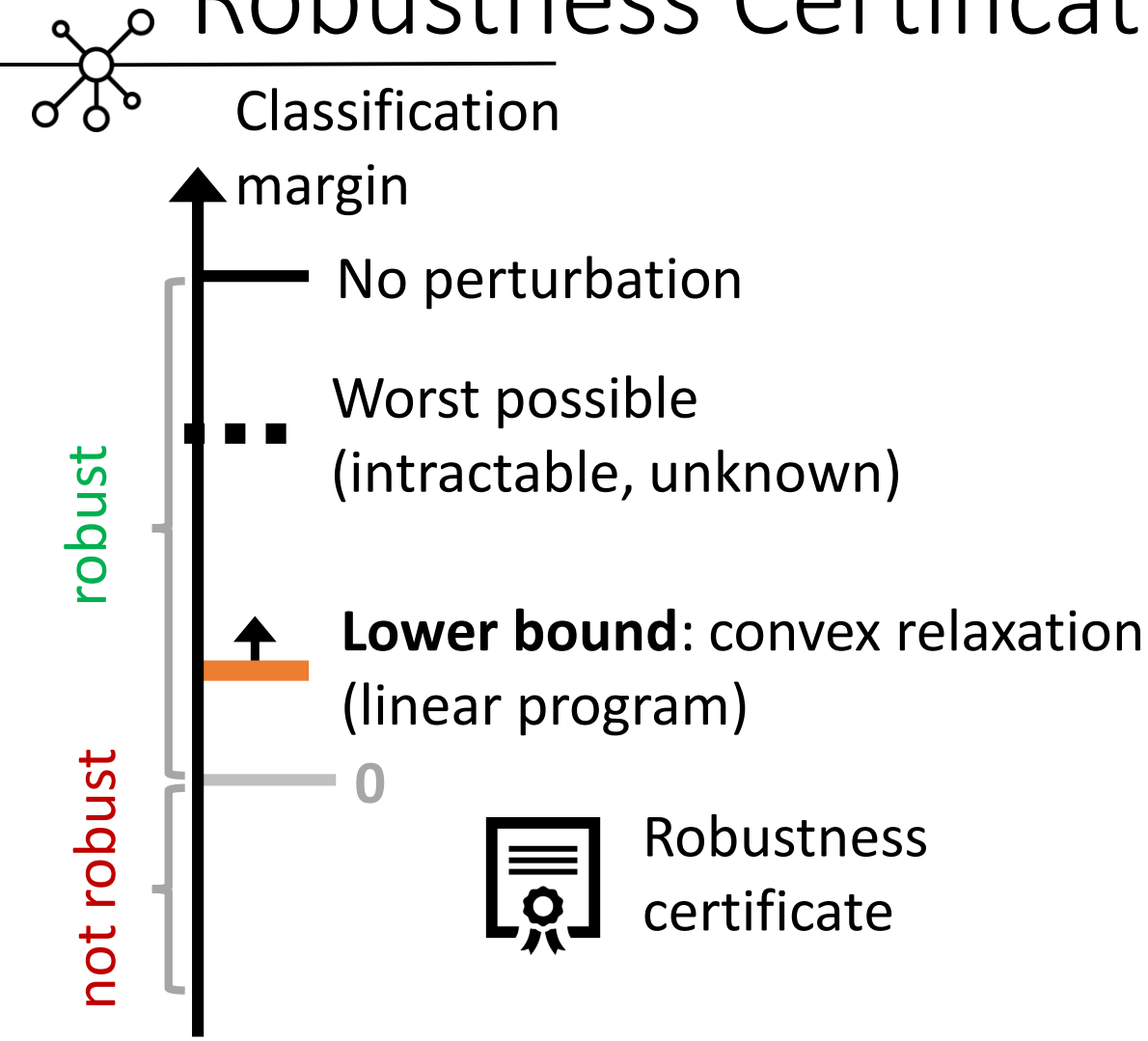
> 0: correct classification

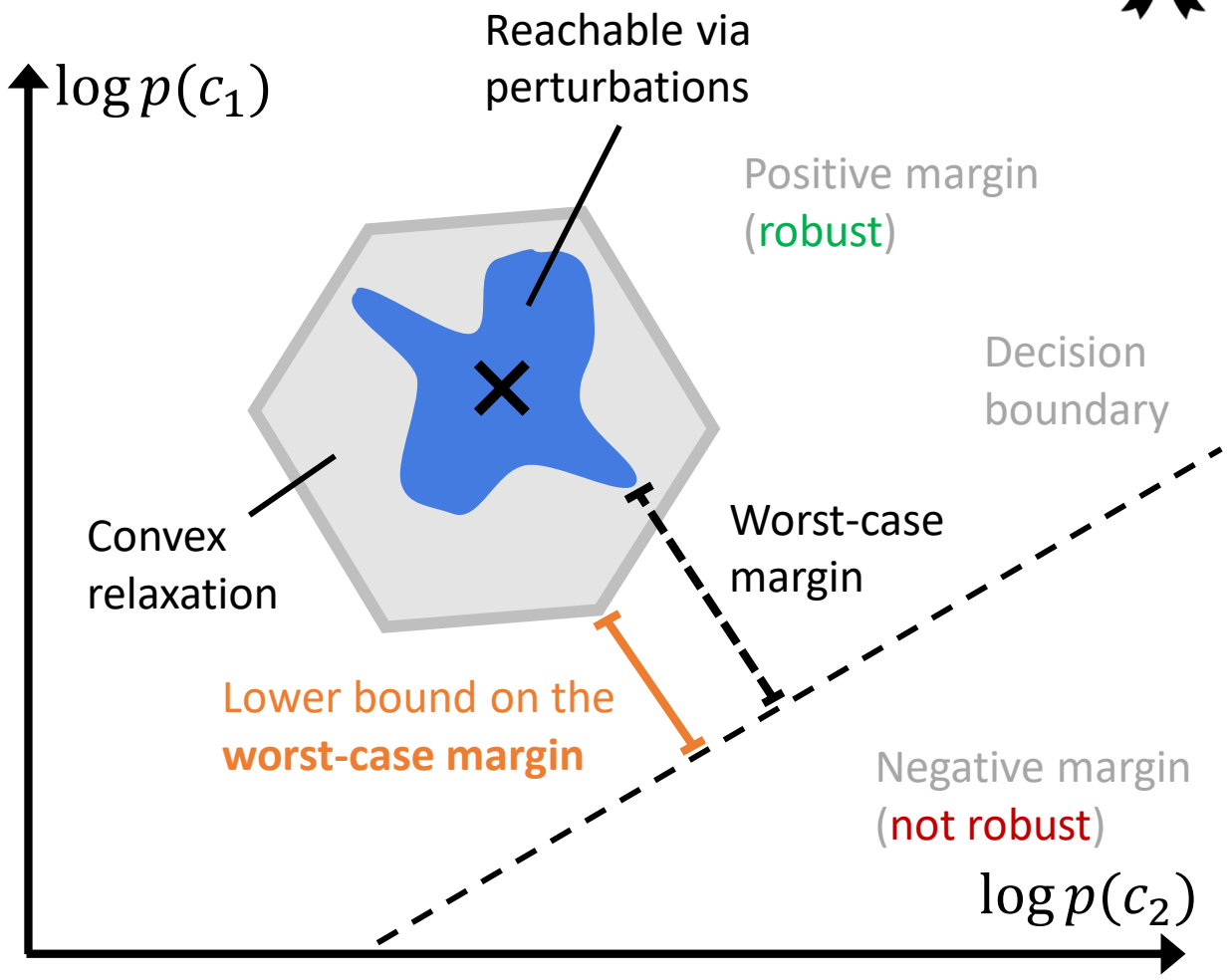
< 0: incorrect classification

Classification margin m

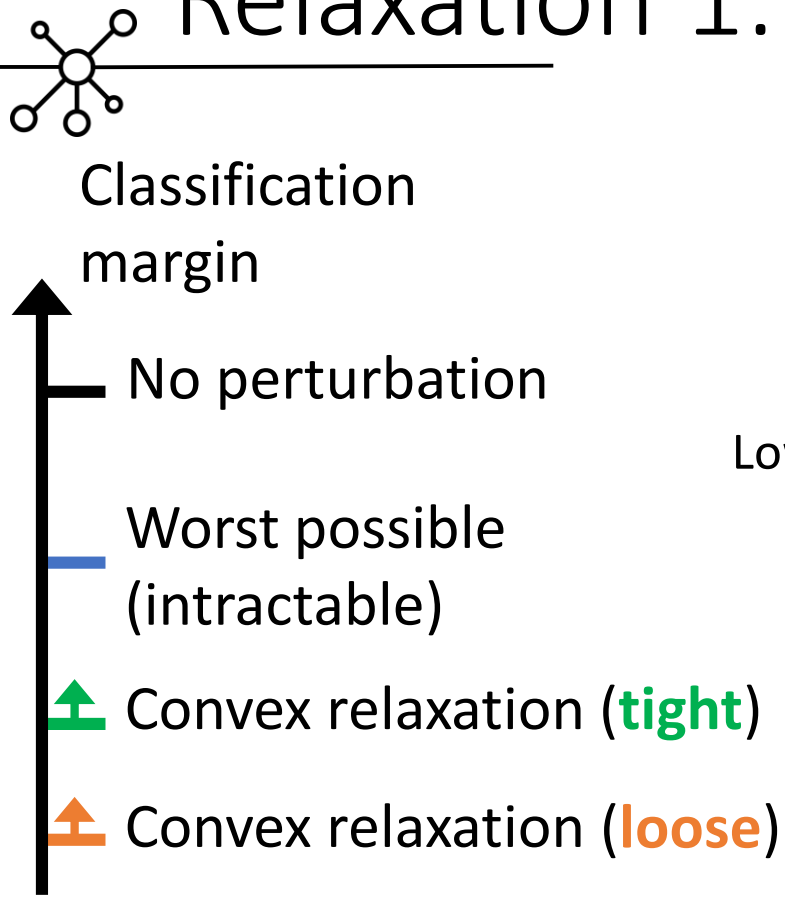
Worst-case margin $m^* = \underset{\text{perturbations}}{\text{minimize}} \; \underset{\text{class } c \neq c^*}{\min} \; \log p(c^*) - \log p(c)$

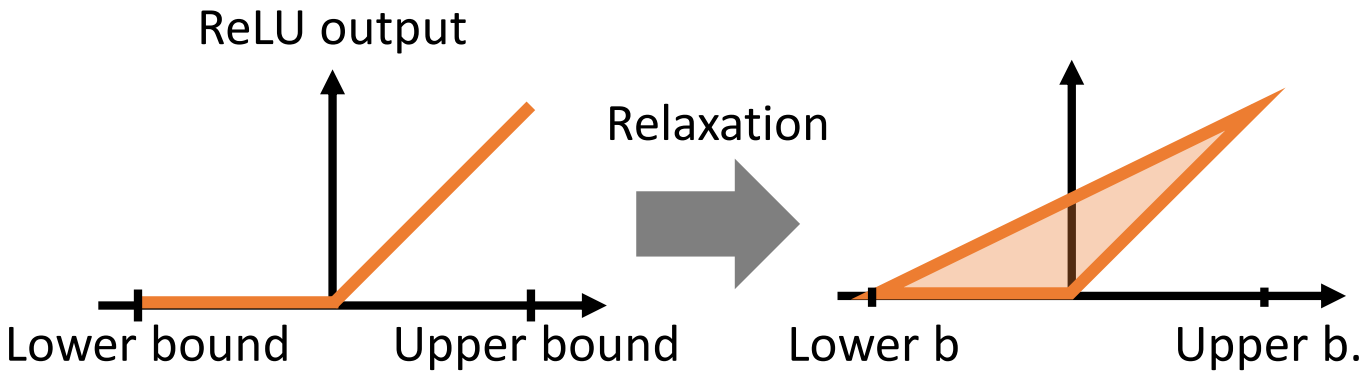
Robustness Certification

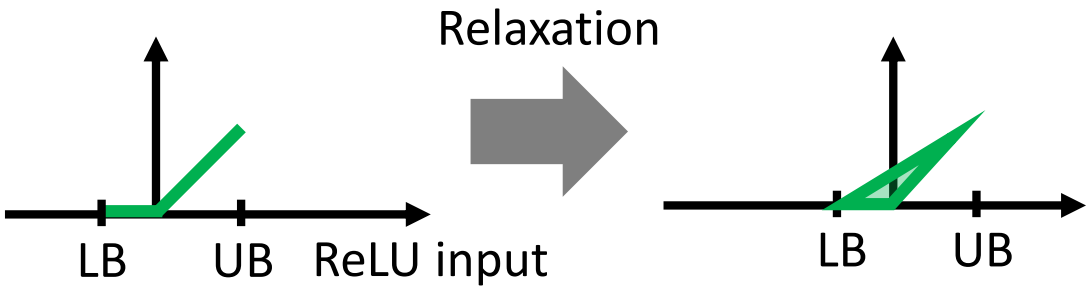




Relaxation 1: Convex ReLU relaxation





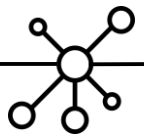


Less relaxation error; more certificates

We derive **tight bounds** on the hidden neurons‘ activations exploiting the **data discreteness** and the **graph structure**.

Convex ReLU relaxation: [Wong and Kolter 2018]

Relaxation 2: Continuous Attributes



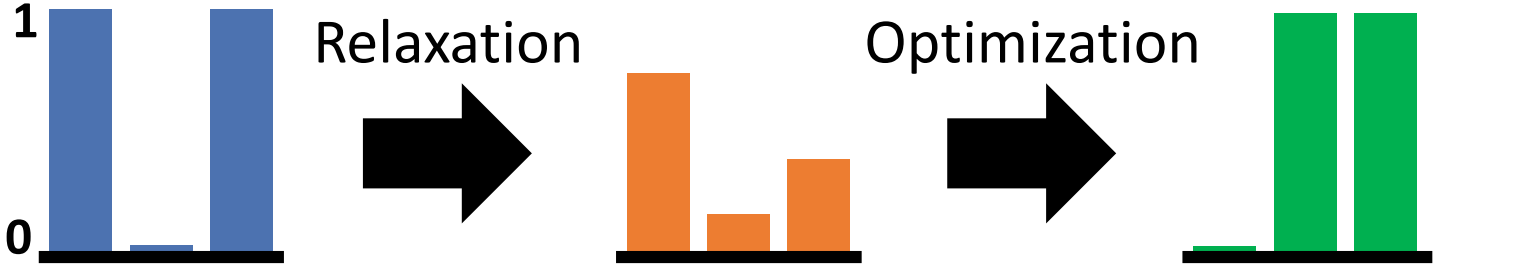
Classification margin

No perturbation

Worst possible (intractable)

Convex and continuous relaxation

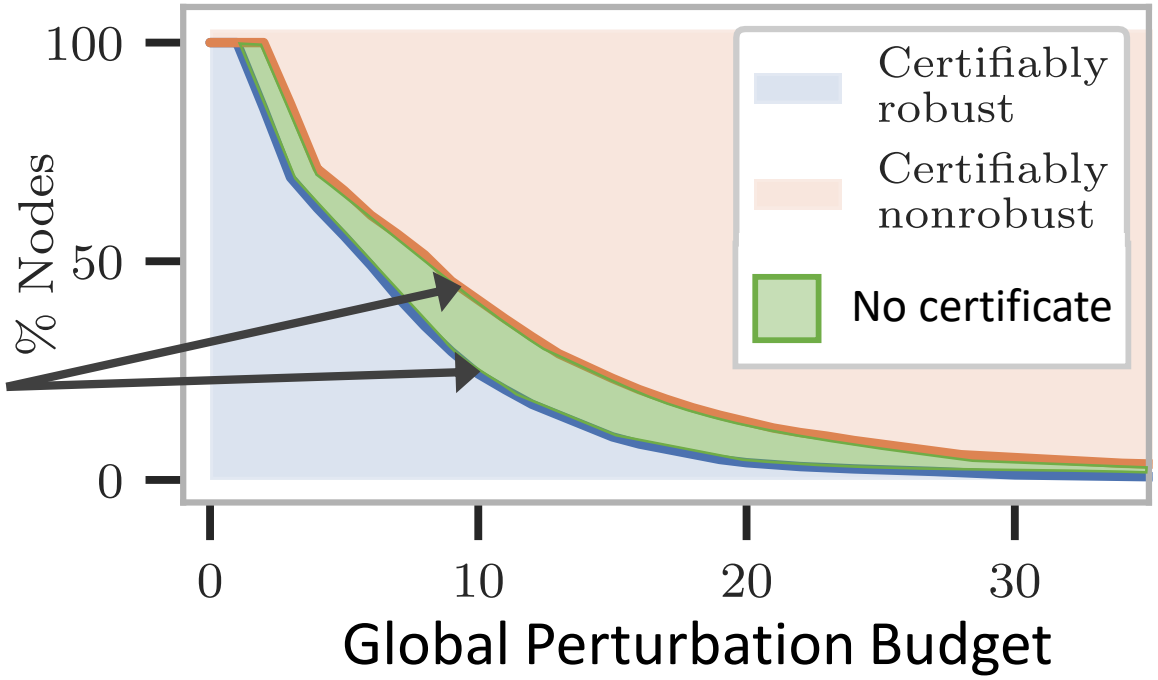
~~Continuous relaxation~~



Optimal solution is discrete despite continuous relaxation.
➔ **no** (additional) **relaxation error**!

Feed the optimal (binary) perturbed features into the original GNN.
Does its prediction change? If yes ➔ **adversarial example.**

Robustness Certification: Citeseer

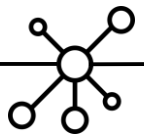


<25% of nodes robust,
>50% certifiably **nonrobust**
for 10 perturbations.

How can we improve robustness?

Dataset: Citeseer

Contribution 2: Robust Training of GCN



Classification margin

No perturbation

Worst possible (intractable)

Linear program **minimum** = **dual** program **maximum**

Any feasible point in the **dual**

- **Using duality**: efficient lower bounds without optimization.
- Lower bounds **differentiable** w.r.t. GNN weights
  ➔ optimize for robustness during **training**.

Training using Robust Hinge Loss

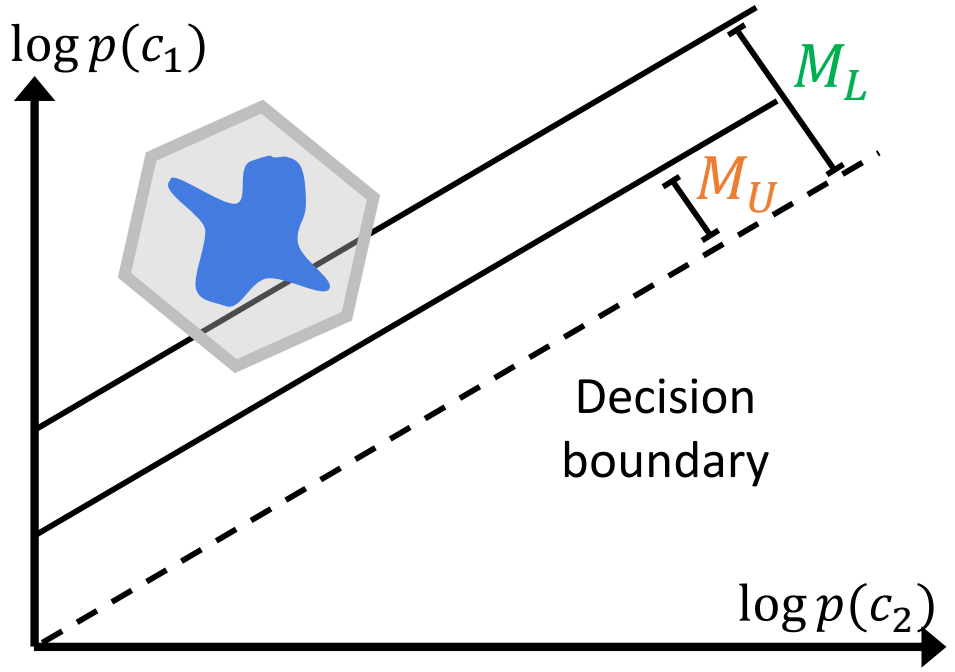
$$\min_{\theta} \sum_{v \in V_L} \mathcal{L}\left(\boldsymbol{p}_v, y_v\right)$$ Cross-entropy loss (labeled nodes)

$$+ \sum_{v \in V_L} \hat{\mathcal{L}}_{M_L}\left(-\boldsymbol{m}_v^*, y_v\right) + \sum_{v \in V \setminus V_L} \hat{\mathcal{L}}_{M_U}\left(-\boldsymbol{m}_v^*, \hat{y}_v\right)$$

Hinge loss (labeled nodes)

Hinge loss (unlabeled nodes)

Lower bound worst-case margins



Training using Robust Hinge Loss

$$\min_{\theta} \sum_{v \in V_L} \mathcal{L}(\boldsymbol{p}_v, y_v)$$ Cross-entropy loss (labeled nodes)

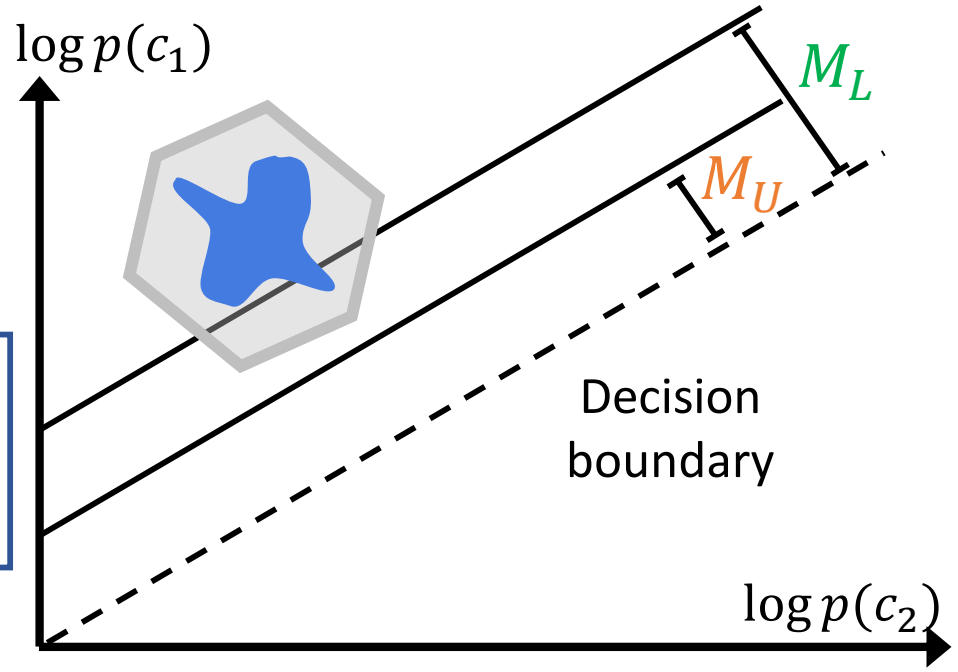
Hinge loss (labeled nodes)

Hinge loss (unlabeled nodes)

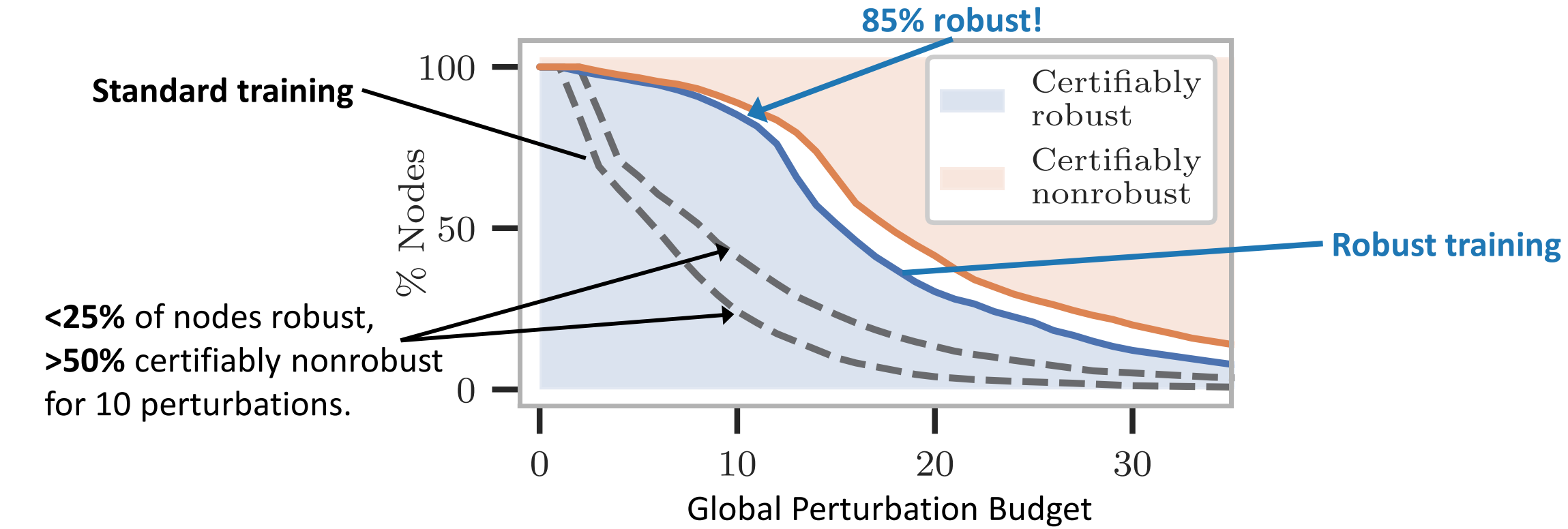
$$+ \sum_{v \in V_L} \hat{\mathcal{L}}_{M_L}(-\boldsymbol{m}^*_v, y_v) + \sum_{v \in V \setminus V_L} \hat{\mathcal{L}}_{M_U}(-\boldsymbol{m}^*_v, \hat{y}_v)$$

= 0 if all robust

Lower bound worst-case margins

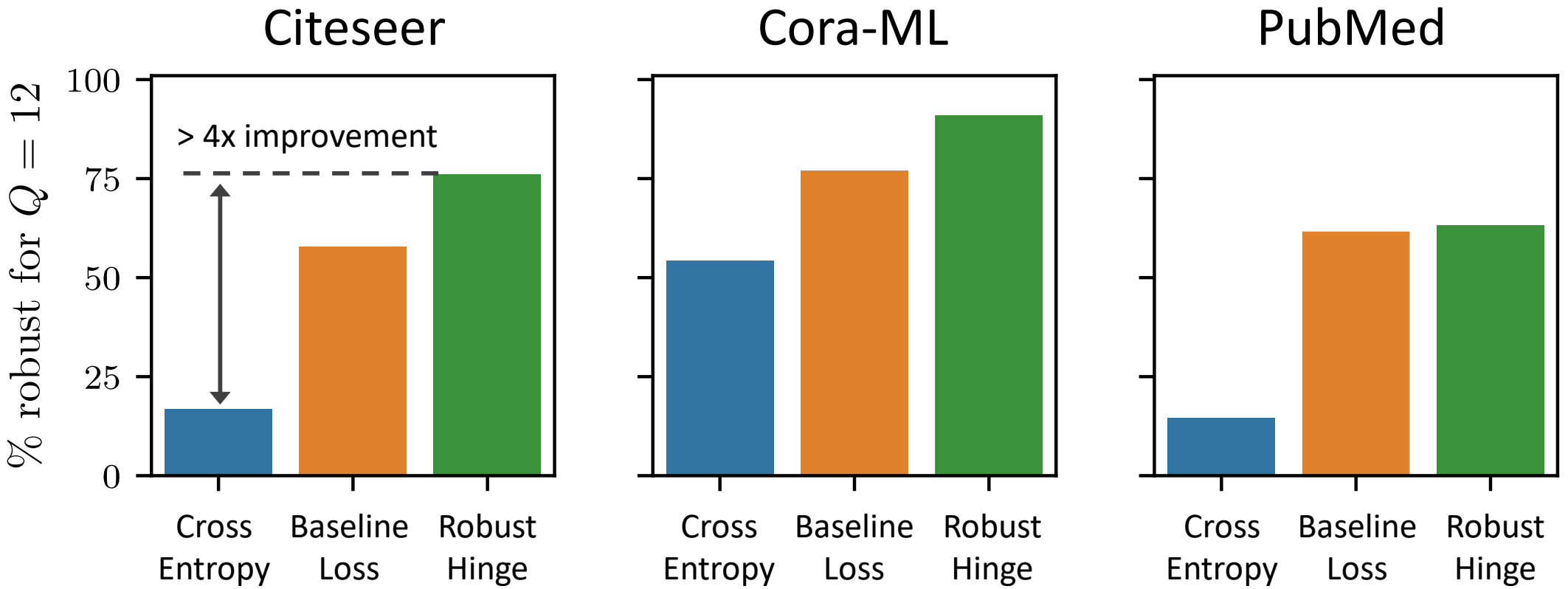


Results



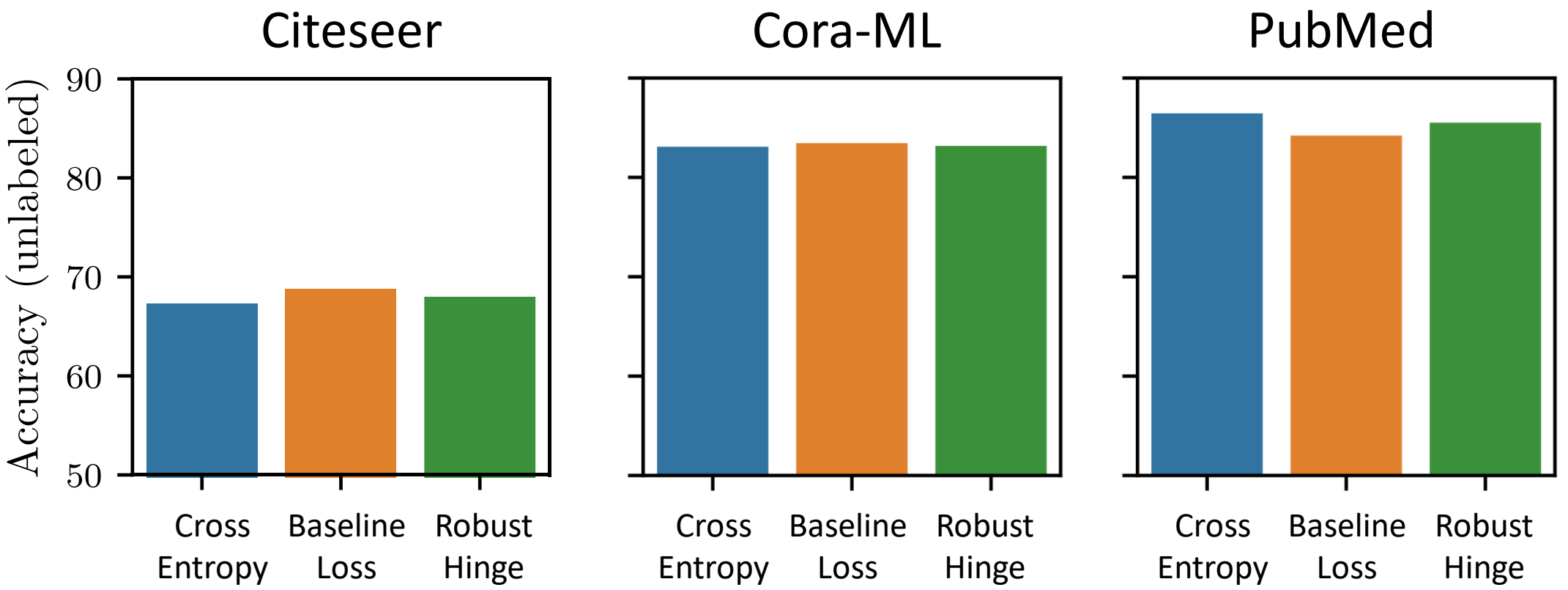
Dataset: Citeseer

Robust Hinge Loss increases Robustness



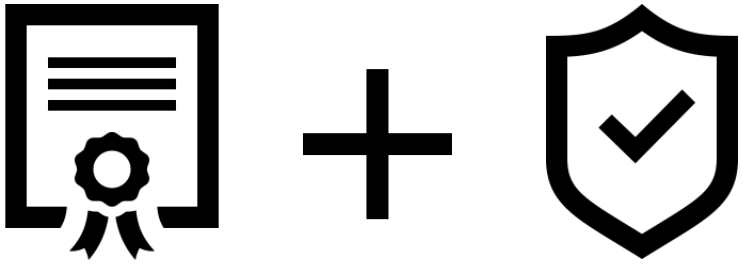
Baseline loss adapted from [Wong and Kolter 2018]

Robust Training: No Cost in Accuracy



Baseline loss adapted from [Wong and Kolter 2018]

Summary



- **Robustness certification** of graph neural networks via **convex relaxation.**
- GCN trained with standard training is **highly non-robust**
- Our robust training increases robust nodes by **up to 4x** without sacrificing **accuracy**.

github.com/danielzuegner/robust-gcn